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

LNAI 6096

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 I

1 Part I

Springer

Lecture Notes in Artificial Intelligence 6096

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 I

Springer

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 Ingenierias 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 3-642-13021-6 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-13021-2 Springer Berlin Heidelberg New York

springer.com

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 keynotes speeches. They were Nitesh Chawla, from the University of Notre Dame, USA, Óscar Cordón 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	University of the West of Scotland, UK
Nicolás García-Pedrajas	University of Córdoba, Spain
Francisco Herrera	University of Granada, Spain

Local Organizing Chair

César García-Osorio	University of Burgos, Spain

Special Session Chairs

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

Publicity Chair

Rafael Alcalá	University of Granada, Spain

Organizing Committee

Cecilio Angulo-Bahón	Technical University of Catalonia
Bonifacio Castaño-Martin	University of Alcalá
Antonio Fernández-Caballero	University of Castilla-La Mancha
Rafael del Castillo-Gomariz	University of Córdoba
Gonzalo Cerruela-García	University of Córdoba
Salvador García	University of Jaén
César García-Osorio	University of Burgos, Spain
Aida de Haro-García	University of Córdoba
Domingo Ortiz-Boyer	University of Córdoba
Jesús Maudes-Raedo	University of Burgos
Carlos Pardo-Aguilar	University of Burgos
Javier Pérez-Rodríguez	University of Córdoba
Juan Antonio Romero del Castillo	University of Córdoba
Miguel Ángel Salido	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 Bioinfomatics 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
Dapoigny, 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
Jacquenet, 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.
Alonso, C.
Alonso, J.
Alonso, R.S.
Alonso, S.
Álvarez, L.
Anguita, D.
Aranda-Corral, G. A.
Argente, E.
Arroyo Castillo, Á.
Bajo, J.
Barber, F.
Baruque, B.
Berlanga, A.
Bermejo, P.
Bielza, C.
Boström, H.
Bustamante, A.
Cabestany, J.
Cabrerizo, F.
Cao, L.
Carrasco, R.
Carrascosa, C.
Casacuberta, F.
Castanedo, F.
Catala, A.
Cavallaro, A.
Chang, C.
Chen, B.
Chen, L.
Chiang, C.
Cilla, R.
Corral, G.
Costa, J. A.
Damas, S.
De Bra, P.
De la Cal, E.
De la Ossa, L.
De Paz, J. F.

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

Additional Reviewers

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

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

Table of Contents – Part I

Application Systems

Application to Robotics

Applications of Agent-Based Systems

Applications to Business and Financial Markets

Applied Intelligent Systems for Future Classroom

Bioinformatics and Biomedical Applications

Computer Vision

Context Information in Intelligent Systems

Control, Simulation, and Modeling

Data Mining

Data Preprocessing in Data Mining

Decision Support Systems

Distributed Problem Solving with Artificial Intelligence Techniques

Table of Contents – Part II

Engineering Knowledge and Semantic Systems

Ensemble Learning: Methods and Applications

Evolutionary Computation and Applications

Fuzzy Systems and Applications

Heuristic Methods and Swarm Intelligence for Optimization

Industrial Applications of Data Mining: New Paradigms for New Challenges

Intelligent Agent-Based Systems

Interactive and Cognitive Environments

Internet Applications

Knowledge Management and Knowledge Based Systems

Machine Learning

Table of Contents – Part III

Neural Network Applications

New Frontiers in Data Analysis, Optimization and Visualization for Bioinfomatics and Neuroscience

Optimization and Heuristic Search

Real Life Applications

Semantic and Linguistic Visual Information: Applications

Soft Computing in Computer Vision/Image Processing

Soft Computing in Information Access Systems on the Web

Soft Computing Methods for Environmental and Industrial Applications

Tabu Search with Consistent Neighbourhood for Strip Packing

Giglia Gómez-Villouta, Jean-Philippe Hamiez*, and Jin-Kao Hao

LERIA, Université d'Angers, 2 Bd. Lavoisier, 49045 Angers (France)
{gomez,hamiez,hao}@info.univ-angers.fr

Abstract. This paper introduces a new tabu search algorithm for a strip packing problem. It integrates several key features: A *consistent* neighborhood, a fitness function including *problem knowledge*, and a diversification based on the *history* of the search. The neighborhood only considers valid, sometimes partial, packings. The fitness function incorporates measures related to the empty spaces. Diversification relies on a set of historically "frozen" objects. Experimental results are shown on a set of well-known hard instances and compared with previously reported tabu search algorithms as well as the best performing algorithms.

Keywords: Tabu search, strip packing, consistent neighborhood.

1 Introduction

This paper is dedicated to the regular, non-guillotine, and without rotation two-dimensional Strip Packing Problem (2D-SPP): Given a finite set of *rectangular* objects, pack **all** of them in a strip of fixed width while minimizing its height. 2D-SPP is a NP-hard problem with a number of practical applications [1,2,3,4].

Given the NP-hard nature of 2D-SPP, many (meta)heuristic procedures have been tried: Greedy randomized adaptive search procedure (GRASP) [5], intensification / diversification walk (IDW) [6], simulated annealing [7,8], tabu search (TS) [7,9,10,11], genetic algorithms [8,11,12], hybrid (meta)heuristics [8,11], and hyper-heuristics [13]. Exact algorithms have also been considered but they are usually limited to "small" instances [14,15].

In this paper, we present CTS (for "Consistent Tabu Search"), a new TS algorithm dedicated to 2D-SPP. Computational results suggest that CTS may be of great interest to efficiently solve 2D-SPP.

In the next section, the 2D-SPP is formally stated. Section 3 is devoted to the detailed presentation of our dedicated TS algorithm for the 2D-SPP. Experimental results are finally shown in Sect. 4.

2 Problem Formulation

A strip is a 2D vertical space with fixed width W and infinite height. The bottom-left (BL) corner of the strip stands for the $(0, 0)$ point of an xy-plane

* Contact author.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 1–10, 2010.

where the x-axis (respectively y-axis) is the direction of the width (resp. height). The set R, for "*R*ectangles", of $n \geq 2$ objects to be positioned in the strip is $R = \{r_1, \ldots, r_n\}$ where the *w*eight (resp. *h*eight) of each $r_{1 \leq i \leq n}$ is $0 < w_i^r \leq W$ (resp. $h_i^r > 0$). The 2D-SPP is then to determine the (x_i^r, y_i^r) coordinates of the BL corner of all $r_i \in R$ so as to minimize the height H of the resulting packing:

$$\text{Minimize: } H = \max_{1 \leq i \leq n} \{y_i^r + h_i^r\} \tag{1}$$

$$\text{Subject to: } 0 \leq x_i^r \leq W - w_i^r \wedge y_i^r \geq 0 \tag{2}$$

$$\wedge(x_i^r \geq x_j^r + w_j^r \vee x_i^r + w_i^r \leq x_j^r \tag{3}$$

$$\vee y_i^r \geq y_j^r + h_j^r \vee y_i^r + h_i^r \leq y_j^r) \ . \tag{4}$$

where (2) forces each r_i to be inside the strip and (3–4) specify that any two r_i and $r_{j \neq i}$ objects must not overlap neither horizontally nor vertically, respectively.

3 CTS: A Consistent Tabu Search for 2D-SPP

Tabu search is an advanced metaheuristic designed for tackling hard combinatorial optimization problems [16]. We first introduce here the way the problem is addressed (Sect. 3.1). Next sections (3.2–3.7) describe then the problem-specific components of CTS, where all p variables (with subscripts) are *p*arameters whose values will be given in the experimentation part (Sect. 4.1). The general CTS procedure is finally summarized in Sect. 3.8.

3.1 Solving Scheme

Let 2D-SPP$_{k>0}$ be the following *satisfaction* problem: Is there a solution s to 2D-SPP such that $H(s) \leq k$? Obviously, 2D-SPP is equivalent to find the lowest k such that 2D-SPP$_k$ holds.

CTS treats the 2D-SPP *optimization* problem (minimizing H) as *successive* 2D-SPP$_k$. Starting from a complete packing s_0 of height $H(s_0)$, e.g. obtained with a greedy method (see Sect. 3.3), CTS tackles 2D-SPP$_k$ with decreasing values of $H(s_0)$ for k. To be more precise, if CTS finds a solution s to 2D-SPP$_k$, it then tries to solve 2D-SPP$_{H(s)-p_H}$ ($p_H > 0$, for decrement of the height).

3.2 Search Space: A Direct Representation

Many approaches for the 2D-SPP consider a search space S composed of the set of all permutations of the objects, see [11] for instance. More precisely, a permutation π of $[1, \ldots, n]$ is used to introduce an order for all the objects which is followed by a given placement heuristic ϕ (or "decoder"). In other words, given (π, ϕ), one can pack all the objects using ϕ and according to the order indicated by π. Based on this permutation representation, several greedy placement heuristics have been investigated for the 2D-SPP. "Bottom Left Fill" (BLF, shortly described in next section) is such a heuristic [17].

CTS does not code packings with permutations but adopts a *direct* representation where a 2D-SPP$_k$ packing $s \in S$ (optimal or not, possibly partial) is a $\{L, E\}$ set:

- $L \subseteq R$ is the set of rectangles *properly* Located in the strip, i.e. r_i verifies (2) with $y_i^r + h_i^r \leq k \; \forall r_i \in L$ and (r_i, r_j) verifies (3–4) $\forall (r_i, r_{j \neq i}) \in L \times L$. So, let $\overline{L} \leftarrow R \backslash L$ be the set of "free" objects, i.e. rectangles not (yet) located.
- E is a set of *rectangular* Empty spaces in the strip. Each $e_i \in E$ is characterized by the coordinates (x_i^e, y_i^e) of its BL corner, a width $0 < w_i^e \leq W$, and a height $0 < h_i^e \leq k$, with $0 \leq x_i^e \leq W - w_i^e$ and $0 \leq y_i^e \leq k - h_i^e$. Each $e_i \in E$ is a *maximal* rectangle[1], i.e. $\forall (e_i, e_{j \neq i}) \in E \times E, x_i^e < x_j^e \vee x_i^e + w_i^e > x_j^e + w_j^e \vee y_i^e < y_j^e \vee y_i^e + h_i^e > y_j^e + h_j^e$ (e_i is not included in e_j).

3.3 Initial Configuration

CTS uses the BLF procedure [17] to construct an initial configuration $s_0 \in S$, where the π permutation orders all $r_i \in R$ first by decreasing width, secondly by decreasing height (when $w_i^r = w_{j \neq i}^r$), randomly last if necessary ($h_i^r = h_{j \neq i}^r$).

Basically, BLF places each object at the left-most and lowest possible free area. It is capable of filling enclosed wasted areas. Notice that, according to the way BLF is implemented, its worst time complexity goes from O(n^3) [19] to O(n^2) [20] for a permutation of n objects. We employed this decoder / order since previous experiments suggested that the BLF placement algorithm usually outperforms other decoders, see [19,21] for instance.

s_0 is a solution to 2D-SPP$_k$ $\forall k \geq H(s_0)$. So, s_0 provides a trivial upper bound for 2D-SPP: $H_{OPT} \leq H(s_0)$, where H_{OPT} is the OPTimum value of (1).

3.4 Fitness Function

This measure, also called "evaluation" or "cost" function, is a key component of TS because it guides the choices of the algorithm at each iteration. CTS uses the following f function (for "$f$itness", to be minimized) to evaluate a (possibly partial) 2D-SPP$_k$ packing $s \in S$:

$$f(s) = \begin{cases} M_w M_h \alpha \text{ if } E = \emptyset \\ \frac{M_w M_h \alpha \delta}{M_a} \text{ otherwise} \end{cases} . \tag{5}$$

where $M_w = \max_{r_i \in \overline{L}} w_i^r$ (Maximum $w$idth of free rectangles), $M_h = \max_{r_i \in \overline{L}} h_i^r$ (Maximum $h$eight of free rectangles), $\alpha = |\{r_i \in \overline{L} / w_i^r = M_w\}|$ (number of free objects with width M_w), $\delta = \sum_{e_j \in E} (W - x_j^e)(k - y_j^e)$[2], and $M_a = \max_{e_j \in E} w_j^e h_j^e$ (Maximum $a$rea of empty spaces).

Roughly speaking, the $f(s)$ value measures the quality of s with respect to 2D-SPP$_k$, the current satisfaction problem considered:

[1] The notion of "maximal rectangular empty space" is called "maximal area" in [18] and "maximal hole" in [7]. $|E|$ is at most in O(n^2) [18].

[2] δ measures the density of s. Indeed, a "small" δ value indicates that (almost) all e_j are concentrated close to the top-right corner of the strip.

- $f(s) = 0$ (or $\overline{L} = \emptyset$ equivalently) signifies that s is a solution to 2D-SPP$_k$, i.e. that $H(s) \leq k$. In this case, s is an *optimal* solution to 2D-SPP (i.e. 2D-SPP$_{k'}$ admits no solution $\forall k' < k$) if $E = \emptyset$: $H_{OPT} = k$.
- $f(s) > 0$ (or $\overline{L} \neq \emptyset$) indicates a *partial* packing. Here, $E = \emptyset$ means that 2D-SPP$_{k'}$ have no solution $\forall k' \leq k$ and that a trivial *lower bound* has been found for 2D-SPP: $H_{OPT} > k$.

f is used to compare any $(s_1, s_2) \in S \times S$: With respect to 2D-SPP$_k$, s_1 is better than s_2 if $f(s_1) < f(s_2)$. However, note that it is inadequate when s_1 and s_2 are both solutions to 2D-SPP$_k$, i.e. when $f(s_1) = f(s_2) = 0$. In this case, s_1 is better than s_2 if $H(s_1) < H(s_2)$.

3.5 Neighborhood and Search Strategy

The neighborhood N is another key element of TS. It defines a structure of the search space S and determines the paths the algorithm will follow to explore S. At each iteration, a *best* neighbor $s' \in N(s)$ is sought to replace the current configuration s even if s' does not improve s in terms of the fitness function f. To be more precise, a neighborhood N over S is any function that associates to each individual $s \in S$ some solutions $N(s) \subset S$. A solution s is a "local optimum" if s is the best (with respect to N and f) among the solutions $s' \in N(s) \cup \{s\}$. The notion of neighborhood can be explained in terms of the "move" operator. Typically, applying a move μ to a solution s changes slightly s and leads to a neighboring solution $s' \in N(s)$. This transition from a solution s to a neighbor s' is denoted by $s' = s \oplus \mu$. Let $\Gamma(s)$ be the set of all possible moves which can be applied to s, then the neighborhood $N(s)$ of s can formally be defined by: $N(s) = \{s \oplus \mu / \mu \in \Gamma(s)\}$.

The main goal of the CTS neighborhood is to empty $\overline{L}$. Basically, it moves one rectangle r_i from $\overline{L}$ to L, at the BL corner either of an empty space $e_j \in E$ (defining a sub-neighborhood N_E) or of another $r_j \in L$ (defining N_L). This location for r_i may generate overlaps with a set $L_i \subseteq L$ of other rectangles: $L_i = \{r_{j \neq i} \in L / x_i^r < x_j^r + w_j^r \wedge x_i^r + w_i^r > x_j^r \wedge y_i^r < y_j^r + h_j^r \wedge y_i^r + h_i^r > y_j^r\}$. All $r_j \in L_i$ are thus removed from L and added to $\overline{L}$ to repair these overlaps. This principle, known as "ejection chains", is used to make s' *consistent* with respect to (3–4). Finally, notice that locating r_i in the strip and the possible deletion of all $r_j \in L_i$ imply updates of E. This is done using the efficient "incremental" procedures introduced in [7].

Let s_* be the overall *best* complete packing, according to (1), found by CTS at iteration m_* (initially $s_* \leftarrow s_0$ and $m_* \leftarrow 0$). Each time a move is performed from s to s', at iteration m, s_* and m_* are updated ($s_* \leftarrow s'$ and $m_* \leftarrow m$) once $\overline{L} = \emptyset \wedge f(s') \leq f(s_*)$.[3]

***N_E*: Consider First the Empty Spaces.** CTS examines two cases here, with two different objectives.

[3] $f(s') = f(s_*)$ may occur only after diversification D_T described in Sect. 3.7. In this particular case, s_* and m_* are updated according to some probability $p_{\approx}$.

$N_E^{|\overline{L}|}$*: Reduce* $|\overline{L}|$*.* **All** $r_i \in \overline{L}$ are tried to be located in the strip to the BL corner of **all** $e_j \in E / x_j^e + w_i^r \leq W \wedge y_j^e + h_i^r \leq k \wedge w_j^e \geq w_i^r \wedge h_j^e \geq h_i^r$ (r_i fits *entirely* in e_j). This generates $|\overline{L}|$ sets $N_E^{|\overline{L}|}(s,i)$ of neighbors for s, some possibly empty: $N_E^{|\overline{L}|}(s) = \cup_{r_i \in \overline{L}} N_E^{|\overline{L}|}(s,i)$.

Note that $N_E^{|\overline{L}|}(s) \neq \emptyset$ means that there is at least a $(r_i, e_j) \in \overline{L} \times E$ such that r_i fits in e_j, i.e. that $|\overline{L}|$ can be reduced. Furthermore, in this case, locating r_i will generate no overlap ($L_i = \emptyset$), hence no repair is needed.

Let $\lfloor N_E^{|\overline{L}|}(s) \rfloor \subseteq N_E^{|\overline{L}|}(s)$ be the set of the *best evaluated* neighbors s' of s: $\lfloor N_E^{|\overline{L}|}(s) \rfloor = \{s'_1 \in N_E^{|\overline{L}|}(s) / \forall s'_2 \in N_E^{|\overline{L}|}(s), f(s'_1) \leq f(s'_2)\}$. If $\lfloor N_E^{|\overline{L}|}(s) \rfloor \neq \emptyset \wedge \overline{L} = \emptyset \; \forall s' \in \lfloor N_E^{|\overline{L}|}(s) \rfloor$, select randomly one $s' \in \lfloor N_E^{|\overline{L}|}(s) \rfloor$ minimizing (1) to become the new starting configuration for the next 2D-SPP$_{H(s')-p_H}$ problem (s' is a solution to 2D-SPP$_k$) and possibly update s_* and m_*. Otherwise, if $\lfloor N_E^{|\overline{L}|}(s) \rfloor \neq \emptyset$, select randomly a $s' \in \lfloor N_E^{|\overline{L}|}(s) \rfloor$ minimizing (5) and make $s \leftarrow s'$.

N_E^D*: Make the Packing Denser.* This case occurs only if $N_E^{|\overline{L}|}$ is not applicable from s, i.e. when $N_E^{|\overline{L}|}(s) = \emptyset$ ($|\overline{L}|$ cannot be reduced).

Here again, **all** $r_i \in \overline{L}$ are tried to be located to the BL corner of **all** $e_j \in E$ but the previous condition on e_j and r_i is relaxed to $x_j^e + w_i^r \leq W \wedge y_j^e + h_i^r \leq k$ since we now know that no r_i can fit *entirely* in a e_j (so, overlaps will temporarily appear and be repaired).

Note that $N_E^D(s) = \emptyset$ means either, $\forall (r_i, e_j) \in \overline{L} \times E$, that all e_j are located rather to the right of the strip with $x_j^e + w_i^r > W$ or to its top with $y_j^e + h_i^r > k$, or that it is forbidden to remove all rectangles overlapping with r_i (see Sect. 3.6).

Let $\lfloor N_E^D(s) \rfloor$ be defined similarly to $\lfloor N_E^{|\overline{L}|}(s) \rfloor$. If $\lfloor N_E^D(s) \rfloor \neq \emptyset$, select randomly one $s' \in \lfloor N_E^D(s) \rfloor$ minimizing (5) to become the new "current" configuration for the next iteration ($s \leftarrow s'$).

$\boldsymbol{N_L}$**: Possibly Consider All** $\boldsymbol{r_j \in L}$**.** This second sub-neighborhood is explored only if N_E is not applicable from s, i.e. when $N_E^{|\overline{L}|}(s) = N_E^D(s) = \emptyset$.

From the current s, **all** $r_i \in \overline{L}$ are tried to be located to the BL corner of **all** $r_j \in L / (w_i^r \neq w_j^r \vee h_i^r \neq h_j^r) \wedge x_j^r + w_i^r \leq W \wedge y_j^r + h_i^r \leq k$ (r_i and r_j have different sizes). This generates $|\overline{L}|$ sets $N_L(s,i)$ of neighbors for s with $0 \leq |N_L(s,i)| \leq |L|$: $N_L(s)$ is the union of these sets.

Similarly to N_E^D, note that $N_L(s) = \emptyset$ means that all $r_j \in L_i$ cannot be removed from the strip (see Sect. 3.6).

Let $\lfloor N_L(s) \rfloor \subseteq N_L(s)$ be the set of the *best evaluated* neighbors of s according to (5): $\lfloor N_L(s) \rfloor = \{s'_1 \in N_L(s) / \forall s'_2 \in N_L(s), f(s'_1) \leq f(s'_2)\}$. If $N_L(s) \neq \emptyset$, choose $s' \in \lfloor N_L(s) \rfloor$ at random for the next iteration.

A Worst Case. If neither N_E nor N_L is applicable from s, i.e. $N(s) = N_E^{|\overline{L}|}(s) = N_E^D(s) = N_L(s) = \emptyset$, apply diversification (see Sect. 3.7).

3.6 Tabu List

To avoid the problem of possible cycling and to allow the search to go beyond local optima, TS introduces the notion of "tabu list", one of the most important components of the method. A tabu list is a special short term memory that maintains a selective history, composed of previously encountered solutions or more generally pertinent attributes (or moves) of such solutions. A simple TS strategy based on this short term memory consists in preventing solutions previously visited from being reconsidered for the next p_τ iterations (integer p_τ, called "tabu tenure", is problem dependent). Now, at each iteration, TS searches for a best neighbor from this dynamically modified neighborhood.

At current iteration m, since a CTS move from s to a neighbor $s' \in N(s)$ consists in locating one $r_i \in \overline{L}$ in the strip, it seems quite natural to forbid r_i leaving the strip from s'. This "reverse" move will then be stored in the tabu list τ for a duration $0 < p_\tau \leq n$ to indicate that r_i cannot be removed from the strip at least up to iteration $m + p_\tau : \tau \leftarrow \tau \cup \{(i, m + p_\tau)\}$.

Note that τ is made empty at the beginning of the search or when CTS finds a solution s for 2D-SPP$_k$, i.e. if $\overline{L} = \emptyset$.

3.7 Diversification

When $N(s) = \emptyset$ (s has no neighbor) or s_* keeps unchanged for a number $p_* > 0$ of iterations (integer p_*), CTS first resets τ and reloads s_*: $\tau \leftarrow \emptyset, s \leftarrow s_*$. This new current *complete* packing s is then perturbed according to two different Diversification schemes called D_I (for "Interchange", performed with probability p_D) and D_T (for "Tetris-like", probability $1 - p_D$). After perturbation, p_* supplementary moves are given to CTS to update s_*.

***D_I*: A Basic Perturbation.** L and $\overline{L}$ are first built according to 2D-SPP$_k$ with $k = H(s) - p_H$: $L \leftarrow \{r_i \in R / y_i^r + h_i^r \leq k\}, \overline{L} \leftarrow R \backslash L$. D_I considers then all $(r_i, r_j) \in L \times \overline{L} / (w_i^r \neq w_j^r \vee h_i^r \neq h_j^r) \wedge x_i^r + w_j^r \leq W \wedge y_i^r + h_j^r \leq k$ (r_i and r_j have different sizes). It simply interchanges two such rectangles (randomly selected) and makes r_j tabu. Note that swapping r_i with r_j may cause overlaps for r_j. In this case, L_j is repaired like in Sect. 3.5.

***D_T*: A Perturbation Based on the History.** During the overall search process, CTS keeps for each $r_i \in R$ the number F_i (for "Frequency") of times r_i leaved the strip.[4] D_T considers a π_F permutation that orders all $r_i \in R$ first by *increasing* frequencies, secondly by decreasing widths (when $F_i = F_{j \neq i}$), then by decreasing heights ($w_i^r = w_{j \neq i}^r$), randomly last if necessary ($h_i^r = h_{j \neq i}^r$). Let the set $\lfloor F \rfloor$ be composed of the first p_τ elements of π_F.

All $r_i \in \lfloor F \rfloor$ are first temporarily removed from the strip and their frequencies are updated.[5] Then, the *partial* packing is pushed down to the basis of the strip,

[4] $F_{1 \leq i \leq n} = 0$ at the beginning of the search.

[5] $F_i \leftarrow 2F_i$ is used here to avoid considering (almost) the same $\lfloor F \rfloor$ set in next applications of D_T while $F_i \leftarrow F_i + 1$ is applied when performing a move.

like in the famous Tetris game. Finally, all $r_i \in \lfloor F \rfloor$ are sorted like in Sect. 3.3 and relocated in the strip with BLF.

CTS deals now with 2D-SPP$_{H(s)-p_H}$: $L \leftarrow \{r_i \in R / y_i^r + h_i^r \leq H(s) - p_H\}, \overline{L} \leftarrow R \backslash L$. This means that CTS possibly considers 2D-SPP$_k$ with $k \geq H(s_0) \geq H(s_*)$.

3.8 CTS: The General Procedure

The CTS algorithm begins with an initial *complete* packing (Sect. 3.3). Then it proceeds iteratively to solve a series of 2D-SPP$_k$ *satisfaction* problems. If CTS finds a solution s to 2D-SPP$_k$, it then tries to solve 2D-SPP$_{H(s)-p_H}$.

While it is not mentioned here for simplicity, note that CTS can also end (see Step 3 below) before reaching the Maximum number of allowed moves $p_M \geq 0$ (integer). This may occur each time s_* is updated whenever the optimum height H_{OPT} (or an upper bound) is known and $H(s_*) \leq H_{OPT}$.

1. *Initialization.* Build s using BLF, $m \leftarrow 0, s_* \leftarrow s, m_* \leftarrow m$.
2. *Generation of the starting configuration for the next 2D-SPP$_k$.*
   $L \leftarrow \{r_i \in R / y_i^r + h_i^r \leq H(s) - p_H\}, \overline{L} \leftarrow R \backslash L$.
3. *Stop condition.* If $m = p_M$ Then: Return $H(s_*)$ and s_*.
4. *Exploration of the neighborhood.* If $N(s) = \emptyset$ Then: Go to step 5.
   $m \leftarrow m + 1$. Update s according to $N(s)$.
   If s_* is replaced by s or $\overline{L} = \emptyset$ Then: Go to step 2.
   If $(m - m_*)$ mod $p_* \neq 0$ Then: Go to step 3.
5. *Diversification.* Modify s using D_I or D_T according to p_D. Go to step 3.

4 Experimentations

We used the set of 21 well-known hard instances defined in [19].[6] The main characteristics of this benchmark are given in the 4 first columns of Table 1, each of the 7 categories "Cat." being composed of 3 different instances.

4.1 Experimentation Conditions

The comparison is based on the percentage gap γ of a solution s from the optimum: $\gamma(s) = 100 * (1 - H_{OPT}/H(s))$. For CTS, mean gap $\overline{\gamma}$ (resp. best gap γ^*) is averaged over a number of 5 runs (resp. over best runs only) per instance.

The CTS parameters are: $p_H = 1$ (to build the starting configuration of 2D-SPP$_k$, the current satisfaction problem considered), $p_{\approx} = [0.4, \ldots, 0.8]$ (probability that a complete packing s replaces s_* whenever $f(s) = f(s_*)$), $p_\tau = [2, \ldots, 6]$ (tabu tenure), $p_* = [200, \ldots, 500]$ (maximum number of moves to update s_*), $p_D \in [0.7, \ldots, 1]$ (probability to apply diversification D_I), $p_M \in [1\,000\,000, \ldots, 20\,000\,000]$ (maximum number of allowed moves per run).

CTS is coded in the c programming language (gcc compiler). All computational results were obtained running CTS on a Bull NovaScale R422 server (2.83 Ghz quad-core Intel® Xeon® E5440 processor, 8 Gb RAM).

[6] They are available e.g. from the "PackLib^2" benchmarks library, see http://www.ibr.cs.tu-bs.de/alg/packlib/xml/ht-eimhh-01-xml.shtml

4.2 Computational Results

CTS is compared in Table 1 with the previously reported TS algorithms, denoted as TS1 [11], TS2 [10], and TS3 [9], and the best performing approaches: GRASP [5] and IDW [6].[7]

In Table 1, "–" marks and the absence of $\overline{\gamma}$ or γ^* values for TS1, TS2, and IDW mean either that $\overline{\gamma}$ or γ^* cannot be computed or that the information is not given in [6,10,11].

Table 1. Mean and best percentage gaps ($\overline{\gamma}$ and γ^* resp.) on instances from [19]

Instances				CTS		TS1 [11]	TS2 [10]	TS3 [9]		GRASP [5]		IDW [6]
Cat.	W	n	H_{OPT}	$\overline{\gamma}$	γ^*	γ^*	γ^*	$\overline{\gamma}$	γ^*	$\overline{\gamma}$	γ^*	$\overline{\gamma}$
C1	20	16–17	20	0	0	7.65	0	1.59	0	0	0	0
C2	40	25	15	0	0	2.08	0	0	0	0	0	0
C3	60	28–29	30	0.87	0	7.14	0	1.08	1.08	1.08	1.08	2.15
C4	60	49	60	1.64	1.64	4.75	–	1.64	1.64	1.64	1.64	1.09
C5	60	73	90	1.74	1.46	4.91	–	1.1	1.1	1.1	1.1	0.73
C6	80	97	120	2.17	1.91	3.74	–	1.37	0.83	1.56	0.83	0.83
C7	160	196–197	240	2.15	2.04	–	–	1.23	1.23	1.36	1.23	0.41

According to Table 1, TS1 is the worst performing (TS) approach for the benchmark tried. Indeed, all other approaches (except TS1) solved the C1 and C2 instances, see lines C1–C2 where $\gamma^* = 0$ or $\overline{\gamma} = 0$.

To our knowledge, only TS2 (and the exact algorithm from [14]) solved **all** the 9 smallest instances (C1–C3). CTS is the first method reaching the same qualitative results, see lines C1–C3 where γ^* is **always** 0 just for TS2 and CTS. Furthermore, note that CTS achieves here the lowest $\overline{\gamma}$ values compared with GRASP, IDW, and TS3.

CTS compares also well with the competitors if one considers the 3 instances from category C4. Indeed, line C4 indicates the same γ^* values (1.64) for CTS, TS3, and GRASP.

CTS obtains worst γ^* or $\overline{\gamma}$ values than those of the best-known approaches (GRASP and IDW) on the largest 3 categories of instances (C5–C7).

5 Conclusions

In this paper, we presented CTS, a Consistent Tabu Search algorithm for a 2D Strip Packing Problem. CTS treats the initial 2D-SPP *optimization* problem (minimizing the height H) as a succession of 2D-SPP$_{k>0}$ *satisfaction* problems: Is there a solution s to 2D-SPP such that $H(s) \leq k$? Starting from a complete packing s_0, CTS tackles 2D-SPP$_k$ with decreasing values of $H(s_0)$ for k.

[7] For indicative purpose, the mean running time of CTS ranges from less than a second (for the smallest instances) to about 33 hours (for the largest instances). The mean computation time of the competing methods varies from less than a second to about 45 minutes.

The key features of CTS include a direct representation of the search space which permits inexpensive basic operations, a consistent neighborhood, a fitness function including problem knowledge, and a diversification based on the history of the search. The performance of CTS was assessed on a set of 21 well-known hard instances. The computational experiments showed that CTS is able to reach the optimal values for the first 9 problem instances (categories C1–C3) and to match the best results for the next 3 instances (C4). Nevertheless, CTS does not compete well with the best performing algorithms on the largest problems (C5–C7), which constitutes the topic for future investigations.

Acknowledgments. This work was partially supported by three grants from the French "Pays de la Loire" region (MILES, RadaPop, and LigeRO projects). The first author is supported by a Chilean CONICIT scholarship. We would like to thank the reviewers of the paper for their useful comments.

References

1. Dowsland, K., Dowsland, W.: Packing Problems. Eur. J. Oper. Res. 56(1), 2–14 (1992)
2. Fowler, R., Paterson, M., Tanimoto, S.: Optimal Packing and Covering in the Plane are NP-Complete. Inf. Process. Lett. 12(3), 133–137 (1981)
3. Wäscher, G., Haußner, H., Schumann, H.: An Improved Typology of Cutting and Packing Problems. Eur. J. Oper. Res. 183(3), 1109–1130 (2007)
4. Garey, M., Johnson, D.: Computers and Intractability – A Guide to the Theory of NP-Completness. W.H. Freeman and Company, San Francisco (1979)
5. Alvarez-Valdes, R., Parreño, F., Tamarit, J.: Reactive GRASP for the Strip-Packing Problem. Comput. Oper. Res. 35(4), 1065–1083 (2008)
6. Neveu, B., Trombettoni, G.: Strip Packing Based on Local Search and a Randomized Best-Fit. In: 5th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems – 1st Workshop on Bin Packing and Placement Constraints (2008)
7. Neveu, B., Trombettoni, G., Araya, I.: Incremental Move for Strip-Packing. In: Avouris, N., Bourbakis, N., Hatzilygeroudis, I. (eds.) ICTAI 2007, vol. 2, pp. 489–496. IEEE Computer Society, Los Alamitos (2007)
8. Zhang, D., Liu, Y., Chen, S., Xie, X.: A Meta-Heuristic Algorithm for the Strip Rectangular Packing Problem. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 1235–1241. Springer, Heidelberg (2005)
9. Hamiez, J.P., Robet, J., Hao, J.K.: A Tabu Search Algorithm with Direct Representation for Strip Packing. In: Cotta, C., Cowling, P. (eds.) EvoCOP 2009. LNCS, vol. 5482, pp. 61–72. Springer, Heidelberg (2009)
10. Alvarez-Valdes, R., Parreño, F., Tamarit, J.: A Tabu Search Algorithm for a Two-Dimensional Non-Guillotine Cutting Problem. Eur. J. Oper. Res. 183(3), 1167–1182 (2007)
11. Iori, M., Martello, S., Monaci, M.: Metaheuristic Algorithms for the Strip Packing Problem. In: Pardalos, P., Korotkikh, V. (eds.) Optimization and Industry – New Frontiers. Appl. Optim., vol. 78, pp. 159–179. Springer, Heidelberg (2003)
12. Bortfeldt, A.: A Genetic Algorithm for the Two-Dimensional Strip Packing Problem with Rectangular Pieces. Eur. J. Oper. Res. 172(3), 814–837 (2006)

13. Araya, I., Neveu, B., Riff, M.C.: An Efficient Hyperheuristic for Strip-Packing Problems. In: Cotta, C., Sevaux, M., Sörensen, K. (eds.) Adaptive and Multilevel Metaheuristics. Stud. Comput. Intell, vol. 136, pp. 61–76. Springer, Heidelberg (2008)
14. Kenmochi, M., Imamichi, T., Nonobe, K., Yagiura, M., Nagamochi, H.: Exact Algorithms for the Two-Dimensional Strip Packing Problem with and without Rotations. Eur. J. Oper. Res. 198(1), 73–83 (2009)
15. Martello, S., Monaci, M., Vigo, D.: An Exact Approach to the Strip Packing Problem. INFORMS J. Comput. 15(3), 310–319 (2003)
16. Glover, F., Laguna, M.: Tabu Search. Kluwer, Dordrecht (1997)
17. Baker, B., Coffman Jr., E., Rivest, R.: Orthogonal Packings in Two Dimensions. SIAM J. Comput. 9(4), 846–855 (1980)
18. El Hayek, J.: Le Problème de Bin-Packing en Deux-Dimensions, le Cas Non-Orienté : Résolution Approchée et Bornes Inférieures (in French). PhD thesis, Université de Technologie de Compiègne, France (2006)
19. Hopper, E., Turton, B.: An Empirical Investigation of Meta-Heuristic and Heuristic Algorithms for a 2D Packing Problem. Eur. J. Oper. Res. 128(1), 34–57 (2001)
20. Chazelle, B.: The Bottom-Left Bin-Packing Heuristic – An Efficient Implementation. IEEE Trans. Comput. 32(8), 697–707 (1983)
21. Imahori, S., Yagiura, M., Nagamochi, H.: Practical Algorithms for Two-Dimensional Packing. In: Gonzalez, T. (ed.) Handbook of Approximation Algorithms and Metaheuristics. Chapman & Hall / CRC Comput. & Inf. Sc. Ser, vol. 13, ch. 36. CRC Press, Boca Raton (2007)

Transition State Layer in the Immune Inspired Optimizer

Konrad Wojdan[1], Konrad Swirski[1], and Michal Warchol[2]

[1] Warsaw University of Technology, Institute of Heat Engineering, Poland
[2] Warsaw University of Technology, Institute of Control and Computational Engineering, Poland

Abstract. The SILO (Stochastic Immune Layer Optimizer) system is a novel, immune inspired solution for an on-line optimization of a large-scale industrial processes. Three layers of optimization algorithm were presented in previous papers. Each layer represents a different strategy of steady state optimization of the process. New layer of the optimization algorithm is presented in this paper. The new *Transition State* layer is responsible for efficient operation of the optimization system during essential process state transitions. New results from SILO implementation in South Korean power plant are presented. They confirm high efficiency of the SILO optimizer in solving technical problems.

Keywords: Steady State Control, Adaptive Control, Industrial Process Optimization, Artificial Immune Systems.

1 Introduction

The SILO (Stochastic Immune Layer Optimizer) system is a tool for an on-line optimization of a large-scale industrial processes. It was described in [1,2]. Depending on the form of the quality indicator, this system performs an on-line economic optimization of a process operating point (the economic form of a quality indicator [2]) or an advanced steady state control of an industrial process (the control form of a quality indicator). The role of SILO optimizer in the plant control structure is presented in figure 1. Until now the SILO optimizer has been used for optimization of combustion process in a large-scale power boilers. In these implementations SILO was responsible for minimization of output signals deviations from setpoints. Thus the control form of the quality indicator is analyzed in this paper.

The operation and structure of SILO optimizer is inspired by the immune system of living creatures. From the engineering point of view the immune system is a disturbed information processing system. It can gather and develop qualifications during learning and testing process. It is characterized by efficient adaptation to external conditions. These features are very important in advanced control solutions. In previous works the immune analogy has been presented [1]. The immune terminology was used to describe the SILO structure and algorithms. Some of reviewers have complained that such description is not easy to

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 11–20, 2010.

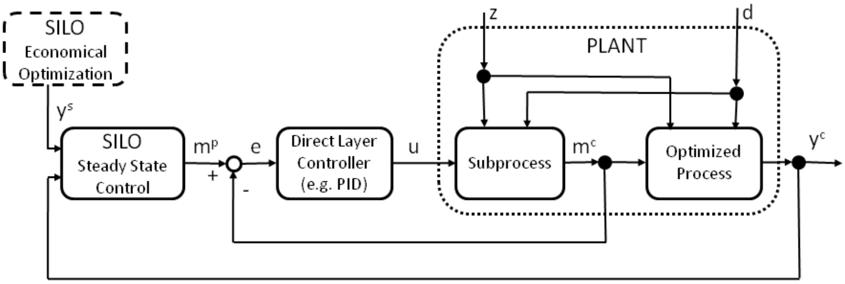


Fig. 1. The role of SILO optimizer in the control structure of a plant. Description: y^s - vector of demand values of controlled outputs, y^c - vector of current, process values of controlled outputs, m^p - control vector calculated by SILO, that represents demand values for controllers implemented in base control layer, m^c - current, process values of control vector elements, d - vector of measured disturbances, z - vector of not measured disturbances.

Table 1. Analogy between immune system and SILO

Immune System	SILO
Pathogen	Measured and non-measured disturbances
B cell	A memory brick that represents static process response for a control change in a particular process operating point
Antibody: antigen binding side	Current process operating point (current values of m, y and d vectors), stored in a B cell
Antibody: efector part	Optimal control vector change
T$_h$ cell	Algorithm which is responsible for selection of proper group of B cells during model creation process
Primary immune response	Model based layers of the optimization algorithm
Secondary immune response	*Quasi-Random Extremum Control* layer of the optimization algorithm

understand for readers who are not familiar with operation of the immune system. In this paper the structure and operation of SILO optimizer is explained without utilization of biological terminology. In table 1 the immune analogy is briefly introduced to keep the compatibility with previous works.

The SILO optimizer is a low cost alternative for solutions that are based on *a priori* defined process model. In the presented solution there is no need to perform long lasting and labour consuming identification experiments *explicite*. Thus the implementation cost is essentially reduced and there is no need for modification of a production schedule of a plant. In the SILO system the optimization strategy depends on a SILO knowledge of the process. Presented solution automatically gathers portions of knowledge about the process. If there is enough such knowledge bricks it uses selected portions of knowledge to forecast plant behavior in the close neighborhood of a current process operating point. The

SILO system improves its efficiency over time. It approximates non-linear static process characteristic more directly than other model based solutions, because it is not limited by the cost of model creation process. Moreover it is able to adapt to time-varying changes of the process characteristics related with the wearing down or failure of devices, changes in chemical properties of components used in a process (e.g. fuel properties), plant modernizations or external condition changes (e.g. seasonality).

One should noticed that SILO uses only information about static dependences between process inputs and outputs. Thus this solution is dedicated only for such processes in which the transition of the process state can be ignored in the optimization task. Such processes are characterized by fast but rare or continuous but slow (in comparison with plant dynamic) disturbance changes. The whole responsibility for process state transition is taken by base control layer. Practical observations of SILO operation in 13 power units (the smallest unit is 200 MW, the largest is 890 MW) confirm that this solution is a good, low-cost alternative for MPC (Model Predictive Control) controllers [3] for certain class of industrial processes (i.e.: combustion process in a large scale power plants). The relatively slow reaction time for an essential disturbance change is a main SILO disadvantage. This solution reaches the highest efficiency in the steady state. In this paper the new sub-layer of an optimization task is introduced. It is responsible for faster reaction for essential disturbances changes. It allows SILO optimizer to obtain better results for regulating units.

2 Knowledge Gathering Process

SILO consists of two main, independent modules – an optimization module and a knowledge gathering module. The operation of both modules is wider discussed in [2]. The knowledge gathering module performs on-line analysis of current and historical values of elements of m, y and d vectors. It searches for a time windows which fits to a special, pre-defined pattern (refer fig. 2). The time window is transformed to a memory brick if all the following conditions are satisfied:

- Disturbances are constant in the whole time window (assuming tolerance for signal noise and oscillations);
- There is at least one significant change of control signal in the assumed moment of a time window;
- The process state at the beginning and at the end of a time window is steady.

The memory brick stores the information about historical, static process response to a control change. The formal definition of a k-th memory brick is as follows:

$$L_k = \left[{}^b\overline{m}^k, {}^p\overline{m}^k, {}^b\overline{y}^k, {}^p\overline{y}^k, \overline{d}^k \right],$$

where: ${}^b\bar{m}^k$ – average values of control signals measured before control change, ${}^p\bar{m}^k$ – average values of control signals measured after control change, ${}^b\bar{y}^k$ – average values of process outputs measured before control change, ${}^p\bar{y}^k$ – average

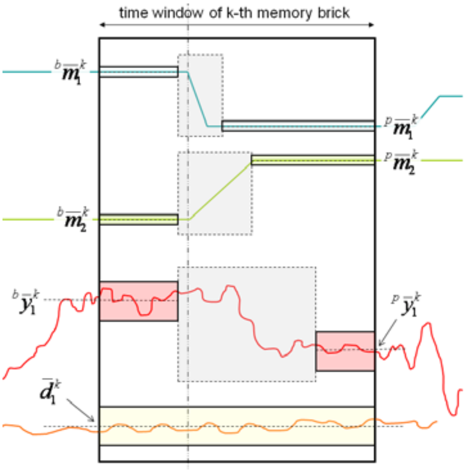


Fig. 2. Time window of the k-th memory brick

values of process outputs measured after control change, $\bar{d}^k$ – average values of measured disturbances.

During normal operation of a plant there are frequent changes of control inputs. Thus, the process of creating memory bricks is continuous. SILO updates its knowledge about the process all the time. It ensures constant adaptation to variable operation conditions. Control changes can be caused by an operator, a base control layer or by an optimization module of SILO.

3 The Optimization Algorithm

The goal of the optimization module is to minimize the following quality indicator

$$J = \sum_{k=1}^{nx} \left[\alpha_k \left(|m_k^c - m_k^s| - \tau_k^{lm} \right)_+ + \beta_k \left((|m_k^c - m_k^s| - \tau_k^{sm})_+ \right)^2 \right] +$$
$$+ \sum_{k=1}^{ny} \left[\gamma_k \left(|y_k^c - y_k^s| - \tau_k^{ly} \right)_+ + \delta_k \left((|y_k^c - y_k^s| - \tau_k^{sy})_+ \right)^2 \right] \qquad (1)$$

where:

α_k – linear penalty coefficient for k-th control variable,
β_k – square penalty coefficient for k-th control variable,
γ_k – linear penalty coefficient for k-th optimized output,
δ_k – square penalty coefficient for k-th optimized output,
τ_k^{lm} – width of insensitivity zone for linear part of penalty for k-th control variable,

τ_k^{sm} – width of insensitivity zone for square part of penalty for k-th control variable,
τ_k^{ly} – width of insensitivity zone for linear part of penalty for k-th output,
τ_k^{sy} – width of insensitivity zone for square part of penalty for k-th output,
$(\cdot)_+$ – "positive part" operator $(x)_+ = \frac{1}{2}(x + |x|)$,
m_k^c – current, process value for k-th control variable,
y_k^c – current, process value for k-th optimized output,
m_k^s – demand value for k-th control variable,
y_k^s – demand value for k-th optimized output.

Optimization method is discussed in [2]. The optimization period is defined as the time range between control vector changes. This time is not shorter than time needed to reach a steady state of process outputs, after a control change. In case of a combustion process in a large scale power boiler, this time period varies from 5 to 15 minutes. The optimization algorithm operates in one of four optimization layers. In each layer a different optimization strategy is used to compute an optimal control vector increment that minimizes a value of a quality indicator. An algorithm that lets SILO switches between layers is a key algorithm of this solution (refer fig. 3). Activation of particular layer depends on SILO knowledge and current process state.

The *Quasi-Random Extremum Control* layer performs quasi-random exploration of solution space. It is being executed when SILO has no knowledge about the process or when the knowledge about the process is insufficient. The solution found in this layer is a starting point for optimization in layers that use a process model. In the single cycle of this algorithm, one subset of control variables is randomly chosen. SILO changes only one control variable from this subset in a single optimization period. Thus each control variable change represents a classical identification experiment that may be transformed to a memory brick. Utilization of extremum control approach [4] allows SILO to minimize a value of a quality indicator in a long period of time.

The *Model Based Optimization* layer consists of *Mixed Model Based Optimization* and *Global Model Based Optimization* layer. The optimal increment of control vector that minimizes the value of a quality indicator is calculated by solving the quadratic programming task with linear constraints related with the range and maximal increment of control variables. The prediction of the process output signals is based on an incremental, linear model

$$\Delta y = \Delta m K, \tag{2}$$

where K is a gain matrix. This model is automatically created in each optimization period. In the *Global Model Based Optimization* layer all the newest memory bricks related with different operating points (global memory bricks) are used to identify the gain matrix K. The model used in *Mixed Model Based Optimization* is more accurate. In the *Mixed Model Based Optimization* layer the K matrix is identified mainly based on the newest local memory bricks that represent static process dependences in the close neighborhood of current process

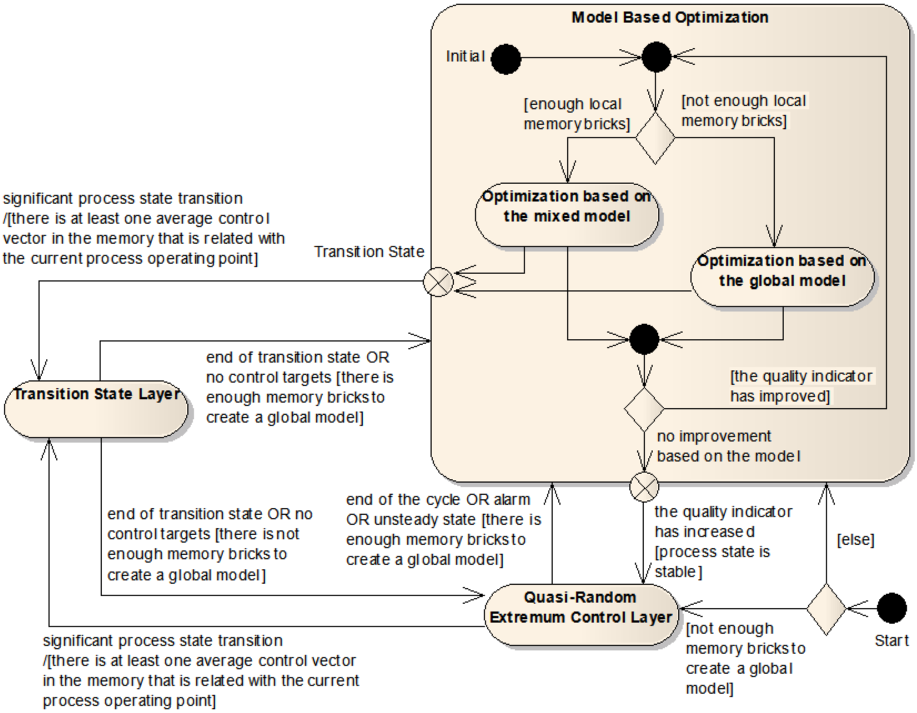

Model Based Optimization
Initial
[enough local memory bricks]
[not enough local memory bricks]
Optimization based on the mixed model
Optimization based on the global model
Transition State
significant process state transition /[there is at least one average control vector in the memory that is related with the current process operating point]
Transition State Layer
end of transition state OR no control targets [there is enough memory bricks to create a global model]
[the quality indicator has improved]
no improvement based on the model
end of transition state OR no control targets [there is not enough memory bricks to create a global model]
end of the cycle OR alarm OR unsteady state [there is enough memory bricks to create a global model]
the quality indicator has increased [process state is stable]
[else]
Quasi-Random Extremum Control Layer
[not enough memory bricks to create a global model]
Start
significant process state transition /[there is at least one average control vector in the memory that is related with the current process operating point]

Fig. 3. Part of the penalty function related with one output signal

operating point. Automatically created model represents the linear approximation of nonlinear static process characteristics in the current process operating point. There are several mechanisms that are responsible for maintaining good conditioning of model identification task. These methods are discussed in [2].

The new *Transition State* layer is being executed in case of a significant transition of process state, caused by a significant change of a disturbance vector. The algorithm implemented in this layer is responsible for safe transition of control vector from the current value to the new, automatically selected demand value, with consideration of constraints related with control increment. The new target value of control vector is a new starting point for optimization algorithm in the new process operating point.

4 The New Transition State Layer

In the previous versions of SILO there was no Transition State layer. Because of this, after a significant process operating point transition, the last solution from previous operating point was a starting point for optimization task in the new operating point. In some situations the transition of an optimal solution was significant and SILO system needed relatively much time to reach the new optimal solution. The new Transition State layer assures faster SILO reaction for a significant disturbance change. Optimization module of SILO executes the Transition State layer if two conditions are satisfied. The first condition is a detection of a significant disturbance change. The second condition is an existence of at least one special vector of average control variables values, related with the value of disturbance vector that is similar to the current value of a disturbance vector (please refer [3]);

In the new Transition State layer the SILO system searches for a new target value of a control vector, that is related with the current value of a disturbance vector. The optimization module transfers the current solution to a new target solution, using a linear trajectory that satisfy the ramp constraints.

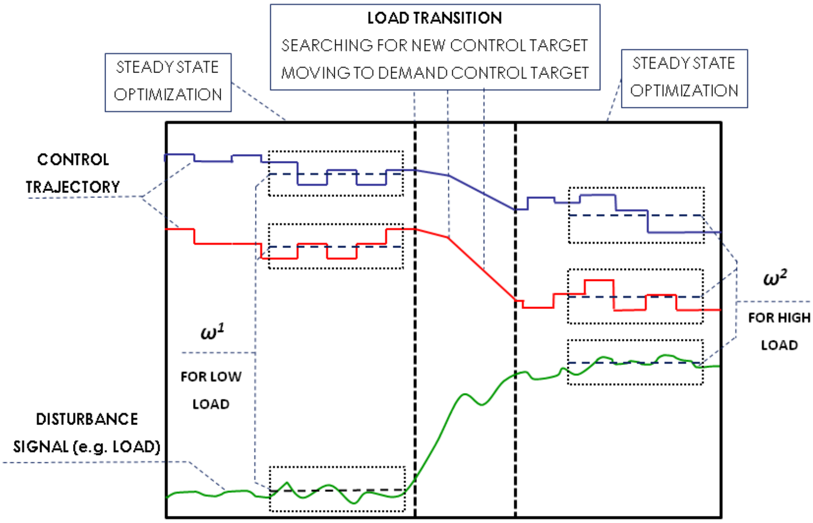

LOAD TRANSITION
SEARCHING FOR NEW CONTROL TARGET
MOVING TO DEMAND CONTROL TARGET
STEADY STATE OPTIMIZATION
STEADY STATE OPTIMIZATION
CONTROL TRAJECTORY
ω^1
FOR LOW LOAD
ω^2
FOR HIGH LOAD
DISTURBANCE SIGNAL (e.g. LOAD)

Fig. 4. The operation of a module that calculates long term averages of control, output and disturbance vectors

In the SILO system there is an additional module that calculates in on-line mode long term averages of control, disturbance and output vectors in the current process operating point (refer fig. [4]). Such information, in conjunction with the timestamp, is stored in the separate knowledge base. The formal expression of such vector is

$$\omega = \left[\bar{m}, \bar{y}, \bar{d}, \tau\right], \tag{3}$$

where τ is a timestamp. This additional module has some mechanisms that protect the knowledge base against redundant information and keep the rational size of this base.

The evaluation, if a k-th ω vector corresponds with the current disturbance vector, is based on locality condition function:

$$\mu\left(\bar{d}^{\omega_k}, d^a\right) = \prod_{i=1}^{nd} g_i\left(\bar{d}_i^{\omega_k}, d_i^a\right) \;,\;\; \forall_{x_1, x_2 \in \Re} \;\; g\left(x_1, x_2\right) \in \{0, 1\}$$

where: d^a denotes the current value of a disturbance vector and $\bar{d}^{\omega_k}$ denotes the vector of average values of disturbances stored in k-th vector ω. Locality conditions are satisfied if $\mu = 1$. Examples of functions used in locality conditions are presented below:

$$\text{example 1: } g\left(x_1, x_2\right) = \begin{cases} 0 \text{ if } \left|x_1 - x_2\right| > \varepsilon \\ 1 \text{ if } \left|x_1 - x_2\right| \le \varepsilon \end{cases} \text{, example 2: } g\left(x_1, x_2\right) \equiv 1 \;.$$

Sometimes utilization of some expert knowledge (e.g. from process operators) can be useful. In SILO system there is a possibility to define *explicite* vectors of initial solutions (initial vector of control variables) for defined operating points

$$s_d = \left[\xi, \rho, m^d, c\left(d\right)\right] ,$$

where: ξ - importance flag (it can be true or false), ρ - priority level, m^d - the initial control vector value, $c\left(d\right) = \left[c_1\left(d_1\right), c_2\left(d_2\right), \ldots, c_n d\left(d_n d\right)\right]$ - vector of affinity conditions. The initial solution is related with the current disturbance vector if all affinity conditions are satisfied. Examples of such conditions are presented below:

$$\text{example 1: } c\left(d_k\right) = \begin{cases} 1 \text{ if } d_k^l \le d_k \le d_k^u \\ 0 \text{ otherwise} \end{cases} \text{, example 2: } c\left(d_k\right) \equiv 1 \;.$$

In each step of the algorithm implemented in the Transition State layer, one target control variable vector is chosen from the set of automatically saved (ω vectors) and *explicite* defined (s_d) control variable vectors. The algorithm of selection of proper target control variable vector is presented in fig. 5.

The optimization module exits the Transition State layer if a disturbance value reaches a new steady state or there will be no target control variable vectors related with the current disturbance vector.

5 Results

Until now the SILO optimizer has been implemented in 13 power plants in USA, Poland, South Korea and Taiwan. Results of a SILO implementation in optimization of a combustion process in one of South Korean power boilers are presented below. The maximum load of this unit is 890 MW. The combustion process takes place in a supercritical once through boiler. Hard coal is a fuel for this boiler. The SILO goal was to decrease NO$_x$ emission below 140 ppm limit, without deterioration of unit efficiency. The process output vector consisted of 8

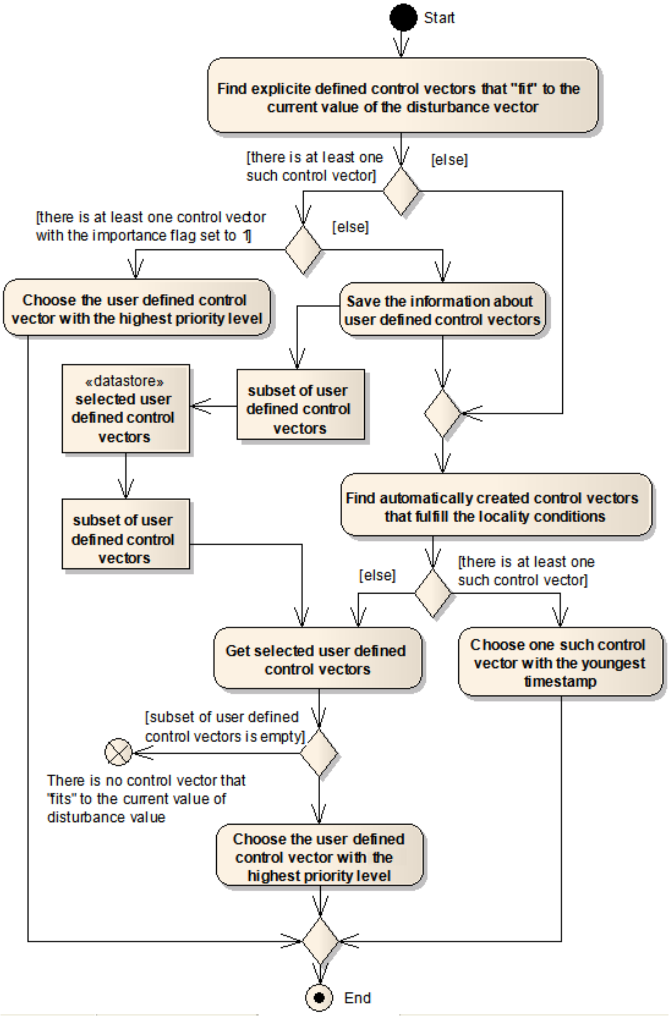

Start
Find explicite defined control vectors that "fit" to the current value of the disturbance vector
[there is at least one such control vector]
[else]
[there is at least one control vector with the importance flag set to 1]
[else]
Choose the user defined control vector with the highest priority level
Save the information about user defined control vectors
«datastore» selected user defined control vectors
subset of user defined control vectors
subset of user defined control vectors
Find automatically created control vectors that fulfill the locality conditions
[else]
[there is at least one such control vector]
Get selected user defined control vectors
Choose one such control vector with the youngest timestamp
[subset of user defined control vectors is empty]
There is no control vector that "fits" to the current value of disturbance value
Choose the user defined control vector with the highest priority level
End

Fig. 5. The selection of proper target control variable vactor

elements: average NO_x emission, average CO emission, 2 measurements of flue gas temperature (side A and B), 2 measurements of reheat steam temperature (side A and B), 2 measurements of superheat steam temperature (side A and B). The control variables vector consisted of 10 elements: O_2 level in exhaust fumes, windbox to furnace differential pressure, CCOFA (Close Coupled Over Fire Air) dampers, auxiliary air dampers, 4 control variables related with SOFA (Separated Over Fire Air) dampers and 2 control variables related with SOFA tilts. Disturbance vector consisted of 11 elements: unit load, unit load to fuel flow relation, burner tilts demand, status of sootblowers, 6 disturbance signals related with status of each coal pulverizer.

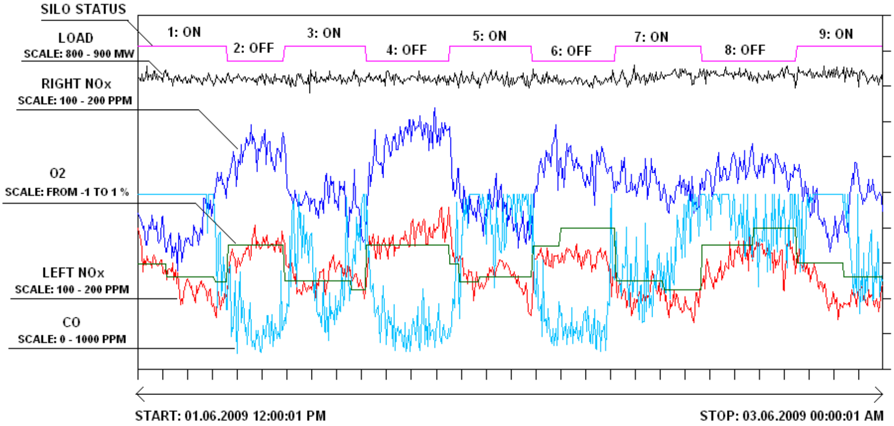


Fig. 6. The SILO influence on NOx and CO emission

SILO was able to reduce NOx emission by 9% and keep it below 140 ppm (refer fig. 6). The unit efficiency has not been decreased. Losses related with higher CO emission and higher LOI (loss of ignition) was compensated by lower O_2 level and by maintaining steam temperatures closer to the setpoint.

6 Future Work

Currently development of the new version of algorithm for automatic model identification is in progress. This algorithm will consider the box constraints related with model gains. The least square method in model identification task will be replaced by the task of quadratic programming with constraints.

References

1. Wojdan, K., Swirski, K.: Immune Inspired Optimizer of Combustion Process in Power Boiler. In: Okuno, H.G., Ali, M. (eds.) IEA/AIE 2007. LNCS (LNAI), vol. 4570, pp. 1022–1031. Springer, Heidelberg (2007)
2. Wojdan, K., Swirski, K., Warchol, M., Maciorowski, M.: Maintaining Good Conditioning of Model Identification Task in Immune Inspired On-line Optimizer of an Industrial Process. Engineering Letters 17(2), 93–100 (2009)
3. Tatjewski, P.: Advanced Control of Industrial Processes: Structures and Agloriithms. Springer, London (2007)
4. Jacobs, O.L.R., Wonham, W.M.: Extremum Control in the Presence of Noise. International Journal of Electronics 11(3), 193–211 (1961)

Improving Timetable Quality in Scheduled Transit Networks

Valérie Guihaire[1,2] and Jin-Kao Hao[2]

[1] HEURISIS, 8 rue Le Notre, 49100 Angers, France
vguihaire@perinfo.eu
[2] LERIA, Université d'Angers, 2 boulevard Lavoisier, 49045 Angers, France
hao@info.univ-angers.fr

Abstract. This work deals with an original problem with regard to the traditionally sequential planning process in public transit networks. This problem aims at modifying the network's timetables without rendering the vehicle and driver schedules obsolete. The objective is to improve the quality of service for passengers through number and quality of transfers. This approach goes in the opposite direction compared to the usual approach which schedules resources once timetables are set. We propose a model and a solution method based on tabu search and a neighborhood specifically developed. Experiments are led on five instances related to a real transit network. Important gains are obtained on the considered case study, allowing for better mobility of users inside the network and on the intermodal level.

Keywords: Transit network, synchronization, timetabling.

1 Introduction

We address the problem of modifying the timetables of a running (timetabled and scheduled) transit network to improve its quality of service through transfers while maintaining the vehicle and driver trip assignments. This problem is met by transit planners in practice but is counter–intuitive regarding the treatment of transit planning in the literature, according to which lines timetabling is operated first and resource scheduling is only operated next [1]. We call this problem Schedules–based re–Timetabling (SbrT) in the rest of this paper.

In a running transit network, i.e. in which all the planning steps have been completed, the context and environment continue to evolve. For instance, changes in the demand or in the offer of coordinated modes can occur, leading to a situation in which changes in the timetable could benefit to the quality of service. The traditional process would be to define and apply changes to the timetables and then restart the resource planning process. However, planning vehicle and driver schedules is an extremely complex task that needs to take into account a great number of constraints, objectives and parameters. This can be too demanding a task to even consider applying timetable changes in the first place. Therefore, to deal with this problem, we propose a model and a method to boost

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 21–30, 2010.

the service quality of transit networks while leaving unchanged the sequences of trips assigned to vehicles and drivers.

With regard to our problem here, one study is noteworthy, although the SbrT is not its specific purpose. In [4], Jansen et al. propose to improve the quality and number of transfers in a transit network through shifts in the line starting times. This implies that all the runs of a given line are shifted by the same value and headways (frequencies) are fixed. It takes no account of resource usage, implying that this method is rather directed toward urban transit networks, and assumes that interlining is not practiced, so that vehicle allocation is not impacted by the changes. In extra–urban networks, the risk is to obtain a timetable that is not compatible with the current vehicle assignments or worse, with the available number of resources. Another approach to reorganize the timetables without requiring additional vehicles can be to include minimizing the fleet size to the objectives [6]. However, the risk is high that the resource schedules will need a total reconstruction, thus not answering our problem here.

2 Problem Description

The problem consists in slightly modifying the timetables so as to improve both quantity and quality of the transfer opportunities, while respecting the trip sequences assigned to vehicles as well as to drivers. Results for this problem are new timetables for the lines.

In order to better understand the scheduling part of the planning process, let us first present the constitution of a resource trip assignment. Such an assignment (see Figure 1) is a sequence of trips, deadheads (to and from the depot as well as between trips), turnaround time and stopping or pausing time according to the type of resource (vehicle or driver).

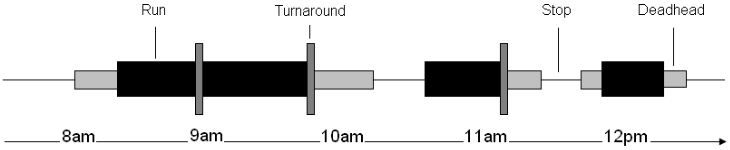


Fig. 1. Vehicle trip assignment

2.1 Objective: Quality and Quantity of Transfer Opportunities

The objective consists in creating numerous and high–quality transfer opportunities for users. A non–captive users policy is applied, meaning a transfer opportunity happens only when the transfer waiting time is inside a predefined interval. A minimal, ideal and maximal waiting time are all specified for each kind of transfer, as well as a level of importance. The quality of the transfer opportunity is proportional to the closeness with the ideal waiting time. Transfers among lines of the network as well as with external modes are considered here.

2.2 Constraints

A set of constraints applies to the timetable:

Maximal Shift. The current timetable has very likely been designed to comply with the other main criterion for quality of service: headways. Headway determination permits to provide a regular service for one, and second, to adapt to demand peaks and prevent overcrowding. In order to preserve this adaptation to the demand fluctuations, we only allow a shifting value comprised inside a definite interval.

Driver Trip Assignment. The existing driver assignment must be compatible with the new timetable. Concretely, each driver must be able to serve the same sequence of line runs, turnarounds, deadheads and pausing times, without any of these elements overlapping. In terms of duration, only pausing times may vary. To respect social issues linked to pause duration while introducing some flexibility in the model, we allow each waiting time to vary inside a pre–defined interval. The human planner thus provides a list of intervals (in minutes) as data for the problem. For instance, if the list is: [0-15];[16-30];[31-45];[46-N], then the duration of an initial 28-minute long pause may vary between 16 and 30 minutes in the final solution.

Vehicle Trip Assignment. The existing vehicle assignment must be compatible with the new timetable. Concretely, each vehicle must be able to serve the same sequence of line runs, turnarounds, deadheads and stopping times, without any of these elements overlapping. In terms of duration, only stopping times may vary.

3 Problem Formulation

We define a model allowing for a high level of flexibility, in which the set of decision variables matches the starting time of all the runs in the network. Additionally, a set of state variables is kept up to date to facilitate the computation of costs linked to transfers. This latter set includes the stopping times of the runs at intermediate stops along the runs. To compute the value of these variables, we consider fixed running times by runs along the search process. This is acceptable due to the short shifting span allowed for each starting time.

3.1 Notations

Let $\mathcal{L}$ be the set of lines, $\mathcal{R}$ the set of runs, $\mathcal{S}$ the set of stops, $\mathcal{V}$ the set of vehicles, $\mathcal{D}$ the set of drivers, $\mathcal{T}$ the set of transfers, and $\mathcal{H}$ the planning horizon.

- For each line $l \in \mathcal{L}$:
  - $\mathcal{R}_l \subset \mathcal{R}$: set of runs served.
- For each run $r \in \mathcal{R}$:
  - $\mathcal{S}_r \subset \mathcal{S}$: set of served stops.
  - $s_r^{\alpha}, s_r^{\omega}$: first and last stop.

  - $h^{\leftarrow}_{r,s}, h^{\rightarrow}_{r,s}$: arriving and leaving time for $s \in \mathcal{S}_r$ in the initial timetable.
  - $h(r)$: initial starting time for run r $(h(r) = h^{\rightarrow}_{r,s^{\alpha}_{r}})$.
  - v_r: vehicle assigned to r.
  - d_r: driver assigned to r.
  - r^{+}_{v}: run following r in the schedule of vehicle $v = v_r$.
  - r^{+}_{d}: run following r in the schedule of driver $d = d_r$.
- For each transfer $t = (l_1, s_1, l_2, s_2) \in \mathcal{T}$ taking place at stops $s_1, s_2 \in \mathcal{S}$ on lines $l_1, l_2 \in \mathcal{L}$:
  - il_t: level of importance
  - $\lfloor wt_t \rfloor, [wt_t], \lceil wt_t \rceil$: minimal, ideal and maximal transfer waiting time.
- $ttv_{r,r'}$ and $ttd_{r,r'}$: turnaround time between two consecutive runs r and r' inside a vehicle and a driver trip assignment respectively.
- $dhv_{r,r'}$ and $dhd_{r,r'}$: deadhead time between two consecutive runs r and r' inside a vehicle and a driver trip assignment respectively. This deadhead can be direct between two runs, or divided in two parts when a stop at the depot is involved.
- $\lfloor dw_{r,r'} \rfloor$ and $\lceil dw_{r,r'} \rceil$: minimal and maximal duration for the pausing time between two consecutive line runs inside a driver trip assignment. It is deduced from the initial pausing time $(h(r') - h^{\leftarrow}_{r,s^{\omega}_{r}} - ttd_{r,r'} - dhd_{r,r'})$, and from the intervals for pause durations (Section 2.2).
- $\lceil S \rceil$: maximal time shift allowed for each run starting time.

Decision Variables and Values. The decision variables are the starting times $\pi(r) \in \mathcal{H}$ of each run $r \in \mathcal{R}$.

The set of state variables include the arriving and departing times of runs at stops along the routes. Let $\pi^{\leftarrow}_{r,s} \in \mathcal{H}$ and $\pi^{\rightarrow}_{r,s} \in \mathcal{H}, \forall r \in \mathcal{R}, \forall s \in \mathcal{S}_r$ represent these state variables. The decision variable $\pi(r)$ corresponds to state variable $\pi^{\rightarrow}_{r,s^{\alpha}_{r}}$.

A *configuration* σ is a complete assignment of values in $\mathcal{H}$ to the set of decision variables.

3.2 Objective: Quality and Quantity of Transfer Opportunities

The cost function relative to transfers is a nonlinear function of the waiting time and favors the most heavily close-to-ideal waiting times. The cost incurred to the configuration is also weighed by the relative level of importance of the transfer. We compute the transfer cost for each couple of run for the arriving line and run for the departing line. Each gap belonging to the allowed interval means a transfer opportunity and generates an addition to the cost of the configuration. This is in the context of a maximization problem.

$$f(\sigma) = \sum_{\substack{t \in \mathcal{T} \\ t=(l_1,s_1,l_2,s_2)}} \left[il_t * \sum_{\substack{r_1 \in \mathcal{R}_{l_1} \\ r_2 \in \mathcal{R}_{l_2}}} fTr(t, \pi^{\leftarrow}_{r_1,s_1}, \pi^{\rightarrow}_{r_2,s_2}) \right]$$

where

$$fTr(t, h_1 \in \mathcal{H}, h_2 \in \mathcal{H}) = \begin{cases} \text{increasing from 0.5 to 1 if} & \lfloor wt_t \rfloor < h_2 - h_1 < [wt_t] \\ \text{decreasing from 1 to 0} \;\; \text{if} & [wt_t] < h_2 - h_1 < \lceil wt_t \rceil \\ 0 & \text{o.w.} \end{cases}$$

Only the costs related to transfers among lines of the network are presented here, but those related to intermodal transfers are computed in the same way and included in the objective function.

3.3 Constraints

Complete Assignment

$$\forall r \in \mathcal{R}, \pi(r) \in \mathcal{H} \tag{1}$$

Maximal Shift

$$\forall r \in \mathcal{R}, |\pi(r) - h(r)| \leq \lceil S \rceil \tag{2}$$

Driver Trip Assignment. Let $att(r, r') = \pi(r') - \pi^{\leftarrow}_{r,s_r^\omega} - dhd_{r,r'} - ttd_{r,r'}$ be the waiting time for the driver between consecutive runs r and r' in his service.

$$\forall d \in \mathcal{D}, \forall (r, r') \in \mathcal{R}^2, (r' = r_d^+) \Rightarrow (\lfloor dw_{r,r'} \rfloor \leq att(r, r') \leq \lceil dw_{r,r'} \rceil) \tag{3}$$

Vehicle Trip Assignment. Let $sta(r, r') = \pi(r') - \pi^{\leftarrow}_{r,s_r^\omega} - dhv_{r,r'} - ttv_{r,r'}$ be the stopping time of the vehicle between consecutive runs r and r' in its service.

$$\forall v \in \mathcal{V}, \forall (r, r') \in \mathcal{R}^2, (r' = r_v^+) \Rightarrow (sta(r, r') \geq 0) \tag{4}$$

4 Solution Approach

A Tabu Search method [3] is used. The initial solution is based on the timetables and resource (vehicle and driver) schedules from the running transit network. The search starts from this feasible solution and remains at all time inside the feasibility domain. A preprocessing phase is applied to reduce the domain of the variables. A neighborhood mechanism is then used to modify the departure times of the trips while preventing overlaps inside the resources schedules. Parameters for our Tabu Search implementation are exposed in the following sections. The output provided by the method consists in new line timetables for the transit network, with better quality with respect to the objective function, while the driver and vehicle trip sequences remain feasible.

4.1 Search Space and Preprocessing

The goal of the preprocessing phase is to reduce the domains of the variables by constraints propagation, and to compute what can be computed once and for all at the beginning of the process, so as to fasten the search.

Domain reduction - node consistency. Given the variables' initial domain $\mathcal{H}$ and the maximal shift constraint, the size of the domain of each variable can be reduced to twice the size of the maximal shift.

$$\forall r \in \mathcal{R}, \pi(r) \in [h(r) - \lceil S \rceil ; h(r) + \lceil S \rceil]$$

Preprocessing - arc consistency. In the model, constraints on the respect of the initial assignments are crucial. It is imperative to insure that none of the elements in these assignments overlaps its neighbors during the search. In order to prevent such a situation, we must consider the duration of each element of the assignments. Each of them has a fixed duration, except for stopping and pausing times. It is thus possible to compute the minimal time interval separating the starting time of a run from the starting time of the runs before and after it inside the vehicle and driver assignments. This interval constrains the domains of the variables along the search. It includes the duration of the run, the turnaround time, the deadheading time, and in the case of the driver schedule, the minimal pausing time. Additionally, with respect to the driver schedule, a maximal time interval is defined that takes into account the maximal pausing time allowed with respect to its initial value and the user-parameterized list of intervals.

4.2 Neighborhood - TripShift

We define a neighborhood mechanism, *TripShift*, to generate simple and fast moves while respecting the contraints. A move from *TripShift* modifies the value of a single variable (i.e. a trip). A neighbor solution is obtained by shifting its value by n minutes. n is chosen inside $\mathbb{Z}^*$ and to respect the contraints, so that the search remains inside the feasibility domain.

Let $mv_{TShift} = (r, n)$ be a move from this neighborhood.

Let $\sigma = (t_1, t_2, ..., t_{|\mathcal{R}|})$ be the current solution and $\sigma' = \sigma \circ mv_{TShift}$ a neighbor solution. Then, $\sigma' = (t'_1, t'_2, ..., t'_{|\mathcal{R}|})$ obeys to the following description:

$$\forall r^* \in \mathcal{R}, \begin{cases} t'_{r^*} = t_{r^*} + n \text{ if } & r^* = r \\ t'_{r^*} = t_{r^*} & \text{o.w.} \end{cases}$$

4.3 Parameters of Tabu Search for the SbrT

- The *initial solution* consists in the current line timetables, vehicle schedules and driver schedules.
- The *neighborhood* used is *TripShift*. It is fully explored at each iteration.

– For the sake of *diversification* and to prevent the search from stagnating, the selection of moves to apply is random among those of maximal cost.
– Each move remains inside the tabu list for a random number of iterations. This number, which determines the *tabu tenure* at each iteration, varies between two pre–defined bounds (Section 5.3).
– The *stop criterion* used is the computational time.

5 Experimentations and Numerical Results

5.1 Data Instances

To define data instances, we used the structure and timetables from a real extra–urban transit network, located in a French area called Loiret. The network serves three average–size cities and many small villages. It counts 50 oriented lines, 673 stops, 30 activities (trains, schools) in connection with the network, and 282 types of possible transfers. A type of transfer is a pair of lines crossing at a stop and among which a transfer possibility may thus be created. The timetable for a typical day has 318 runs. Three transport operators serve the network (on 8, 16 and 26 lines each). Concerning the transfer objective, 180 possibilities occur in the current network, for an objective function value of 947.462. Since resource usage data is confidential, we do not have access to the constitution of the vehicle and driver trip assignments defined by the transit operators. We used a classical approach to determine five sets of resource schedules, whose characteristics are presented in Table 1:

Table 1. Number of resources involved in each instance

	Instance1	Instance2	Instance3	Instance4	Instance5
Number of vehicles	91	91	91	91	91
Number of drivers	170	171	163	171	159

– the vehicle trip assignments are generated using an exact auction algorithm [2] depending on a weighing strategy; five sets of assignments are generated, with at least 30% difference among each pair of them (with respect to consecutive trips in the assignments).
– the driver trip assignments are generated based on the vehicle trip assignments using a classical bi–level driver scheduling algorithm [5].

5.2 Numerical Results

Our algorithm was coded in C++, compiled with VC++ 9.0, on a laptop equipped with Windows Vista, a 2.10 Ghz Intel(R) Core(TM)2 Duo CPU processor and 4Go RAM. For these tests, we used a computational time of 1 minute as unique stop criterion for our algorithm. A series of 30 tests was launched on each instance, given the non deterministic behavior of the tabu algorithm.

The tabu tenure is dynamic. The number of iterations during which each move remains in the list varies between the arbitrary values of 7 an 20 (see Section 5.3). The maximal allowed shift by run is 10 minutes (in each direction), a value which represents a compromise between size of the search space (which needs to be large enough to provide opportunities for improvement) and respect of the initial resources organisation (especially in terms of headway repartition).

Table 2 presents the number of transfer possibilities by data instance, in the initial timetable (row 2) and in the solution obtained by the algorithm (row 3, mean and standard deviation). Row 4 shows the percentage of improvement between the two solutions. Table 3 presents the results of the same series of tests, in terms of value of the objective function.

Table 2. Mean number (standard deviation) of transfer opportunities, and improvement compared to the initial solution

	Initial Solution	Tabu Search with TripShift	
		mean (s-d)	%
Instance 1	180	245.07 (0.25)	36.15
Instance 2		247 (0)	37.22
Instance 3		245.2 (0.41)	36.22
Instance 4		243 (0)	35.00
Instance 5		245.1 (0.31)	36.17

Table 3. Mean value (standard deviation) of the objective function, and improvement compared to the initial solution

	Initial Solution	Tabu Search with TripShift	
		mean (s-d)	%
Instance 1	947.462	1057.37 (0.12)	11.60
Instance 2		1061.63 (0)	12.05
Instance 3		1072.58 (0.08)	13.21
Instance 4		1047.14 (0)	10.52
Instance 5		1072.35 (0.06)	13.18

Those tables show that the algorithm brings a sizeable improvement to the initial solution, in terms of both criteria used. On average, it brings a 36.15% increase in the number of transfer possibilities, and a 12.11% improvement in terms of value of the objective function. This difference is explained by the additional consideration in the objective function of the quality of the new and existing transfer possibilities.

These test results show that the considered problem, going backwards in the traditional transit planning process, owns the potential for improvement in the level of quality of service. Furthermore, the proposed model offers a good flexibility for re–timetabling the network without condemning the resources schedules.

5.3 Discussion

Computational time. The computational time needed to attain the final solution is very brief. The algorithm reaches its best value after 19.3 seconds on average in the performed series of tests. This computational time is acceptable for the transit operators, considering that SbrT is a punctual operation that does not require real–time results.

Tabu Tenure. We chose to use a variable number of iterations of presence in the tabu list for each generated move. This number is randomly chosen inside a pre–defined interval. A set of different intervals was tested, and the impact of this parameter revealed to be very weak. Table 4 shows the results obtained with the extreme values tested for the interval, the upper bound of the largest interval matching approximately half the average size of the neighborhood. The method slightly benefitted from the increase in the number of iterations of presence. Complementarily, Table 5 shows that when moves remain in the tabu list between 500 and 1000 iterations, the complete neighborhood exploration (with associated evaluations) is on average, 2.65 times longer than when moves remain in the list for 7 to 20 iterations.

Table 4. Objective function value by instance and tabu list size

	TripShift	
	[7-20]	[500-1000]
Instance 1	1057.37	1057.52
Instance 2	1061.63	1062.62
Instance 3	1072.58	1072.65
Instance 4	1047.14	1047.25
Instance 5	1072.35	1072.37

Table 5. Average time (in seconds) needed for full exploration of the neighborhood by instance and tabu list size

	TripShift	
	[7-20]	[500-1000]
Instance 1	0.012	0.031
Instance 2	0.010	0.028
Instance 3	0.013	0.033
Instance 4	0.011	0.030
Instance 5	0.012	0.032

We observe that the results are of better quality using the interval [500-1000] compared with interval [7-20] in all five cases, however the observed variation is small. At the same time, larger values for the number of iterations of presence slows the search down, since a great number of verifications needs to be performed. In view of the combination of these two effects, we chose to keep the [7-20] interval in our method, to prevent negative effects in larger networks.

Upper Bound Computation. We tried to compute an upper bound for the problem, by taking advantage of the maximal shift constraint, which restricts the domain of each variable. We determined the set of all the transfer opportunities that could be generated independently (ie when the domains of both variables - runs implied in the transfer - encompassed values that fit into the allowed waiting time interval for the transfer). For each one, we computed the best cost that could be achieved. The upper bound obtained consists in 399 transfer opportunities and a 1332.11 value for the objective function. The drawback of this method is that it does not take into account the fact that a run can be involved in many transfer opportunities and assigns a potentially different value to the run for each transfer opportunity, providing unrealistic goals for the algorithm. This explains the large difference between the bound and the results actually achieved.

Neighborhood. We developed and tested a second neighborhood, which consists in shifting at once all the elements of a driver trip assignment by the same value, while preserving the feasibility of the vehicle trip assignments. This neighborhood proved to define a somewhat too small search space, preserving most of the initial schedules to the expense of fewer improvement possibilities. *TripShift* defines in comparison an interesting compromise between the improvement of the quality of service and the preservation of the existing resource usage.

6 Conclusion

An original problem was considered here, that goes backwards in the traditional transit planning process: the Scheduled–based re–Timetabling problem. We define a model based on the assignment of new starting times to all the runs in the timetables. Major constraints impose that the existing vehicle and driver trip assignments are preserved, and that their elements do not overlap despite the shifts in their starting times. The advantages of this model are its flexibility and its high level of preservation of the initial schedules (both in terms of number of resources needed and social acceptability).

We developed an adapted neighborhood mechanism that we integrated inside a method relying on tabu search. Tests were carried out on five generated instances based on a real network, and allowed a sensible improvement in terms of transfer opportunities on the case study. This combination of model and method shows the potential of the SbrT on a running schedule in terms of quality of service for transit users.

Acknowledgments. We wish to thank the reviewers for their helpful comments.

References

1. Ceder, A., Wilson, N.H.M.: Bus Network Design. Transportation Research Part B 20(4), 331–344 (1986)
2. Freling, R., Wagelmans, A., Paixao, J.M.P.: Models and algorithms for single-depot vehicle scheduling. Transportation Science 35(2), 165–180 (2001)
3. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers & Operations Research 13(5), 533–549 (1986)
4. Jansen, L.N., Pedersen, M.B., Nielsen, O.A.: Minimizing passenger transfer times in public transport timetables. In: 7th Conference of the Hong Kong Society for Transportation Studies: Transportation in the Information Age, pp. 229–239 (2002)
5. Ceder, A.: Crew scheduling. In: Public Transit Planning and Operation: Theory, Modeling and Practice, pp. 279–318. Butterworth-Heinemann (2007)
6. Guihaire, V., Hao, J.K.: Transit Network Re-timetabling and Vehicle Scheduling. In: MCO, vol. 14, pp. 135–144. Springer, Heidelberg (2008)

An Investigation of IDA* Algorithms for the Container Relocation Problem

Huidong Zhang[1], Songshan Guo[1], Wenbin Zhu[2,⋆],
Andrew Lim[3], and Brenda Cheang[3]

[1] Department of Computer Science, School of Information Science and Technology
Sun Yat-Sen University, Guangzhou, Guangdong, PR China (510275)
zwdant@gmail.com, issgssh@mail.sysu.edu.cn
[2] Department of Computer Science and Engineering, School of Engineering
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong S.A.R.
i@zhuwb.com
[3] Department of Management Sciences, College of Business
City University of Hong Kong, Tat Chee Ave, Kowloon Tong, Hong Kong S.A.R.
{lim.andrew,brendac}@cityu.edu.hk

Abstract. The container relocation problem, where containers that are stored in bays are retrieved in a fixed sequence, is a crucial port operation. Existing approaches using branch and bound algorithms are only able to optimally solve small cases in a practical time frame. In this paper, we investigate iterative deepening A* algorithms (rather than branch and bound) using new lower bound measures and heuristics, and show that this approach is able to solve much larger instances of the problem in a time frame that is suitable for practical application.

Keywords: Container Relocation Problem, IDA*, Heuristics.

1 Introduction

The retrieval of containers out of storage and onto transport vehicles is a common and crucial port operation. The containers in a storage bay must be retrieved one by one in a fixed sequence that is pre-determined by various constraints (e.g., maintenance of vessel balance, safety issues, etc.). The problem arises when the next container to be retrieved is not at the top of its stack, since all other containers above it must then be first relocated onto other stacks within the bay. The relocation of a container is an expensive operation that essentially dominates all other aspects of the problem, and therefore it is important that the retrieval plan minimizes the number of such relocations.

There are two versions of this *Container Relocation Problem* in existing literature: the *restricted* variant only allows the relocation of containers that are above the target container, while the *unrestricted* variant allows the relocation of

[⋆] Corresponding Author.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 31–40, 2010.

any container. Previous research has largely concentrated on only the restricted variant, which is also the focus of this study. We propose two new lower bound measures that are superior to the existing lower bound the restricted variant. We also investigate the effects of some greedy heuristics for the purpose of rapid updating of best known solutions during interative depeening A* search. Our research shows that using IDA* for the Container Relocation Problem is far superior to the standard branch and bound approach proposed in existing literature. In particular, we show that the restricted variant can be well solved for instances of practical size using IDA*.

2 Problem Description

In the container yard of a port, containers are stored in *blocks* (see Figure 1); each block consists of multiple *bays*; each bay consists of multiple *stacks* of containers; and each stack consists of multiple *tiers*. Container blocks typically comprise up to 20 bays, with each bay having maximum capacities between 2-10 stacks and 3-7 tiers.

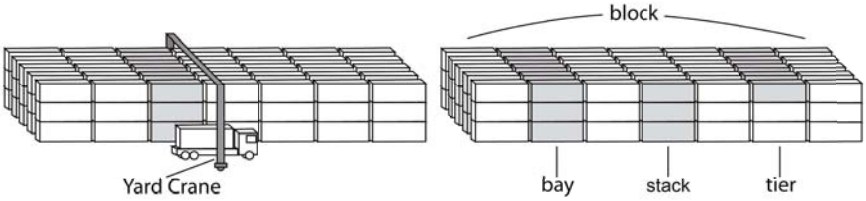


Fig. 1. Blocks in a Container Yard

Containers are retrieved from the bays using yard cranes, loaded onto AGVs, and transported onto quay cranes that finally load them onto vessels. The loading sequence of the containers by the quay cranes seeks to minimize the berth time of the vessel while satisfying various loading constraints (e.g., vessel balance and safety issues); the quay crane loading schedule determines the pickup sequence of the containers from the yards.

This study focuses on the yard crane scheduling problem, where the goal is to produce an operational plan for the retrieval of all containers in a given pickup sequence that minimizes the total time spent. Figure 2 illustrates an instance of the problem, where the numbers represent the pickup sequence of the containers. If the container that is to be retrieved next in the pickup sequence (called the *target* container) is on top of its stack, then it is simply retrieved. However, if this is not the case then all containers on top of the target container must first be relocated onto other stacks in the bay. The nature of yard crane operations is such that minimizing the total time taken by an operational plan is functionally equivalent to minimizing total number of *relocations* ([1,2,3,4,5]).

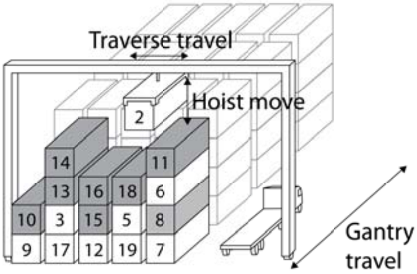


Fig. 2. Yard Crane Operations

3 Existing Approaches

Kim and Hong [4] described a standard branch and bound algorithm for the restricted variant that can optimally solve instances of up to 5 stacks by 5 tiers within 4 minutes, but is too slow for practical use for larger instances. A heuristic rule for generating solutions based on the expected number of relocations for a random operational plan was proposed as an alternative; this heuristic is able to find solutions that are within 10% of optimal for instances of up to 5 × 5 with running time of up to 2 seconds. While the unrestricted variant was mentioned, no solution was suggested for it.

Kim and Hong's work [4] was extended by Aydin [6], which proposed alternative branching rules for the standard branch and bound algorithm for the restricted variant while using the same lower bound. In combination with two greedy heuristics, the branch and bound algorithm was found to perform well for instances of up to 7 stacks by 7 tiers with about 37 containers, although the test cases had relatively low load factors ($55\% - 75\%$). A variant of the container relocation problem where traverse travel is modeled as a variable cost proportional to the number of stacks traveled was also considered (which is beyond the scope of this study).

Existing work on the storage of inbound containers so as to minimise relocations during retrieval include [7] and [2], which both proposed formulae for estimating the expected number of relocations. Kim et al. [3] developed a mathematical model and used a dynamic programming technique for this purpose, producing a decision tree for use during the storage process.

Avriel et al. [8] proposed a 0-1 integer-programming model and a heuristic method to solve a similar (but not identical) relocation problem of stowing containers onto a vessel.

4 Notation and Terminology

The following notation will be employed in this paper.

L	: A bay layout
N	: Number of containers in the initial bay layout
S	: Maximum number of stacks in the bay
T	: Maximum number of tiers in the bay
$F(L)$	: The minimum number of relocations required for layout L
$c_{s,t}$	: The retrieval order of the container on tier t of stack s
$\bar{s}$	: The stack containing the target
$\bar{t}$	: The tier containing the target; the target container is therefore $c_{\bar{s},\bar{t}}$
$smallest(s)$	: The smallest retrieval order in stack s, i.e., $smallest(s) = \min c_{s,t}$.

By convention, we order the stacks in ascending order 1..S from left to right, and the tiers 1..T from bottom to top.

5 Iterative Deepening A*

For a given bay layout, if the target container is at the top of its stack, then it can be immediately retrieved, resulting in a smaller bay layout with the same cost in terms of number of required relocations. Successive target containers can be retrieved as long as they are on top of their respective stacks at the time of retrieval, until the minimum equivalent layout is reached where there are either no more containers or there exists other container(s) on top of the target container. Hence, all layouts encountered in the search can be replaced by its minimum equivalent layout.

Every node n in the search tree corresponds to a minimum layout; every path from the root node to a leaf node corresponds to a solution to the initial layout. Branching occurs when relocation takes place. If the bay has S stacks, then every node in the search tree will have at most $S - 1$ children. Figure 3 illustrates the nodes in the first two levels of a search tree of the restricted variant.

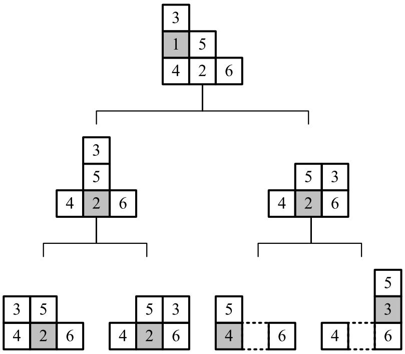


Fig. 3. First 2 Levels of a Search Tree for the Restricted Variant

The approach proposed in this study is an iterative deepening A* (IDA*) algorithm. In the first iteration, a strictly optimistic (i.e., non-overestimating) lower bound cost threshold t is estimated for the initial problem, which guarantees the permissibility of the solution. A depth-first branch and bound is then performed. Each node n in the search tree has a corresponding cost $f(n) = g(n) + h(n)$, where $g(n)$ is the number of relocations already made to arrive at that node (known as *confirmed* relocations) and $h(n)$ is a strictly optimistic estimation of the minimum number of relocations required to solve the problem from that node (known as *identified* relocations). Over the course of the search, the best lower bound $lb(n)$ for each node n is updated as all its children n' are evaluated, where $lb(n) = \max lb(n') + 1$.

One iteration of IDA* ends when the branch and bound has determined that all remaining nodes n have $f(n) > t$. This process is repeated with the threshold t incremented by 1, or set to the best lower bound found for the initial layout in the previous iteration, whichever is greater. The algorithm is completed when a solution is found with cost equal to t.

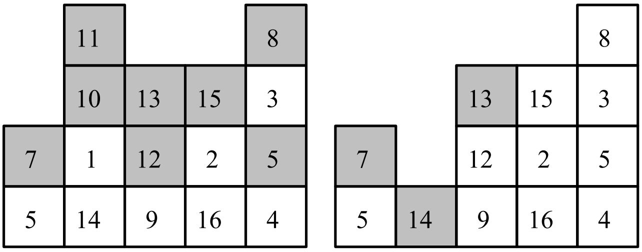
5.1 Lower Bounds

Recall that the estimated cost $f(n)$ of a particular layout is the sum of its confirmed relocations $g(n)$ and its identified relocations $h(n)$. In order for the solution to be admissible, $h(n)$ must never overestimate the number of relocations required. We examine three admissible lower bound measures that can be used as $h(n)$.

Lower Bound 1 (LB1): If a container is situated above another container with smaller retrieval order, then it must be relocated at least once. LB1 counts the number of such containers, i.e.,

$$LB1(L) = |\{c_{s,t} \mid \exists_{t',t'<t} \text{ such that } c_{s,t'} < c_{s,t}\}| \qquad (1)$$

All the shaded containers in Figure 4(a) must be relocated at least once. This is the lower bound described by Kim and Hong [4].



(a) Lower Bound 1 (LB1) (b) Lower Bound 3 (LB3)

Fig. 4. Lower Bounds LB1 and LB3

Lower Bound 2 (LB2): Observe in Figure 4(a) that there are three possible destinations to relocate container 11: on stacks 1, 3 and 4 (the rightmost stack is full and cannot hold any more containers). For all of the three possible choices, there will be a container with a smaller retrieval order below container 11 Hence, container 11 has to be relocated at least one more time. This inspection can be performed on all containers above the target (container 1), and the number of such cases is added to LB1, i.e.,

$$LB2(L) = LB1(L) + |\{c_{\bar{s},t} \mid t > \bar{t} \text{ and } \forall_{s' \neq \bar{s}}, \exists_{t'} \text{ such that } c_{s',t'} < c_{\bar{s},t}\}| \quad (2)$$

Lower Bound 3 (LB3): Given a layout L and a set of containers S, let $L - S$ denote the resultant layout when all containers in S are removed from L. It is apparent that for any layout $L' = L - S$, $F(L') \leq F(L)$. This observation allows an extension of LB2 to all containers (rather than only containers in stack $\bar{s}$), as illustrated by the following example.

Let L be the bay layout given in Figure 4(a). Consider container 15 on stack 4 in L; the container in stack 4 with smallest retrieval order is container 2, which will therefore be the target container when stack 4 is next considered. Suppose all containers with a smaller retrieval order than 2 is removed (container 1 in this case) along with all containers above them (containers 10 and 11 in this example); this results in a smaller layout $L' = L - \{c_{2,2}, c_{2,3}, c_{2,4}\}$ as depicted in Figure 4(b). We can then use the same analysis as LB2 for container 15 in L', i.e., for all of the possible destination stacks for its relocation, there will be a container with a smaller retrieval order than 15. Hence, container 15 must be relocated an additional time.

$$L'_{s,t} = L - \{c_{s',t'} \mid \exists_{t'',t'' \leq t'} \text{ such that } c_{s',t''} < \mathit{smallest}(s)\} \quad (3)$$

$$LB3(L) = LB1(L) + \sum_{s,t}(LB2(L'_{s,t}) - LB1(L'_{s,t})) \quad (4)$$

5.2 Probe Heuristics

The deepest nodes explored in each iteration of IDA* can be viewed as forming a frontier. Subsequent iterations push the frontier further and further towards the leaf of the search tree and stops when the first leaf node is reached.

IDA* requires the computation of $f(n')$ for all children n' of a node n before we can determine if n is on the frontier of the current iteration. It could potentially improve the effectiveness of the search if we could make good use of this information. If a child n' of a node on the frontier seems promising, we could invest the time to complete the partial solution represented by n' using a heuristic, which may turn out to be superior to the best known solution. In particular, we will apply a *heuristic probe* to n' if $f(n') \leq (lb + b)/2$, where lb is the lower bound of the root node found in previous iteration, and b is the cost of the best known solution found so far. The best known solution is updated if the heuristic finds a better solution, thus narrowing the search window.

This subsection describes the heuristics used to generate complete solutions during probing. These heuristics are used when the target container is not on the top of its stack, and provide criteria for moving a container from the top of one stack to another. We repeat the process until all containers are retrieved. Clearly, the height of destination stack must be less than T (we call such stacks *incomplete stacks*) and the destination stacks must differ from the source stack. In the following discussion, we implicitly assume that only incomplete stacks (other than the source stack) are considered when selecting the destination stack.

We examine four such heuristics.

PR1: The stack with minimum height; if there are multiple eligible stacks, then the leftmost eligible stack (with the smallest number) is selected.

PR1 is a simple heuristic that selects the shortest stack on the premise that it minimizes the possibility that the relocated container will be placed above a container with smaller retrieval order.

PR2: The stack with highest $smallest(s)$ (refer to Section 4).

PR2 is motivated by the observation that if a container c is relocated to a stack with $smallest(s)$ greater than its retrieval order, then container c need not be relocated again in the future. By picking the stack with the highest $smallest(s)$ value, we maximise the probability of this occurence.

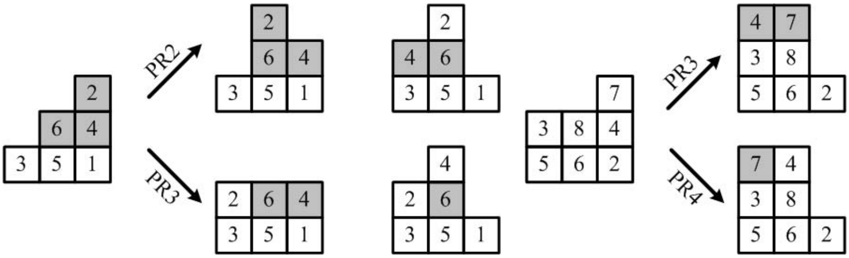
PR3:

a) If there are one or more stacks with $smallest(s)$ greater than the retrieval order of target container $c_{\bar{s},\bar{t}}$, select the stack with lowest $smallest(s)$;
b) Otherwise, select the stack with highest $smallest(s)$.

If there are multiple stacks with $smallest(s)$ greater than the retrieval order of container c, any such stack is an equally good choice for c. Therefore, we prefer the stack with the lowest $smallest(s)$ to maximize the number of containers that could satisfy this criterion in the future. Figure 5(a) illustrates the motivation for case a) for PR3 when compared to PR2, where containers that require at least one additional relocation are shaded.

PR4: Similar to PR3, except in case b) we perform an additional check: if the container to be relocated is not the 2nd smallest in the source stack, and the stack with highest $smallest(s)$ has only one available slot, then pick the stack with the 2nd highest $smallest(s)$ instead.

PR4 is motivated by the example illustrated by Figure 5(b). No matter which stack is selected as the destination for container 7, it must be relocated again. If we follow case b) of PR3 and choose stack 2 for container 7, then container 4 has to be relocated to stack 1, and at least one more relocation is needed for container 4. However, if we follow PR3 and pick stack 1 (the stack with 2nd highest $smallest(s)$) for container 7, then container 4 can be relocated to stack 2 and no future relocation is needed.



(a) PR2 versus PR3 (b) PR3 versus PR4

Fig. 5. Comparison of Probe Heuristics

6 Experiments and Analysis

Each test case is composed of three parameters N, S and T. A total of 125 sets of instances are randomly generated, with $S \in [6, 10]$, $T \in [3, 7]$ and $N \in [(S-1)\times T, S\times T-1]$; 100 instances were generated for each combination of S, T and N. Note that a layout has solutions if and only if any container with retrieval order i has at most $i-1+e$ containers over it, where $e = S\times T-N$ is number of empty slots. This check only needs to be performed for containers with retrieval order $i = 1, 2, ..., T-e$; infeasible layouts are discarded and regenerated.

All experiments were conducted on a DELL personal desktop computer with a Pentium 4 CPU clocked at 3.0GHz, loaded with 1.0GB RAM and running the Windows XP operating system. All algorithms were implemented in C++ and compiled using Visual Studio C++ 2008 Team Edition (the default Release configuration for the Win32 platform was used).

The number of relocations estimated by lower bound LB2 is 4.26% larger than that of LB1 on average. The number of relocations estimated by lower bound LB3 is 10-20% larger than that of LB1. In general, the gap in performance between LB3 and LB1 increases as the size of the instances increase.

When the probe heuristics PR1, PR2, PR3 and PR4 are applied to the initial layouts of each test case, the average number of relocations (over all 12,500 cases) are 44.07, 37.87, 33.36, and 33.07 respectively. Since none of the heuristics strictly dominates all others over all test cases, we created a composite heuristic PR+ that simply takes the best solution found by PR1, PR2, PR3 and PR4.

Our best approach for solving the restricted variant using IDA* is as follows. We first construct an initial solution using heuristic PR+ (i.e., the best solution found by PR1, PR2, PR3 and PR4). During the search, LB3 is used as the lower bound to prune nodes, and heuristic PR4 is used to probe the children of frontier nodes. We refer to the resulting algorithm as IDA_PR+_LB3_PR4. We impose a strict time limit of 1 second of CPU time for each test instance; once the time limit is reached, the best solution found so far is reported.

To accurately gauge the performance of the algorithm, we first try to solve as many instances optimally as possible disregarding the time limit. We used a high performance computing server to run IDA_PR+_LB3_PR4 over a long period of

Table 1. Collated Results of IDA_PR+_LB3_PR4 for the Restricted Variant

Instance		Best Known		IDA_PR+_LB3_PR4				
T	Count	LB	Reloc	Reloc	LB Gap	Best Gap	time(s)	opt %
3	1500	13592	13592	13592	0.00%	0.00%	0	100.00%
4	2000	33095	33095	33095	0.00%	0.00%	0.12	100.00%
5	2500	63506	63507	63511	0.01%	0.01%	11.105	99.80%
6	3000	105549	105669	105918	0.35%	0.24%	117.729	92.63%
7	3500	160329	162660	164482	2.59%	1.12%	468.413	61.29%
Total	12500	376071	378523	380598	1.20%	0.55%	597.367	87.35%

time. For each instance we allow IDA_PR+_LB3_PR4 to explore as many as 2^{30} nodes.

Table 1 gives the collated results for the 12,500 instances. The results for all instances with same number of tiers are grouped together and the total number of relocations for the group is reported. *Count* shows the total number of instances in the group. The next two columns *LB* and *Reloc* gives the best known lower bounds and solutions, totaled over all instances in the group; the disparities between these two values arise when the algorithm is unable to find an optimal solution despite searching 2^{30} nodes.

The remaining five columns provide the results of IDA_PR+_LB3_PR4 under the 1 second time limit on a Pentium 4 desktop computer. *Reloc* gives the total number of relocations of the best solutions found by the algorithm for all instances in the group. This is followed by the percentage difference between this value and the best known lower bound (*LB Gap*), and between this value and the best known solutions (*Best Gap*), respectively. *time(s)* gives the total time in seconds elapsed before IDA_PR+_LB3_PR4 produced a solution, summed over all instances in the group; this value is somewhat inaccurate as we used the standard C library clock() function for this purpose, which has a precision of 1 millisecond, and many instances can be solved in less than 1ms. Finally, the last column *opt %* gives the percentage of instances in the group that was solved optimally, where the number of relocations of the solution matches the best known lower bound.

The data shows that the majority of instances with $T \leq 6$ can be solved optimally within 1 second. For large instances where $T = 7$, more than half the instances are solved optimally, and the average gap between the solution found and the best known lower bound is about 2.59%; this equates to an average difference from optimal of fewer than 2 relocations. We can conclude that our IDA_PR+_LB3_PR4 can solve instances of up to 7 tiers and 10 stacks to near optimal within 1 second for the restricted variant.

Our algorithm is a significant improvement over Kim and Hong [4], whose branch and bound approach is only able to solve instances of up to 5 tiers by 5 stacks within 4 minutes, and the solution found by the proposed heuristic is about 10% off the optimal solution.

Although Aydin [6] used a branch and bound algorithm similar to Kim and Hong [4] and is able to solve about 91% of instances of up to 7 tiers by 7 stacks, the test case generation strategies for that research have load factors ranging from 55% - 75%. Hence, the largest instance consists of only $7 \times 7 \times 0.75 = 37$ containers. In contrast, the test instances in our work are nearly fully loaded and consists of up to 69 containers. Since every subtree of the search tree with a root node containing m containers is itself a search tree, our algorithm actually solves several instances of the problem with $m = 37$ containers over the course of searching for solutions for larger instances.

7 Conclusions

In this paper, we examined the use of IDA* algorithms on the container relocation problem. We introduced two new lower bound measures for branch and bound pruning; in particular, the LB3 lower bound measure has proven to be much more effective than the LB1 measure that has been suggested in existing literature. We also made use of probe heuristics for promising partial solutions in order to narrow the search window; the heuristic PR3 has been found to be especially useful for this purpose. The resultant IDA_PR+_LB3_PR4 algorithm is able to solve instances of up to 7 tiers and 7 stacks to within 1 to 2 relocations of optimal for the restricted variant, which largely covers the set of practical cases. Hence, the restricted variant of the container relocation problem can be considered well solved for instances of practical size.

References

1. Ashar, A.: On selectivity and accessibility. Cargo Systems, 44–45 (June 1991)
2. Kim, K.H.: Evaluation of the number of rehandles in container yards. Comput. Ind. Eng. 32(4), 701–711 (1997)
3. Kim, K.H., Park, Y.M., Ryu, K.R.: Deriving decision rules to locate export containers in container yards. European Journal of Operational Research 124(1), 89–101 (2000)
4. Kim, K.H., Hong, G.P.: A heuristic rule for relocating blocks. Comput. Oper. Res. 33(4), 940–954 (2006)
5. Watanabi, I.: Selection process. Cargo Systems, 35–36 (March 1991)
6. Aydin, C.: Improved rehandling strategies for container retrieval process. Master's thesis, Graduate School of Engineering and Natural Sciences, Sabanci University (August 2006)
7. de Castillo, B., Daganzo, C.F.: Handling strategies for import containers at marine terminals. Transportation Research Part B: Methodological 27(2), 151–166 (1993)
8. Avriel, M., Penn, M., Shpirer, N., Witteboon, S.: Stowage planning for container ships to reduce the number of shifts. Annals of Operations Research 76(0), 55–71 (1998)

MADS/F-Race: Mesh Adaptive Direct Search Meets F-Race

Zhi Yuan, Thomas Stützle, and Mauro Birattari

IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium
{zyuan,stuetzle,mbiro}@ulb.ac.be

Abstract. Finding appropriate parameter settings of parameterized algorithms or AI systems is an ubiquitous task in many practical applications. This task is usually tedious and time-consuming. To reduce human intervention, the study of methods for automated algorithm configuration has received increasing attention in recent years.

In this article, we study the mesh adaptive direct search (MADS) method for the configuration of parameterized algorithms. MADS is a direct search method for continuous, global optimization. For handling the stochasticity involved in evaluating the algorithm to be configured, we hybridized MADS with F-Race, a racing method that adaptively allocates an appropriate number of evaluations to each member of a population of candidate algorithm configurations. We experimentally study this hybrid of MADS and F-Race (MADS/F-Race) and compare it to other ways of defining the number of evaluations of each candidate configuration and to another method called I/F-Race. This comparison confirms the good performance and robustness of MADS/F-Race.

1 Introduction

The performance of parameterized algorithms for computationally hard problems depends on the appropriate choice of parameter values. Parameter settings are still often searched in a trial-and-error fashion and typically this is a time-consuming, labor intensive process. In this article, we focus on the *offline* algorithm configuration problem (ACP) [1]. (We call an *algorithm configuration* the full specification of all algorithm parameters, that is, a fully instantiated algorithm.) In the offline ACP, one distinguishes two phases. In a first *training phase*, an algorithm configuration is determined that minimizes the expected cost of the solutions generated by the algorithm. This algorithm configuration is then used in a *production phase*, where the algorithm is run on unseen instances. It is important to stress that the crucial aspect here is generalization, that is, the ability of the tuning method to find parameter settings that work well on previously unseen instances.

In the settings we are interested in, the evaluation of a candidate algorithm configuration is stochastic. Stochasticity may arise due to two main reasons. Firstly, the algorithm may take randomized decisions during the search. This is a main characteristic of stochastic local search (SLS) algorithms [2], but also of

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 41–50, 2010.

any randomized search algorithms. Secondly, the performance of an algorithm, even if it is deterministic, usually depends on the particular instance being tackled. In fact, the particular instance can naturally be seen as being drawn according to some, typically unknown, probability distribution; it is, in this sense, a second stochastic factor. Tackling this stochastic aspect of the problem was the objective of the F-Race method [3]. The crucial idea is to evaluate a set of candidate algorithm configurations in parallel on a stream of instances and to use a particular racing method for determining the best one. In F-Race, at each evaluation step, that is, after each additional instance, the non-parametric family-wise Friedman test is applied to check whether there is evidence that at least one of the configurations is significantly different from the others. If the null hypothesis of no differences is rejected, Friedman post-tests are applied to eliminate those candidate configurations that are significantly worse than the best one.

F-Race is a general method that can be used to select the best candidate under stochastic evaluation. It can be used independent of the way the candidate algorithm configurations are actually generated. In this article, we adopt an iterative sampling method called mesh adaptive direct search (MADS) [4]. MADS is a direct search method for continuous function optimization. In fact, in this article we consider only configuration tasks with numerical parameters with large domains: either these parameters are continuous, or they have a (large) integer domain such that rounding real numbers to the closest integers is reasonable. The latter is usually the case for many algorithm parameters such as population sizes or tabu list lengths. One contribution of the article is the hybridization of MADS with F-Race for the evaluation of the candidate algorithm configurations, resulting in the MADS/F-Race algorithm. Another contribution is the application of the leave-one-out cross-validation for experimental comparison of MADS/F-Race to alternative uses of MADS. Finally MADS/F-Race is also compared to the state-of-the-art tuning method I/F-Race [5].

2 Combining MADS and F-Race

The MADS class of algorithms tackles global optimization problems that can be stated as

$$\min_{p \in \Omega} f(p) \tag{1}$$

for $f : \Omega \in \mathbb{R}^d \to \mathbb{R} \cup +\infty$, where $d = |\Omega|$ is the dimensionality of the variable space. The algorithm does not require derivative information or continuity of f. In fact, the evaluation function f can be considered as a black-box. MADS algorithms are reported to be robust for non-smooth optimization problems [4], and its convergence properties given only bound constraints is shown in [6]. It is an extension to the generalized pattern search [7] by allowing a more general exploration of the variable space.

Algorithm 1. The mesh adaptive direct search

```
Step 0: Initialization Given $p_0$, a feasible initial point, $\Delta_0 > 0$, set $k = 0$ and go to Step 1.
Step 1: search step Let $L_k \subset M_k$ be the finite set of mesh points, and let $q^* = \arg\min_{q \in L_k} f(q)$ be the iteration best point from search. If $f(q^*) < f(p_k)$, declare $k$ successful and set $p_{k+1} = q^*$ and go to Step 3; otherwise go to Step 2.
Step 2: poll step Generate the frame $F_k$ as in Eq. 3, and let $q^* = \arg\min_{q \in F_k} f(q)$ be the iteration best point from poll. If $f(q^*) < f(p_k)$, declare $k$ successful and set $p_{k+1} = q^*$; otherwise declare $k$ unsuccessful. Go to Step 3.
Step 3: Parameter update If iteration $k$ is unsuccessful, set $p_{k+1} = p_k$, otherwise set $p_{k+1}$ as the improved mesh point. Update mesh size parameter $\Delta_k$ according to Eq. 4, $k \leftarrow k + 1$ and go to Step 1.
```

2.1 A General Overview of MADS

MADS is an iterative algorithm that generates at each iteration a population of trial points. The trial points must lie on the *current mesh*, whose coarseness is controlled by a *mesh size parameter* $\Delta_k \in \mathbb{R}_+$. At iteration k, the set of mesh points is given by

$$M_k = \bigcup_{p \in S_k} \{p + \Delta_k z : z \in \mathbb{Z}^d\} \tag{2}$$

where S_k is the point set that was evaluated in the previous iterations. At each iteration of MADS, two steps are applied, the search step and the poll step. In the search step, trial points are randomly sampled on the current mesh. In the poll step, trial points are generated around the incumbent one. The trial points of poll form the set called *frame*:

$$F_k = \{p_k \pm \Delta_k b : b \in B_k\} \tag{3}$$

where $B_k = \{b^1, b^2, \dots, b^d\}$ is a basis in $\mathbb{Z}^d$. At iteration k, the mesh size parameter is updated as follows:

$$\Delta_{k+1} = \begin{cases} \Delta_k/4 & \text{if no improved mesh point is found;} \\ 4\Delta_k & \text{if an improved mesh point is found, and } \Delta_k \leq \frac{1}{4}; \\ \Delta_k & \text{otherwise.} \end{cases} \tag{4}$$

An outline of the MADS algorithm is given in Algorithm 1.

2.2 The Hybrid of F-Race and MADS

The stochasticity associated to the evaluation of candidate algorithm configurations usually requires to reevaluate a configuration several times to reduce the variance of the performance estimate. The most straightforward way of doing so is to apply each candidate algorithm configuration a fixed, same number of times; this variant we call MADS(fixed). The number of times an algorithm candidate configuration is run in MADS(fixed) we denote by nr, where nr is a parameter.

Table 1. Parameters used in MADS. d denotes the dimension of the parameter space.

parameter	value
Initial search type	Latin hypercube search
#Points in initial search	d^2
Iterative search type	random search
#Points in iterative search	$2d$
#Points in poll step	$2d$
Speculative search	yes
Initial mesh size parameter	1.0

Two disadvantages of MADS(fixed) are that (i) a priori an appropriate value of nr is not known, and (ii) the same amount of computation time is allocated to good and bad performing candidates, wasting useful computation.

These disadvantages are reduced by hybridizing F-Race and MADS, resulting in the MADS/F-Race algorithm. This hybrid can be obtained in a rather straightforward way. At each MADS iteration, independent of whether it is the search or poll step, the population of candidate configurations sampled by MADS is evaluated by F-Race, together with the incumbent point p_k. In this hybrid, F-Race identifies the best point among the sampled ones and $\{p_k\}$. If F-Race determined p_k as the best, the iteration is declared unsuccessful; otherwise the iteration is successful, and the MADS parameters are updated accordingly.

2.3 Implementation Details of MADS/F-Race

For the setting of the MADS algorithm, we follow [6]. The specific parameter values are listed in Table 1. For MADS/F-Race, we essentially only need to set the maximum number of algorithm evaluations we allow in each application of F-Race. Here, we explore a fixed setting of 10 times the number N_l of trial points generated from MADS at the l-th iteration.

3 Case Study: MADS/F-Race vs. MADS(fixed)

We have compared the performance of MADS/F-Race and MADS(fixed) on six benchmark *configuration problems*. Each *configuration problem* consists of a parameterized algorithm to be configured, and an optimization problem to which this algorithm is applied. The six configuration problems are listed below:

- MMASTSP: the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System ($\mathcal{MM}$AS) algorithm [8] applied to the traveling salesman problem (TSP). There are six parameters of $\mathcal{MM}$AS to be configured (listed in Table 2). No local search is applied.
- MMASTSP_ls: the $\mathcal{MM}$AS applied to TSP with 2-opt local search on solutions constructed by each ant. The six parameters to be configured are the same as MMASTSP.
- ACSTSP: the ant colony system (ACS) algorithm [9] for the TSP. The six parameters to be configured are listed in Table 2. No local search is applied.

Table 2. Range of each parameter considered for configuring MMASTSP (including MMASTSP_ls) and ACSTSP (including ACSTSP_ls)

MMASTSP		ACSTSP	
parameter	range	parameter	range
α	$[0.0, 5.0]$	α	$[0.0, 5.0]$
β	$[0.0, 10.0]$	β	$[0.0, 10.0]$
ρ	$[0.0, 1.00]$	ρ	$[0.0, 1.00]$
γ	$[0.01, 5.00]$	q_0	$[0.0, 1.0]$
m	$[1, 1200]$	m	$[1, 1200]$
nn	$[5, 100]$	nn	$[5, 100]$

Table 3. Range of each parameter considered for configuring ILSQAP. The individual parameters refer to parameters concerning the tabu search algorithm, the simulated-annealing type of acceptance criterion of the ILS, and the perturbation strength.

parameter	range	parameter	range	parameter	range	parameter	range
tc	$[0, 5]$	ia	$[3, 99]$	iu	$[0, 2]$	iy	$[0, 1]$
ti	$[0, 20]$	it	$[0, 10]$	ix	$[1, 100]$	iz	$[0, 1]$

- ACSTSP_ls: ACS applied to TSP with 2-opt local search. The six parameters to be configured are the same as in ACSTSP.
- ILSQAP: an iterated local search (ILS) algorithm [10] applied to the quadratic assignment problem (QAP). This ILS algorithm applies a robust tabu search as the local search and a simulated annealing type acceptance criterion. The eight parameters to be configured are listed in Table 3;
- SAQAP: the simulated annealing (SA) for QAP with three parameters to be configured (Table 4).

In our experiments, each of the configuration problems contains 12 *domains*. These are defined by considering for each configuration problem four computation time limits (1, 2, 5, and 10 CPU seconds) for stopping the algorithm to be configured, and three values of *configuration budget* (1000, 2000, and 4000). The term *configuration budget* is determined as the maximum number of times that the algorithm to be configured can be applied during the configuration process. Take MADS(fixed) for example, given a configuration budget of 1000, and $nr = 20$, then each parameter configuration will be evaluated 20 times, and in total 50 different parameter configurations will be evaluated.

3.1 Comparison of MADS/F-Race to Random MADS(fixed)

As a first benchmark test for MADS/F-Race, we compare its performance to MADS(fixed), for which six values for nr are tested: $nr \in \{1, 5, 10, 20, 30, 40\}$. Given no a priori information about the best choice for nr, we select one of the six values randomly for each problem domain, and compare it with MADS/F-Race

Table 4. Range of each parameter considered for configuring SAQAP; the parameters define the annealing schedule

parameter	range	parameter	range	parameter	range
sb	$[0.0, 10.0]$	sc	$[0.0, 1.0]$	sd	$[1, 100000]$

across all domains, with blocking on each instance. The comparison is done using the Wilcoxon signed rank test with continuity correction; the α-level chosen is 0.05. The experimental results show that `MADS/F-Race` statistically significantly performs better than `MADS(fixed)` with randomly chosen nr on each individual domain. The average percentage improvement of the cost obtained by `MADS/F-Race` over the cost obtained by the random `MADS(fixed)` is around 0.25%.

3.2 The Leave-One-Out Cross-Validation

The dominance of `MADS/F-Race` over a random selection of the value nr for `MADS(fixed)` is convincing if no a priori knowledge about an appropriate value for nr is available. As our knowledge of `MADS(fixed)` grows, we may learn a good setting for `MADS(fixed)`, with which it can be applied to an unknown problem domain. In the sequel, we study how well the knowledge of `MADS(fixed)` over the learned domains can be generalized to an unknown domain. To this end, a statistical technique named leave-one-out cross-validation is applied.

Cross-validation is a technique that is commonly used in machine learning and statistics, to assess how the results of statistical analyses are generalized to an independent data set. To do so, on each cross-validation iteration, the available data set is partitioned into two sets, the training set based on which the quality of the candidate settings are observed and analyzed, and the test set on which the previously assessed results are tested. This process is iterated, using different partitions in each iteration, to reduce the variability of the validation.

There are various ways how the cross-validation can be performed. In our case, the common leave-one-out cross-validation is applied. In each iteration, one domain is picked for testing, and the rest of the domains serve as the training set. The best candidate chosen in the training set is applied on the test domain, and its results are collected into a validation set. The process repeats until each domain has collected its validation data. A more formal description of how we apply the leave-one-out cross-validation is given in Algorithm 2.

The leave-one-out cross-validation in our experiments takes as candidates the `MADS(fixed)` with six values of nr. The goal is to collect the validation set as described above, and compare it with the results obtained by `MADS/F-Race`. Note that the data collected in the validation set are first trained, while `MADS/F-Race` is not. The unfairness of the comparison will show the advantage of `MADS/F-Race` in robustness over the naive `MADS(fixed)`.

Algorithm 2. The leave-one-out cross-validation

```
Given a set of domains $D$, a set of candidate algorithms $A^c$, and the function $c(a, E)$
referring to the expected cost of algorithm $a \in A^c$ on a set of domains $E \in D$.
Goal is to collect the validation set $V^c$
Set $V^c \leftarrow \emptyset$
for $d \in D$ do
   Select $\bar{a} = \arg\min_{a \in A^c} c(a, D \setminus \{d\})$
   Set $V^c \leftarrow V^c \cup \{c(\bar{a}, d)\}$
end for
Return $V^c$ for assessment
```

3.3 Comparison of MADS/F-Race to Tuned MADS(fixed)

Here we compare MADS/F-Race with the tuned version of MADS(fixed). Firstly we applied the leave-one-out cross-validation for MADS(fixed) over all the six configuration problems. This is done as follows. Across $12 \cdot 6 = 72$ domains, the validation set of each domain is collected by applying MADS(fixed) with the value of nr, which has the best performance for the other 71 domains. We observed that the best value of nr determined by each combination of the 71 domains is always five. As previously, we apply the Wilcoxon test. It shows that MADS/F-Race significantly outperforms the tuned MADS(fixed). The observed difference in the mean performance between the tuned MADS(fixed) and MADS/F-Race is 0.17%.

Moreover, we observed that the best setting of nr in MADS(fixed) may differ for each individual configuration problem. To further fine-tune nr for MADS(fixed), we did a blocking on the set of domains by configuration problems, and applied the leave-one-out cross-validation on 12 domains of each configuration problem separately. Since nr is better tuned, the results of MADS(fixed) in this case are better than obtained by MADS(fixed) tuned across all configuration problems. The comparison results of MADS/F-Race and MADS(fixed) tuned in individual configuration problems are listed in Table 5, together with the best value of nr for each configuration problem. It shows that MADS/F-Race obtains significantly better results than tuned MADS(fixed) on individual configuration problems MMASTSP, ACSTSP and SAQAP, while in the rest of the configuration problems MMASTSP_ls ACSTSP_ls and ILSQAP, MADS/F-Race has significantly worse results than tuned MADS(fixed). Also note that when tuning nr for individual configuration problem, the tuned nr values vary a lot from 5 to 40.

The fact that the advantage of MADS/F-Race is more clear in some domains than in others motivates us to examine factors of the configuration problems that may affect the performance of the configuration algorithms MADS(fixed) and MADS/F-Race. Given that F-Race is particularly designed to deal with the stochastic aspect of the problem, one conjecture is that for problem domains where the stochasticity is relatively high, better performance is obtained by using F-Race. As a scale-free indicator of the variability of the configurations in a problem domain, we have investigated the *variation coefficient*. The variation coefficient of each configuration problem is obtained as follows. Firstly, 200

Table 5. Comparison results of `MADS/F-Race` and `MADS(fixed)` with nr tuned in six individual configuration problems separately by leave-one-out cross-validation. The column entries with label "best.nr" show the tuned nr value for each configuration problem. It is worth noticing that, in each cross-validation process, this nr value selected by each training set can vary. But in our case, it happened to remain constant. The column entries with the label `per.dev` show the mean percentage deviation of cost obtained by `MADS/F-Race` comparing with tuned `MADS(fixed)`. $+x$ $(-x)$ means that the solution cost of `MADS/F-Race` is x% more (less) than tuned `MADS(fixed)`, i.e. `MADS/F-Race` performs x% worse (better) than tuned `MADS(fixed)`.

problem	`per.dev`	best.nr	problem	`per.dev`	best.nr
MMASTSP	−0.12	5	MMASTSP_ls	+0.055	5
ACSTSP	−0.64	40	ACSTSP_ls	+0.027	40
SAQAP	−0.030	30	ILSQAP	+0.13	40

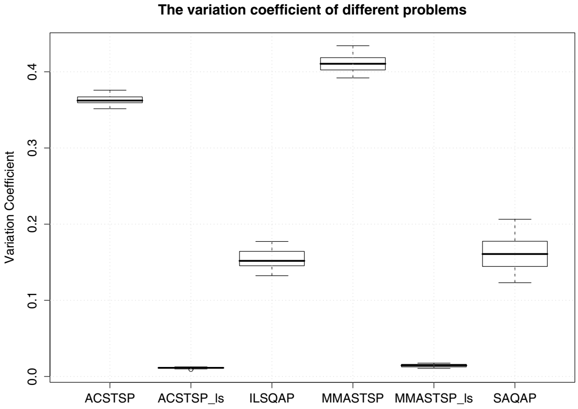


Fig. 1. The variation coefficient of each problem used

randomly generated parameter settings are applied on 25 generated instances. Then, for each of the instances we compute the ratio of the standard deviation over the mean cost of the 200 parameter settings. A high value of the variation coefficient indicates steep slopes in the search landscape, while a low value of the variation coefficient indicates a flat landscape, which is often the case for algorithms when a strong local search is adopted. The box-plot of the measured variation coefficients on the six configuration problems is shown in Figure 1.

Interestingly, the two configuration problems MMASTSP and ACSTSP, on which `MADS/F-Race` performs particularly well, are the two with the highest variation coefficient. Although the variation coefficient of SAQAP is not significantly higher than that of ILSQAP, it is still much larger than that of ACSTSP_ls and MMASTSP_ls. This gives further evidence for our conjecture that the impact of `F-Race` is the larger the higher the variability of algorithm performance.

Table 6. Comparison of MADS/F-Race and I/F-Race for configuring six benchmark configuration problems. The column entries with the label per.dev shows the mean percentage deviation of cost obtained by MADS/F-Race comparing with I/F-Race. $+x$ $(-x)$ means that the solution cost of MADS/F-Race is x% more (less) than I/F-Race, i.e. MADS/F-Race performs x% worse (better) than I/F-Race.

problem	per.dev	problem	per.dev
MMASTSP	−0.34	MMASTSP_ls	+0.053
ACSTSP	+1.12	ACSTSP_ls	−0.064
SAQAP	−0.80	ILSQAP	+0.48

3.4 Comparison of MADS/F-Race and I/F-Race

Finally, we compared the proposed MADS/F-Race to I/F-Race, another state-of-the-art configuration algorithm [5]. I/F-Race was applied to the same benchmark configuration problems as MADS/F-Race. In each domain one configuration procedure is run, and the tuned parameters are evaluated on 300 test instances. All the results are compared using paired Wilcoxon's signed rank test. When measured across all test domains, we could not detect statistically significant differences between MADS/F-Race and I/F-Race (p-value was 0.82). However, analyzing the results separately for each configuration problem (see Table 6), MADS/F-Race is significantly better than I/F-Race on domains MMASTSP, ACSTSP_ls and SAQAP, and significantly worse on domains ACSTSP, MMASTSP_ls and ILSQAP.

4 Conclusions

In this article, we have explored the hybridization of MADS with F-Race. Computational results have shown that this hybridization is successful. Further analysis of the performance variability among the benchmark configuration problems has shown that it is particularly useful for tackling the stochastic aspect of the problem. This insight shed also more light on situations under which similar combinations of optimization methods with statistical methods such as racing [11,12] are particularly useful.

In future works, we will examine possibilities for improving MADS/F-Race and also I/F-Race by considering alternative ways of defining the races by, for example, re-using previous evaluations. Another direction is the study of other continuous optimization techniques for algorithm configuration to widen the set of available methods for automated algorithm configuration. We strongly believe that the integration of appropriate statistical methodologies with effective sampling approaches will allow a further boost of the performance of automated algorithm configuration tools.

Acknowledgments. We would like to thank Mohamed Saifullah bin Hussin to make available his iterated local search and simulated annealing implementations for the quadratic assignment problem. This work has been supported by META-X, an ARC project funded by the French Community of Belgium. The authors acknowledge support from the fund for scientific research F.R.S.-FNRS of the French Community of Belgium, of which they are research associates (MB and TS), or an aspirant (ZY), respectively.

References

1. Birattari, M.: Tuning Metaheuristics: A machine learning perspective. Springer, Berlin (2009)
2. Hoos, H.H., Stützle, T.: Stochastic Local Search—Foundations and Applications. Morgan Kaufmann Publishers, San Francisco (2005)
3. Birattari, M., Stützle, T., Paquete, L., Varrentrapp, K.: A racing algorithm for configuring metaheuristics. In: Langdon, W.B., et al. (eds.) Proceedings of GECCO 2002, pp. 11–18. Morgan Kaufmann Publishers, San Francisco (2002)
4. Audet, C., Dennis, J.: Mesh adaptive direct search algorithms for constrained optimization. SIAM Journal on Optimization 17(1), 188–217 (2007)
5. Birattari, M., Yuan, Z., Balaprakash, P., Stützle, T.: F-Race and iterated F-Race: An overview. In: Bartz-Beielstein, T., et al. (eds.) Experimental Methods for the Analysis of Optimization Algorithms. Springer, Berlin (to appear)
6. Audet, C., Orban, D.: Finding optimal algorithmic parameters using derivative-free optimization. SIAM Journal on Optimization 17, 642 (2006)
7. Audet, C., Dennis, J.: Analysis of generalized pattern searches. SIAM Journal on Optimization 13, 889–903 (2000)
8. Stützle, T., Hoos, H.H.: $\mathcal{MAX} - -\mathcal{MIN}$ Ant System. Future Generation Computer Systems 16(8), 889–914 (2000)
9. Dorigo, M., Gambardella, L.M.: Ant Colony System: A cooperative learning approach to the traveling salesman problem. IEEE Transactions on Evolutionary Computation 1(1), 53–66 (1997)
10. Stützle, T.: Iterated local search for the quadratic assignment problem. European Journal of Operational Research 174(3), 1519–1539 (2006)
11. Yuan, B., Gallagher, M.: Combining Meta-EAs and racing for difficult EA parameter tuning tasks. In: Lobo, F., Lima, C., Michalewicz, Z. (eds.) Parameter Setting in Evolutionary Algorithms, pp. 121–142. Springer, Berln (2007)
12. Smit, S., Eiben, A.: Comparing parameter tuning methods for evolutionary algorithms. In: Proceedings of the 2009 IEEE Congress on Evolutionary Computation, pp. 399–406. IEEE Press, Piscataway (2009)

An Improvement in Audio-Visual Voice Activity Detection for Automatic Speech Recognition

Takami Yoshida[1], Kazuhiro Nakadai[1,2], and Hiroshi G. Okuno[3]

[1] Graduate School of Information Science and Engineering,
Tokyo Institute of Technology, Tokyo, Japan
yoshida@cyb.mei.titech.ac.jp

[2] Honda Research Institute Japan, Co., Ltd., Saitama, Japan
nakadai@jp.honda-ri.com

[3] Graduate School of Informatics, Kyoto University, Kyoto, Japan
okuno@kuis.kyoto-u.ac.jp

Abstract. Noise-robust Automatic Speech Recognition (ASR) is essential for robots which are expected to communicate with humans in a daily environment. In such an environment, Voice Activity Detection (VAD) strongly affects the performance of ASR because there are many acoustically and visually noises. In this paper, we improved Audio-Visual VAD for our two-layered audio visual integration framework for ASR by using hangover processing based on erosion and dilation. We implemented proposed method to our audio-visual speech recognition system for robot. Empirical results show the effectiveness of our proposed method in terms of VAD.

Index Terms: Audio-Visual integration, Voice Activity Detection, Speech Recognition.

1 Introduction

Service/home robots which are required to communicate with humans in daily environments should have a noise-robust Automatic Speech Recognition (ASR) function. In such an environment, there are many kinds of noises such as other speakers and robot's own noise. So, robots should cope with contaminated input signals.

To realize such a robot, there are two approaches. One is sound source separation to improve SNR of the input speech. The other is the use of another modality, that is, audio-visual (AV) integration.

For sound source separation, we can find several studies, especially, in the field of "Robot Audition" proposed in [1], which aims at building listening capability for a robot by using its own microphones. Some of them reported highly-noise-robust speech recognition such as three simultaneous speech recognition [2]. However, in a daily environment where acoustic conditions such as power, frequencies and locations of noise and speech sources dynamically change, the performance of sound source separation sometimes deteriorates, and thus ASR

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 51–61, 2010.

does not always show such a high performance. For AV integration for ASR, many studies have been reported as *Audio-Visual Speech Recognition (AVSR)* [3,4,5]. However, they assume that the high resolution images of the lips are always able to be available. And, most AVSR assumed that the voice activity is given in advance. Thus, their methods have difficulties in applying them to robots.

To solve the difficulties, we proposed a ***two-layered AV integration framework*** [6]. This framework applies audio-visual integration both to voice activity detection (the first layer AV integration) and to speech recognition (the second layer AV integration) for improving noise robustness. The AVSR system showed high noise-robustness and high speech recognition performance in acoustically and/ or visually noisy conditions. However, the effectiveness of AV integration is shown when the image resolution is low. Thus, this integration method is of less use.

For this issue, we introduce two approaches. One is feature integration with time-lag. Visual features and audio features are not synchronized, so this method takes time-lag into account. The other is the use of pattern classification methods, that is, erosion and dilation.

The rest of this paper is organized as follows: Section 2 discusses issues in audio and visual voice activity detection, and Section 3 shows an approach for AV-VAD. Section 4 describes our automatic speech recognition system for robots using two-layered AV integration, that is, AV-VAD and AVSR. Section 5 shows evaluation in terms of VAD and ASR performance. The last section concludes this paper.

2 Issues in Audio and Visual Voice Activity Detection

This section discusses issues in voice activity detection (Audio VAD) and lip activity detection (Visual VAD) for robots and their integration (AV-VAD), because VAD is an essential function for ASR.

2.1 Audio VAD

VAD detects the start and the end points of an utterance. When the duration of the utterance is estimated shorter than the actual one, that is, the start point is detected with some delay and/or the end point is detected earlier, the beginning and the last part of the utterance is missing, and thus ASR fails. Also, an ASR system requires some silent signal parts (300-500 ms) before and after the utterance signal. When the silent parts are too long, it also affects the ASR system badly. Therefore, VAD is crucial for ASR, and thus, a lot of VAD methods have been reported so far. They are mainly classified into three approaches as follows:

A-1: The use of acoustic features,
A-2: The use of the characteristics of human voices,
A-3: The use of intermediate speech recognition results using ASR.

Common acoustic features for **A-1** are energy and Zero-Crossing Rate (ZCR), but energy has difficulty in coping with an individual difference and a dynamic change in voice volume. ZCR is robust for such a difference/change because it is a kind of frequency-based feature. On the other hand, it is easily affected by noise, especially, when the noise has power in speech frequency ranges. Therefore, a combination of energy and ZCR is commonly used in conventional ASR systems. However, it is still prone to noise because it does not have any prior knowledge on speech signals.

For **A-2**, Kurtosis or Gaussian Mixture Model (GMM) is used. This shows high performance in VAD when it is performed in an expected environment, that is, an acoustic environment for a VAD test is identical to that for GMM training. However, when the acoustic environment changes beyond the coverage of the model, VAD easily deteriorates. In addition, to achieve noise robust VAD based on these methods, a large number of training data is required.

A-3 uses the ASR system for VAD, and thus, this is called decoder-based VAD. An ASR system basically has two stages for recognition. At the first stage, the ASR system computes log-likelihood of silence for an input signal at every frame. By using the computed log-likelihood, VAD is performed by thresholding x_{dvad} defined by

$$x_{dvad} = \log(p(\omega_0|x)) \tag{1}$$

where x is audio input, and ω_0 shows the hypothesis that x is silence.

Actually, this mechanism is already implemented on open-sourced speech recognition software called "Julius"[1]. It is reported that this approach shows quite high performance in real environments Although this approach sounds like the chicken-or-egg dilemma, this result shows that integration of VAD and ASR is effective.

Thus, each method has unique characteristics, and none of them are suitable for all-purpose use. **A-1** is still commonly-used, **A-3** has the best performance.

2.2 Visual VAD

Visual VAD means lip activity detection in visual speech recognition which corresponds to audio VAD in ASR. The issues in visual VAD for integration with audio VAD and AVSR are as follows:

B-1: The limitation of frame rate,
B-2: The robust visual feature.

The first issue is derived from the hardware limitation of conventional cameras. The frame rate of a conventional camera is 30 Hz, while that of acoustic feature extraction in ASR is usually 100 Hz. Thus, when we integrate audio and visual features, a high speed camera having a 100 Hz capturing capability or a synchronization technique like interpolation is necessary.

For the second issue, a lot of work has been studied in the AVSR community so far. A PCA-based visual feature [7], and a visual feature based on width and

[1] `http://julius.sourceforge.jp/`

length of the lips[8] were reported. However, these features are not robust enough for VAD and AVSR because visual conditions change dynamically. Especially, the change in a facial size is hard to be coped with, since the facial size is directly related to facial image resolution. Thus, an appropriate visual feature should be explored further.

2.3 Audio-Visual VAD

AV integration is promising to improve the robustness of VAD, and thus, audio and visual VAD should be integrated to improve AVSR performance in the real world. In this case, we have two main issues. One is AV synchronization as described above. The other is the difference between audio and visual VAD. The ground truth of visual VAD is not always the same as that of audio VAD, because extra lip motions are observed before and after an utterance to open/close the lips. AV-VAD which integrates audio and visual VAD should take their differences into account. To avoid this problem, Murai *et al.* proposed two-stage AV-VAD [9]. First, they extract lip activity based on a visual feature of inter-frame energy. Then, they extract voice activity by using speech signal power from the extracted lip activity. However, in this case, when either the first or the second stage fails, the performance of the total system deteriorates.

In robotics, AV-VAD and AVSR have not been studied well although VAD is essential to cope with noisy speech. Asano *et al.* used AV integration for speech recognition, but their AV integration was limited to sound source localization [10]. Nakadai *et al.* also reported that AV integration in the level of speaker localization and identification indirectly improved ASR in our robot audition system [11]. However, in their cases, VAD was just based on signal power for a speaker direction which is estimated in AV sound source localization, that is, they indirectly used AV integration for VAD.

3 Improved AV-VAD

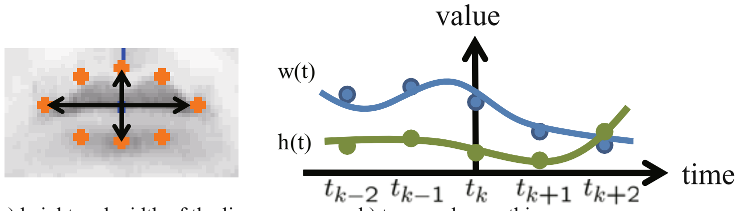
This section describes AV-VAD in our two-layered AV integration.

3.1 Audio VAD

For audio VAD, three approaches are described in the previous section, and the **A-3** approach has the best performance. Thus, we used decoder-based VAD as one of **A-3** approaches.

3.2 Visual VAD

We use a visual feature based on width and length of the lips, because this feature is applicable to extract viseme feature in the second layer of AV integration, i.e., AVSR.



a) height and width of the lips

b) temporal smoothing

Fig. 1. Visual feature extraction

To extract the visual feature, we, first, use Facial Feature Tracking SDK which is included in MindReader[2]. Using this SDK, we detect face and facial components like the lips. Because the lips are detected with its left, right, top, and bottom points, we easily compute the height and the width of the lips, and normalize them by using a face size estimated in face detection shown in Fig. 1a).

After that, we apply temporal smoothing for the consecutive five-frame height and width information by using a 3rd-order polynomial fitting function as shown in Fig. 1b). The motion of the lips is relatively slow, and the visual feature does not contain high frequency components. Such high frequency components are regarded as noise. This is why temporal smoothing is performed to remove the noise effect. Thus, we can get four coefficients such as $a_i - d_i$ for height and another four for width described as below.

$$y_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3, \tag{2}$$

where $y_i(t)$ corresponds to fitting result of height or width information ($h(t)$ or $w(t)$ in Fig. 1b). In total, eight coefficients are obtained as a visual feature vector.

For the frame rate problem mentioned in Section 2-B, we propose to perform up-sampling for the extracted eight coefficients so that they can easily synchronize with audio features. As a method of up-sampling, we used cubic spline interpolation

3.3 Audio-Visual VAD

AV-VAD integrates audio and visual features using a Bayesian network shown in Fig. 2, because the Bayesian network provides a framework that integrates multiple features with some ambiguities by maximizing the likelihood of the total integrated system. Actually, we used the following features as the inputs of the Bayesian network:

- The score of log-likelihood for silence calculated by Julius (x_{dvad}),
- Eight coefficients regarding the height and the width of the lips (x_{lip}),
- The belief of face detection which is estimated using Facial Feature Tracking SDK (x_{face}).

[2] `http://mindreader.devjavu.com/wiki`

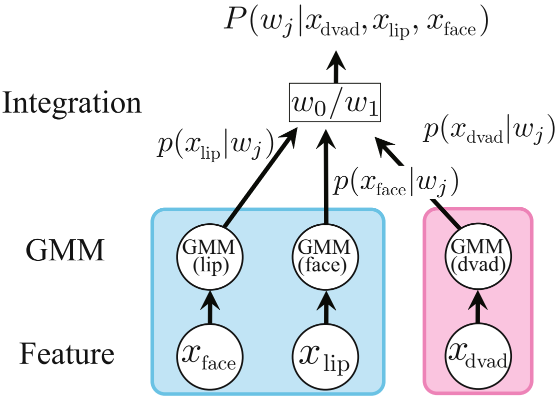


Fig. 2. AV-VAD based on a Bayesian network

Since the score of log-likelihood tend to extract voice activity with delay. So, we use the log-likelihood of previous frame, instead of current frame.

Audio and visual features have errors more or less, the Bayesian network is an appropriate framework for AV integration in VAD. The Bayesian network is based on the Bayes theory defined by

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}, \qquad j = 0, 1 \tag{3}$$

where x corresponds to each feature such as x_{dvad}, x_{lip}, or x_{face}. A hypothesis ω_j shows that ω_0 or ω_1 corresponds to a silence or a speech hypothesis, respectively. A conditional probability, $p(x|\omega_j)$, is obtained using a 4-mixture GMM which is trained with a training dataset in advance. The probability density function $p(x)$ and probability $P(\omega_j)$ are also pre-trained with the training dataset. A joint probability, $P(\omega_j|x_{dvad}, x_{lip}, x_{face})$, is thus calculated by

$$P(\omega_j|x_{dvad}, x_{lip}, x_{face}) = P(\omega_j|x_{dvad})P(\omega_j|x_{lip})P(\omega_j|x_{face}). \tag{4}$$

By comparing this probability and threshold, we estimate voice activity. Next, we perform dilation and erosion for the temporal sequence of estimated voice activity. Dilation and erosion are commonly used in pattern recognition. Fig. 3 shows the results of these processes. In dilation, a frame is added to the start-point and end-point of voice activity as below.

$$\hat{V}[k] = \begin{cases} 1 & \text{if } V[k-1] = 1 \text{ or } V[k+1] = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $V[k]$ is the estimated voice activity at k frame and $\hat{V}[k]$ is the result of dilation. In erosion, a frame is removed from the start- and end-point of voice activity as below.

$$\hat{V}[k] = \begin{cases} 0 & \text{if } V[k-1] = 0 \text{ or } V[k+1] = 0 \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

AV-VAD performs these processes several times and decides voice activity.

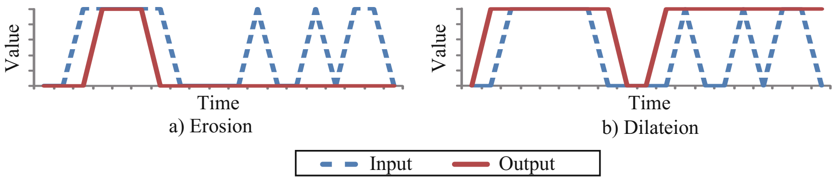


Fig. 3. Erosion and dilation

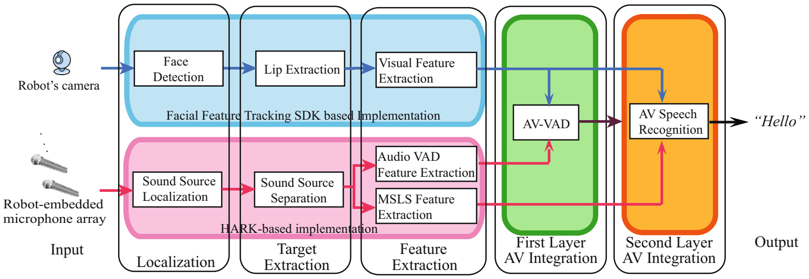

Robot's camera
Face Detection
Lip Extraction
Visual Feature Extraction
Facial Feature Tracking SDK based Implementation
Robot-embedded microphone array
Sound Source Localization
Sound Source Separation
Audio VAD Feature Extraction
MSLS Feature Extraction
HARK-based implementation
Input
Localization
Target Extraction
Feature Extraction
AV-VAD
First Layer AV Integration
AV Speech Recognition
Second Layer AV Integration
"Hello"
Output

Fig. 4. Two-Layered Audio-Visual Integration Framework

4 System Implementation

Fig. 4 shows our automatic speech recognition system for robots with two-layered AV integration, that is, AV-VAD and AVSR. It consists of four implementation blocks as follows;

- Facial Feature Tracking SDK based implementation for visual feature extraction,
- HARK-based implementation for microphone array processing to improve SNR and acoustic feature extraction,
- The first layer AV integration for AV-VAD,
- The second layer AV integration for AVSR.

Four modules in *Facial Feature Tracking SDK based implementation block* were already described in Section 3.2, and *the first layer AV integration for AV-VAD* was also explained in Section 3.3. Thus, the remaining two blocks are mainly described in this section.

4.1 HARK-Based Implementation Block

This block consists of four modules, that is, sound source localization, sound source separation, audio VAD feature extraction, and MSLS feature extraction. Their implementation is based on HARK mentioned in Section 1. The audio VAD feature extraction module was already explained in Section 3.1, and thus, the other three modules are described. We used an 8 ch circular microphone array which is embedded around the top of our robots head.

For sound source localization, we used MUltiple SIgnal Classification (MUSIC) [12]. This module estimates sound source directions from a multi-channel audio signal input captured with the microphone array.

For sound source separation, we used Geometric Sound Separation (GSS) [13]. GSS is a kind of hybrid algorithm of Blind Source Separation (BSS) and beamforming. GSS has high separation performance originating from BSS, and also relaxes BSS's limitations such as permutation and scaling problems by introducing "geometric constraints" obtained from the locations of microphones and sound sources obtained from sound source localization.

For an acoustic feature for ASR systems, Mel Frequency Cepstrum Coefficient (MFCC) is commonly used. However, sound source separation produces spectral distortion in the separated sound, and such distortion spreads over all coefficients in the case of MFCC. Since Mel Scale Logarithmic Spectrum (MSLS) [14] is an acoustic feature in a frequency domain, and thus, the distortion concentrates only on specific frequency bands. Therefore MSLS is suitable for ASR with microphone array processing. We used a 27-dimensional MSLS feature vector consisting of 13-dim MSLS, 13-dim ΔMSLS, and Δlog power.

4.2 The Second Layer AV Integration Block

This block performs AVSR. We simply introduced our reported AVSR for robots [15] as mentioned in Section 1, because this AVSR system showed high noise-robustness to improve speech recognition even when either audio or visual information is missing and/or contaminated by noises. This kind of high performance is derived from missing feature theory (MFT) which drastically improves noise-robustness by using only reliable acoustic and visual features by masking unreliable ones out. In this paper, this masking function is used to control audio and visual stream weights which are decided to be optimal manually in advance. For ASR implementation, MFT-based Julius [16] was used.

5 Evaluation

In this experiment, we used a Japanese word AV dataset. This dataset contains 10 male speech data and 266 words for each male. Audio data was sampled at 16 kHz and 16 bits, and visual data was 8 bit monochrome and 640x480 pixels in size recorded at 33 Hz. For training an AV-VAD model, we used 216 acoustically and visually clean AV data by 5 males in this AV dataset. For AVSR acoustic model training, we used 216 clean AV data by 10 males in this AV dataset.

The audio data was 8 ch data converted from 1 ch data so that each utterance comes from 0 degrees by convoluting a transfer function of the 8 ch robot-embedded microphone array. After that, we added a music signal from 60° as a noise source. The SNR changed from 20 dB to 0 dB at 10 dB increments. For the test dataset, another 50 AV data which were not included in the training dataset were selected from the synthesized 8 ch AV data.

In experiment, **C-1** AV-VAD reported by [6] and **C-2** proposed method were examined. Fig. 5 and Fig. 6 shows VAD results in various conditions using ROC

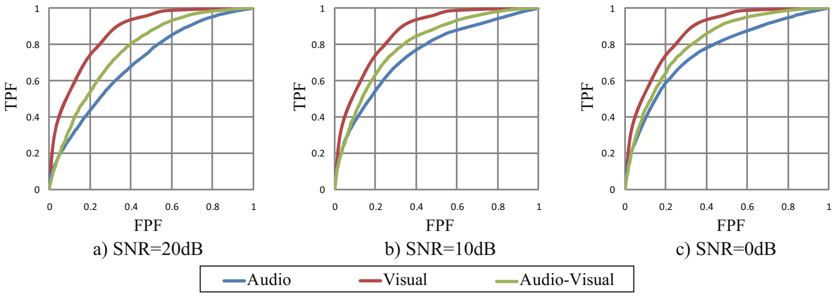


Fig. 5. The ROC curve of VAD with **C-1**

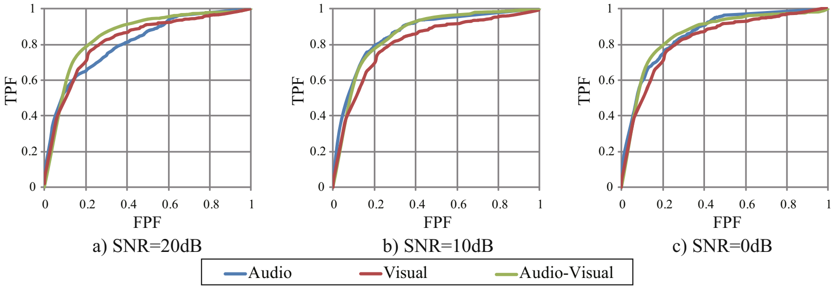


Fig. 6. The ROC curve of VAD with **C-2**(proposed method)

curves. By comparing Fig.5 and Fig. 6, we can see that proposed method improves Audio-VAD performance. The AV-VAD performances were the best in every condition with **C-2**, while Visual-VAD performances were the best with **C-1**. This result shows that the combination of proposed method and AV integration is effective in VAD.

6 Conclusion

We introduced two method for AV-VAD. And we implemented these two methods to two-layered audio-visual integration AVSR system. VAD performance was evaluated using high resolution images, and we show that our proposed method improves AV-VAD performance when the input images are high resolution.

The future work is to cope with visual noises such as reverberation, illumination, and facial orientation. In this paper, we evaluate robustness for acoustical noises and face size changes, but other dynamic changes exist in a daily environment. To apply AVSR to a robot, AVSR should be robust for such changes.

Acknowledgments

We thank Prof. R. W. Picard and Dr. R. E. Kaliouby, MIT for allowing us to use their system.

References

1. Nakadai, K., Lourens, T., Okuno, H.G., Kitano, H.: Active audition for humanoid. In: Proceedings of 17th National Conference on Artificial Intelligence, pp. 832–839 (2000)
2. Yamamoto, S., Nakadai, K., Nakano, M., Tsujino, H., Valin, J.M., Komatani, K., Ogata, T., Okuno, H.G.: Real-time robot audition system that recognizes simultaneous speech in the real world. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5333–5338 (2006)
3. Potamianos, G., Neti, C., Iyengar, G., Senior, A., Verma, A.: A cascade visual front end for speaker independent automatic speechreading. Speech Technology, Special Issue on Multimedia 4, 193–208 (2001)
4. Tamura, S., Iwano, K., Furui, S.: A stream-weight optimization method for multi-stream hmms based on likelihood value normalization. In: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 469–472 (2005)
5. Fiscus, J.: A post-processing systems to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER). In: Proceedings of Workshop on Automatic Speech Recognition and Understanding, pp. 347–354 (1997)
6. Yoshida, T., Nakadai, K., Okuno, G.H.: Automatic speech recognition improved by two-layered audio-visual speech recognition for robot audition. In: Proceedings of 9th IEEE-RAS International Conference on Humanoid Robots, pp. 604–609 (2009)
7. Liu, P., Wang, Z.: Voice activity detection using visual information. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 609–612 (2004)
8. Rivet, B., Girin, L., Jutten, C.: Visual voice activity detection as a help for speech source separation from convolutive mixtures. Speech Communication 49, 667–677 (2007)
9. Murai, K., Nakamura, S.: Face-to-talk: audio-visual speech detection for robust speech recognition in noisy environment. IEICE TRANSACTIONS on Information and Systems E86-D, 505–513 (2003)
10. Asano, F., Motomura, Y., Aso, H., Yoshimura, T., Ichimura, N., Nakamura, S.: Fusion of audio and video information for detecting speech events. In: Proceedings of International Conference on Information Fusion, pp. 386–393 (2003)
11. Nakadai, K., Matsuura, D., Okuno, H.G., Tsujino, H.: Improvement of recognition of simultaneous speech signals using av integration and scattering theory for humanoid robots. Speech Communication 44, 97–112 (2004)
12. Asano, F., Goto, M., Itou, K., Asoh, H.: Real-time sound source localization and separation system and its application to automatic speech recognition. In: Proceedings of International Conference on Speech Processing, pp. 1013–1016 (2001)
13. Valin, J.M., Rouat, J., Michaud, F.: Enhanced robot audition based on microphone array source separation with post-filter. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2123–2128 (2004)

14. Nishimura, Y., Shinozaki, T., Iwano, K., Furui, S.: Noise-robust speech recognition using multi-band spectral features. Acoustical Society of America Journal 116, 2480–2480 (2004)
15. Koiwa, T., Nakadai, K., Imura, J.: Coarse speech recognition by audio-visual integration based on missing feature theory. In: Proceedings of IEEE/RAS International Conference on Intelligent Robots and Systems, pp. 1751–1756 (2007)
16. Nishimura, Y., Ishizuka, M., Nakadai, K., Nakano, M., Tsujino, H.: Speech recognition for a humanoid with motor noise utilizing missing feature theory. In: Proceedings of 6th IEEE-RAS International Conference on Humanoid Robots, pp. 26–33 (2006)

Robust Ego Noise Suppression of a Robot

Gökhan Ince[1,3], Kazuhiro Nakadai[1,3], Tobias Rodemann[2],
Hiroshi Tsujino[1], and Jun-Ichi Imura[3]

[1] Honda Res. Inst. Japan Co., Ltd. 8-1 Honcho, Wako-shi, Saitama 351-0188, Japan
{gokhan.ince,nakadai,tsujino}@jp.honda-ri.com

[2] Honda Res. Inst. Europe GmbH, Carl-Legien Strasse 30, 63073 Offenbach, Germany
tobias.rodemann@honda-ri.de

[3] Dept. of Mech. and Env. Informatics, Grad. School of Information Science and Eng., Tokyo Inst. of Tech. 2-12-1-W8-1, O-okayama, Meguro-ku, Tokyo, 152-8552, Japan
imura@mei.titech.ac.jp

Abstract. This paper describes an architecture that can enhance a robot with the capability of performing automatic speech recognition even while the robot is moving. The system consists of three blocks: (1) a multi-channel noise reduction block comprising consequent stages of microphone-array-based sound localization, geometric source separation and post filtering, (2) a single-channel template subtraction block and (3) a speech recognition block. In this work, we specifically investigate a missing feature theory based automatic speech recognition (MFT-ASR) approach in block (3), that makes use of spectrotemporal elements that are derived from (1) and (2) to measure the reliability of the audio features and to generate masks that filter unreliable speech features. We evaluate the proposed technique on a robot using word error rates. Furthermore, we present a detailed analysis of recognition accuracy to determine optimal parameters. Proposed MFT-ASR implementation attains significantly higher recognition performance compared to the performances of both single and multi-channel noise reduction methods.

Keywords: Ego noise, noise reduction, robot audition, speech recognition, missing feature theory, microphone array.

1 Introduction

Robots with microphones (such as NEC Papero [1]) are usually equipped with adaptive noise cancellation, beamforming and acoustic echo cancellation methods for robust speech recognition in noisy environments. However, the robot's own noise, so called ego noise, can also cause mis-recognition of spoken words during an interaction with a human, even if there are no other interfering sound sources in the environment. One special type of ego noise, which is observed while the robot is performing an action using its motors, is called *ego-motion noise*. This type of interference is more difficult to cope with compared to background noise or static fan-noise of the robot, because it is non-stationary and, to a certain extent, similar to the signals of interest. Therefore, conventional noise

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 62–71, 2010.

reduction methods like spectral subtraction do not work well in practice. Several researchers tackled ego-motion noise problem by predicting and subtracting ego-motion noise using templates recorded in advance [2], [3]. Ince *et al.* [4] proposed to use a parameterized template subtraction which incorporates tunable parameters to deal with variations in the ego-motion noise. However, all those methods suffer from the distorting effects of *musical noise* [5] that comes along with nonlinear single-channel based noise reduction techniques and reduces the intelligibility and quality of the audio signal. Besides, this method has the following weakness: when applied together with a nonlinear stationary background noise prediction technique, e.g. Minima Controlled Recursive Averaging (MCRA) [6], in order to cope with the dynamically-changing environmental factors (e.g. stationary noise, reverberation), it creates a series of two consecutive nonlinear noise reduction operations. These operations produce even more musical noise, eventually causing damaged acoustic features and deteriorated recognition performance of automatic speech recognition (ASR).

It was shown that unreliable speech features degrade recognition performance severely [7]. Missing feature theory (MFT), which can be basically described as a filtering operation applied to the missing or damaged acoustic features, has already found useful applications like recognition of speech corrupted by music and several types of noise (refer to [7] for a comprehensive study) or simultaneous speech recognition of several speakers in the field of robot audition [8], [9], where they based their models of mask generation on the disturbing effect of leakage noise over speech caused by an imperfect source separation. In this work, we incorporate MFT to solve the ego-motion noise problem of a robot. To estimate the reliability of the features of speech, which is subject to residuals of motor noise after template subtraction and to improve the performance of ASR, we propose to use MFT with a model that is based on the ego-motion noise estimations. To generate suitable masks, we propose to integrate also a multi-channel framework that consists of sound source localization (SSL), sound source separation (SSS), and speech enhancement (SE), of which the first two steps make use of the directivity properties of motor noises to cancel them and thus provide additional information about the reliability assessment. In this respect, the main contribution of our work will be the incorporation of an original missing feature mask (MFM) generation method based on the signals available from two blocks (template subtraction & multi-channel noise reduction) that run in parallel. The mask relies on a measure of a frequency bin's quality calculated from the similarity of two totally different- yet complementary- approaches. Firstly, a binary mask, that uses either 0 or 1 to estimate the reliability of each acoustic feature, is suggested. We, later, enhance the proposed method further by using a soft mask represented as continuous values between 0 and 1 that yields more detailed information about the reliability. We demonstrate that the proposed methods achieve a high noise elimination performance and thus ASR accuracy.

The rest of the paper is organized as follows: Section 2 describes the proposed system and briefly summarizes the preprocessing-stages, namely SSL, SSS, SE and template subtraction. Section 3 investigates speech recognition integration and computation of the missing feature masks in detail. Conducted experiments and consecutive results are presented in Section 4. The last section gives a conclusion and future work.

2 System Overview

As sensors we use an array of multiple omnidirectional microphones mounted around the head of the robot. The overall architecture of the proposed noise reduction system is shown in Fig. 1. The first block of our processing chain, composed of elements for performing SSL, extracts the location of the most dominant sources in the environment. The estimated locations of the sources are used by a linear separation algorithm called Geometric Source Separation (GSS) [8]. It can be considered as a hybrid algorithm that exerts Blind Source Separation (BSS) [10] and beamforming. The next stage after SSS is a speech enhancement step called multichannel Post Filtering (PF). This module attenuates stationary noise, e.g. background noise, and non-stationary noise that arises because of the leakage energy between the output channels of the previous separation stage for each individual sound source. These three main modules constitute the **multi-channel noise reduction** block [11], whereas the second block performs **template subtraction** [4]. Altogether, both branches are responsible for the *audio features* for speech recognition and *spectrograms* to be processed further in the MFM generation stage. Finally, a new third block, **MFT-based speech recognition**, designed to achieve a more robust ASR uses both the features and spectrograms created in the pre-processing stages in order to extract the most suitable features. This part will be discussed in Section 3 in detail.

2.1 Multi-channel Noise Reduction System [11]

In order to estimate the Directions of Arrival (DoA) of each sound source, we use a popular adaptive beamforming algorithm called MUltiple Signal Classification (MUSIC) [12]. It detects the DoA by performing eigenvalue decomposition on the correlation matrix of the noisy signal, by separating subspaces of undesired interfering sources and sound sources of interest, and finally by finding the peaks occurring in the spatial spectrum. A consequent source tracker system performs a temporal integration in a given time window.

Geometric Source Separation [10], later on extended to be an adaptive algorithm that can process the input data incrementally [13], makes use of the locations of the sources explicitly. To estimate the separation matrix properly, GSS introduces cost functions that must be minimized in an iterative way (see [13] for details). Moreover, we use adaptive step-size control that provides fast convergence of the separation matrix [14].

After the separation process, a multi-channel post filtering operation proposed by Valin [13] is applied, which can cope with nonstationary interferences as well as stationary noise. This module treats the transient components in the spectrum as if they are caused by the leakage energies that may occasionally arise due to poor separation performance. For this purpose, noise variances of

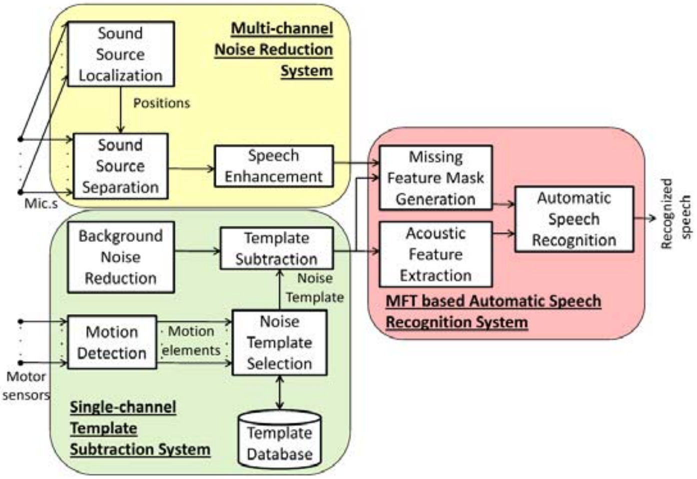

Sound Source Localization
Multi-channel Noise Reduction System
Positions
Sound Source Separation
Speech Enhancement
Mic.s
Missing Feature Mask Generation
Automatic Speech Recognition
Recognized speech
Acoustic Feature Extraction
MFT based Automatic Speech Recognition System
Background Noise Reduction
Template Subtraction
Noise Template
Motion Detection
Motion elements
Noise Template Selection
Motor sensors
Single-channel Template Subtraction System
Template Database

Fig. 1. Proposed noise cancellation system

both stationary noise and source leakage are predicted. Whereas the former one is computed using the MCRA [5] method, to estimate the latter the algorithm proposed in [13] is used.

2.2 Single-Channel Template Subtraction System [4]

During the motion of the robot, current position (θ) information from each joint is gathered regularly in the template generation (database creation) phase. Using the difference between consecutive motor position outputs, velocity ($\dot{\theta}$) values are calculated, too. Considering that J joints are active, joint position vectors with the size of $2J$ are generated. The resulting vector has the form of $F{=}[\theta_1, \dot{\theta}_1, \theta_2, \dot{\theta}_2 \ldots, \theta_J, \dot{\theta}_J]$. At the same time, motor noise is recorded and the spectrum of the motor noise is calculated in parallel with motion element acquisition. Both joint position vectors and spectra are continuously labeled with time tags so that they can be synchronized. Finally, a large noise template database that consists of short noise templates for desired joint configurations is created. In the prediction phase a nearest neighbor search in the database is conducted for the best matching template of motor noise for the current time instance (frame at that moment) using the joint-status vectors. The templates are used as weights inside the spectral subtraction routine.

3 MFT-Based Automatic Speech Recognition System

Different strategies, which make use of a confidence-based weighting of the time-frequency representation of audio signals, can enhance the quality of speech. As stated in [7], Missing Feature Theory is a very promising approach that basically applies a mask to decrease the contribution of unreliable parts of distorted speech. By keeping the reliable parameters that are essential for speech recognition, a substantial increase in recognition accuracy is achieved [8], [9]. In this section, we will discuss the basic steps of such an ASR system and how this approach can be adapted to fit to the ego-motion noise problem by presenting a robust mask design method for estimating reliability of speech based on the current motor noise.

3.1 Acoustic Feature Extraction

Acoustic features are extracted from the refined spectrum, which is the final product of the noise reduction stage (See Fig. 1). Because we do not want to have the distortions spreading to all coefficients of the cepstrum, we avoided the usage of Mel-Frequency Cepstral Coefficients (MFCC) in contrast to conventional ASR systems. Instead, we used the Mel-Scale Log Spectrum (MSLS), whose detailed calculation method can be found in [15]. Moreover, linear regression of each spectral coefficient is represented as a *delta* feature and it is used to enhance the quality of acoustic features. A consequent stage of spectral mean normalization improves noise robustness of MSLS features by subtracting the average of the features in the last 5 sec. from the current features.

3.2 MFM Generation

The reliability of features is computed for each frame and for each mel-frequency band. If continuous values between 0 and 1 constitute the mask, it is called a *soft mask*. On the other hand, a *hard mask* contains only discrete values, either 0 or 1. In this paper, we used both methods to assess their performance for this particular type of ego-noise problem. We start by explaining some underlying findings about the ego-motion noise suppression capabilities of the preprocessing stages of our proposed system in the next two paragraphs as a motivation. Then, we show how to derive the masks later in detail.

GSS lacks the ability to catch motor noise originating from the same direction of the speaker and suppress it, because the noise is considered as part of the speech. Moreover, when the position of the noise source is not detected precisely, GSS cannot separate the sound in the spatial domain. As a consequence, motor noise can be spread to the separated sound sources in small portions. However, multi-channel noise suppression systems work very well for weaker motion noises like arm or leg motions compared to head motion noise, as we found out in our experiments [11]. Additionally, it is optimally designed for "simultaneous multiple speakers" scenarios with background noise and demonstrates a very good performance when no motor noise is present.

On the other hand, template subtraction does not make any assumption about the directivity or diffuseness of the sound source and can match a pre-recorded template of the motor noise at any moment. The drawback of this approach is, however, due to the non-stationarity, the characteristics of predicted and actual noise can differ to a certain extent.

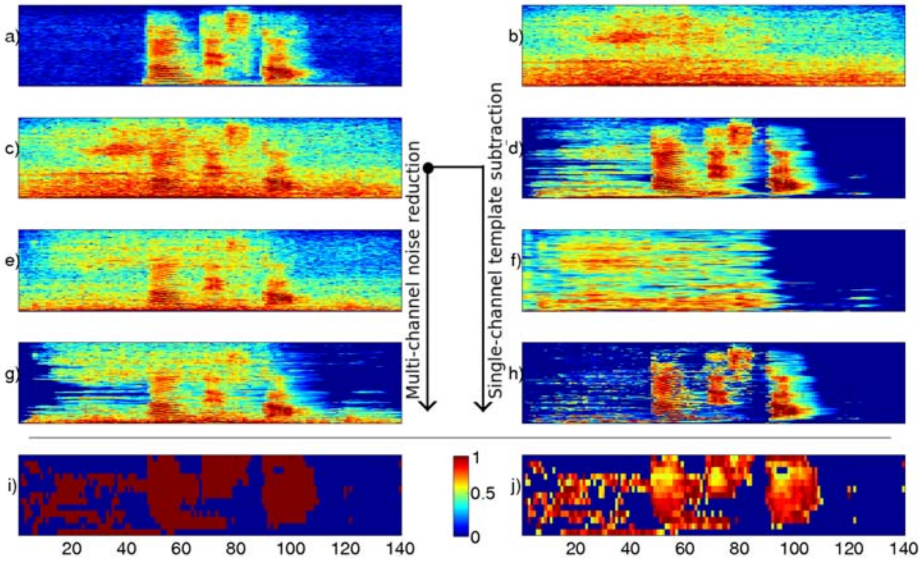


Fig. 2. Spectrogram of (a) clean speech, (b) motor noise + background noise, (c) noisy speech (**a**+**b**), (d) background noise reduction (MCRA) applied to **c**, (e) GSS applied to **c**, (f) extracted template for template subtraction, (g) PF applied to **e**, (h)template subtraction applied to **d** using **f**, (i) hard mask generated using **g** and **h**, (j) hard mask generated using **g** and **h**. In (a)-(h), y-axis represents frequency bins between 0 and 8kHz, in (i)-(j) 13 static mel-features are represented in y-axis. x-axis represents in all panels the index of frames.

As we have stated, the strengths and weaknesses of both approaches are distinct and thus can be used in a complementary fashion. A speech feature is considered unreliable, if the difference between the energies of refined speech signals generated by multi-channel and single-channel noise reduction systems is above a threshold T. Computation of the masks is performed for each frame, k, and for each frequency band, f. First, a continuous mask is calculated like following:

$$m(f,k)=\frac{|\hat{S}_m(f,k)-\hat{S}_s(f,k)|}{\hat{S}_m(f,k)+\hat{S}_s(f,k)}, \tag{1}$$

where $\hat{S}_m(f,k)$ and $\hat{S}_s(f,k)$ are the estimated energy of the refined speech signals, which were subject to multi-channel noise reduction and resp. single-channel template subtraction. Both signals are computed using a mel-scale filterbank. The numerator term represents the deviation of the two outputs, which is a measure of the uncertainty or unreliability. The denominator term, however, is a scaling constant and is given by the average of the two estimated signals. (To simplify the equation, we remove the scalar value in the denominator, so that $m(f,k)$ can take on values between 0 and 1.) Depending on the type of the mask (hard or soft) used in the MFT-ASR, Eq.(2) or Eq.(3) is selected.

1. For hard (binary) mask:

$$M(f,k)=\begin{cases}1, & \text{if } m(f,k) < T \\ 0, & \text{if } m(f,k) \geq T\end{cases}. \tag{2}$$

2. For soft mask [9]:

$$M(f,k)=\begin{cases}\dfrac{1}{1+\exp(-\sigma(m(f,k)-T))}, & \text{if } m(f,k) < T \\ 0, & \text{if } m(f,k) \geq T\end{cases}, \tag{3}$$

   where σ is the tilt value of a sigmoid weighting function.

Fig. 2 gives a general overview about the effect of each processing stage until the masks are generated. In Fig. 2c), we see a tightly overlapped speech (Fig. 2a)) and motor noise (Fig. 2b)) mixture with an SNR of -5dB. GSS+PF in Fig. 2g) reduces only a minor part of the motor noise while sustaining the speech. On the other hand, template subtraction (Fig. 2h)) reduces the motor noise aggressively while damaging some parts of the speech, where some features of the speech get distorted. The hard mask (Fig. 2i)) gives us a filter eliminating unreliable and still noisy parts of the speech (T=0.5). The soft mask (Fig. 2j)), in addition, provides more detailed information about the reliability degree of each feature so that the noise-free features are weighted more than the noise-containing parts in the MFT-ASR ($\{T,\sigma\}=\{0.5,5\}$). Furthermore, we observe that weights in the first 50 frames contaminated with noise were given either zero or low weights in the mask. Note that speech features are located between the [50 110]-th frames.

3.3 MFT-ASR

Missing Feature Theory Based Automatic Speech Recognition (MFT-ASR) is a Hidden Markov Model based speech recognition technique [7]. Suppose $M(i)$ is the MFM vector that is generated as in Sec. 3.2 for the i-th acoustic feature, the output probability can be given as follows:

$$b_j(x)=\sum_{l=1}^{L} P(l|S_j)\exp\left\{\sum_{i=1}^{I} M(i)\log f(x(i)|l,S_j)\right\}, \tag{4}$$

where $b_j(x)$ is the output probability of j-th state, $x(i)$ is denoted as an acoustic feature vector, I represents the size of the acoustic feature vector, $P(\cdot)$ is the probability operator and S_j is the j-th state. Density in each state S_j is modeled using mixtures of L Gaussians with diagonal-only covariance. Please note that when all mask values are set to 1, Eq.(4) becomes the same as the output probability calculation of a conventional ASR.

4 Results

In this section we present comparative results for pre-processing based ASR, hard and soft mask based ASR, and the influence of selected parameters for template subtraction and MFT-ASR on the performance. To evaluate the performance of the proposed techniques, we use Honda's humanoid robot ASIMO. The robot is equipped with an 8-ch microphone array on top of its head. Of the robots many degrees of freedom, we use only 2 motors for head motion, and 4 motors for the motion of each arm with altogether 10 degrees of freedom. We recorded random motions performed by the given set of limbs by storing a training database of 30 minutes and a test database 10 minutes long. Because the noise recordings are comparatively longer than the utterances used in the isolated word recognition, we selected those segments, in which all joints contribute to the noise. The noise signal consisting of ego noise (incl. ego-motion noise) and environmental background noise is mixed with clean speech utterances used in a typical human-robot interaction dialog. This Japanese word dataset includes 236 words for 4 female and 4 male speakers. Acoustic models are trained with Japanese Newspaper Article Sentences (JNAS) corpus, 60-hour of speech data spoken by 306 male and female speakers, hence the speech recognition is a word-open test. We used 13 static MSLS, 13 delta MSLS and 1 delta power. Speech recognition results are given as average WER of instances from the test set. In 'isolated word recognition' tasks, WER is an evaluation criteria alternative to Word Correct Rate (WCR), such that $WER = 100\% - WCR$ holds. The position of the speaker is kept fixed at 0° throughout the experiments. The recording environment is a room with the dimensions of $4.0\,\mathrm{m}\times 7.0\,\mathrm{m}\times 3.0\,\mathrm{m}$ with a reverberation time (RT_{20}) of 0.2s. Although the position of the original sound source was given in advance to avoid the mis-recognition due to localization errors, we did not fix the ego-noise direction of the robot. In this experiment, the SSL module predicted it automatically.

Fig. 3a) illustrates the ASR accuracies for all methods under consideration. The results are evaluated using an acoustic model trained with MCRA-applied speech data, except GSS+PF method for which we used a matched acoustic model for that condition. We evaluated MFMs for three heuristically selected threshold parameters $T = \{0.25, 0.5, 0.75\}$. In the preliminary tests we found out that the feature set that is derived at the output of template subtraction achieves higher accuracy in a range of 10 to 20% in WER compared to the features after multi-channel noise reduction. So, we concluded that the former feature type is more suitable to be used in an MFT-ASR. Single-channel results are used as a baseline. MFM-ASR outperforms both mere single (TS) & multi-channel (GSS+PF) noise reduction methods. In the case of $T<0.5$, indicating a mask model based on a reliability estimation exerting sharp differentiation of both evidences of noise reduced spectra was unable to improve the ASR, because essential features belonging to the speech are thrown away, thus WERs deteriorate. On the contrary, higher thresholds improved the outcomes significantly.

In the second part of our experiments, we compared the results of hard masking with optimal threshold ($T=0.75$) obtained in the first part of the experiments,

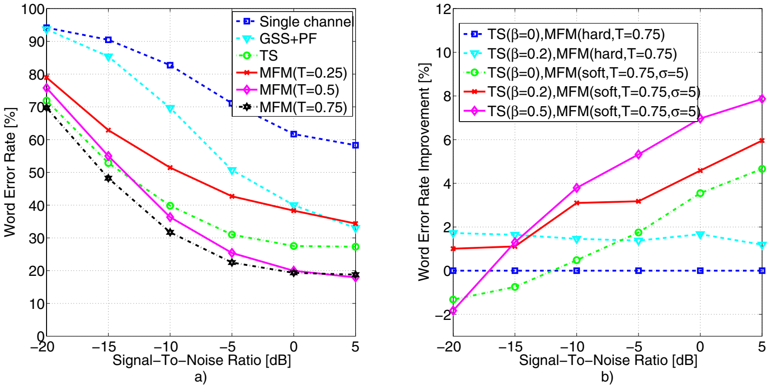


Fig. 3. Speech recognition performance for a) different processing stages b) soft mask - hard mask comparison for given parameters

to the results of soft masking for a parameter set of σ=$\{5, 10, 50\}$. All three cases with the given parameters yielded more or less the same WER improvements, however, outside this range the results become very sensible to σ and worsen eventually. Therefore, we will only present the results for σ=5. Besides, we inspected the effect of decreasing the aggressiveness level of the template subtraction, by leaving an artificial floor on the bottom of the spectra. So far, the parameter called *spectral floor* (β, where $0 \leq \beta \leq 1$) [4] was set to zero. We assess the results for β=$\{0, 0.2, 0.5\}$ in the framework of soft-hard mask comparison in Fig. 3b) by giving the WER improvement relative to the hard mask results obtained for β=0 and T=0.75. By increasing β, we observed that the WERs improve considerably. That means that a tradeoff between "noise reduction level" and "signal distortion" contributed to the mask quality substantially. Furthermore, soft masks reduce the WERs even further by up to 8% compared to hard masks. This reduction is attained due to the improved probabilistic representation of the reliability of each feature. Optimal results are obtained when we use a soft mask with the following parameter set: $\{T, \sigma, \beta\}$=$\{0.75, 5, 0.5\}$.

5 Summary and Outlook

In this paper we presented a method for eliminating ego-motion noise from speech signals. The system we proposed to utilize (1) a multi-channel noise reduction stage, (2) a template subtraction scheme, and finally (3) a masking stage to improve speech recognition accuracy. We used an MFM model, which is based on the similarity measurements of ego-motion noise estimations gathered from (1) and (2). We validated the applicability of our approach by evaluating its performance for different settings for both hard and soft MFM. Our method demonstrated significant WER improvement for hard masking (45% relative to single-channel recognition) and soft masking (up to 53%).

In future work, we plan to find an optimized parameter set for template subtraction and especially for MFM-ASR block in a wider range. The next step is an evaluation in real time and a real situation, which involves speech recognition of several speakers simultaneously while the robot is performing some motion.

References

1. Sato, M., Sugiyama, A., Ohnaka, S.: An adaptive noise canceller with low signal-distortion based on variable stepsize subfilters for human-robot communication. IEICE Trans. Fundamentals E88-A(8), 2055–2061 (2004)
2. Nishimura, Y., Nakano, M., Nakadai, K., Tsujino, H., Ishizuka, M.: Speech Recognition for a Robot under its Motor Noises by Selective Application of Missing Feature Theory and MLLR. In: ISCA Tutorial and Research Workshop on Statistical And Perceptual Audition (2006)
3. Ito, A., Kanayama, T., Suzuki, M., Makino, S.: Internal Noise Suppression for Speech Recognition by Small Robots. In: Interspeech, pp. 2685–2688 (2005)
4. Ince, G., Nakadai, K., Rodemann, T., Hasegawa, Y., Tsujino, H., Imura, J.: Ego Noise Suppression of a Robot Using Template Subtraction. In: Proc. of the IEEE/RSJ International Conference on Robots and Intelligent Systems (IROS), pp. 199–204 (2009)
5. Cohen, I.: Noise Estimation by Minima Controlled Recursive Averaging for Robust Speech Enhancement. IEEE Signal Processing Letters 9(1) (2002)
6. Boll, S.: Suppression of Acoustic Noise in Speech Using Spectral Subtraction. IEEE Transactions on Acoustics, Speech, and Signal Processing ASSP-27(2) (1979)
7. Raj, B., Stern, R.M.: Missing-feature approaches in speech recognition. IEEE Signal Processing Magazine 22, 101–116 (2005)
8. Yamamoto, S., Nakadai, K., Nakano, M., Tsujino, H., Valin, J.M., Komatani, K., Ogata, T., Okuno, H.G.: Real-time robot audition system that recognizes simultaneous speech in the real world. In: Proc. of the IEEE/RSJ International Conference on Robots and Intelligent Systems, IROS (2006)
9. Takahashi, T., Yamamoto, S., Nakadai, K., Komatani, K., Ogata, T., Okuno, H.G.: Soft Missing-Feature Mask Generation for Simultaneous Speech Recognition System in Robots. In: Proceedings of International Conference on Spoken Language Processing (Interspeech), pp. 992–997 (2008)
10. Parra, L.C., Alvino, C.V.: Geometric Source Separation: Merging Convolutive Source Separation with Geometric Beamforming. IEEE Trans. Speech Audio Process 10(6), 352–362 (2002)
11. Ince, G., Nakadai, K., Rodemann, T., Hasegawa, Y., Tsujino, H., Imura, J.: A Hybrid Framework for Ego Noise Cancellation of a Robot. In: Proc. of the IEEE/RSJ International Conference on Robotics and Automation, ICRA (to appear, 2010)
12. Schmidt, R.: Multiple emitter location and signal parameter estimation. IEEE Trans. on Antennas and Propagation 34(3), 276–280 (1986)
13. Valin, J.-M., Rouat, J., Michaud, F.: Enhanced Robot Audition Based on Microphone Array Source Separation with Post-Filter. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 2123–2128 (2004)
14. Nakajima, H., Nakadai, K., Hasegawa, Y., Tsujino, H.: Adaptive step-size parameter control for real-world blind source separation. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 149–152 (2008)
15. Nishimura, Y., Shinozaki, T., Iwano, K., Furui, S.: Noise-robust speech recognition using multi-band spectral features. In: Proc. of 148th Acoustical Society of America Meetings, vol. 1aSC7 (2004)

Integrating a PDDL-Based Planner and a PLEXIL-Executor into the Ptinto Robot

Pablo Muñoz[1], María D. R-Moreno[1], and Bonifacio Castaño[2]

[1] Departamento de Automática, Universidad de Alcalá
28805 Alcalá de Henares (Madrid), Spain
{pmunoz,mdolores}@aut.uah.es

[2] Departamento de Matemáticas, Universidad de Alcalá
28805 Alcalá de Henares (Madrid), Spain
bonifacio.castano@uah.es

Abstract. Ptinto is a prototype of a hexapod robot for exploration tasks in rocky and cumbersome areas. The main objective of this prototype is to design and test a complex kinematic control system, including both new hardware and software technologies.

In this paper we describe the autonomous architecture that we have developed for the control of the system. We use a deliberator that must be able to make a safe trajectory between two or more points avoiding obstacles. When a trajectory has been created, the executive takes this plan to control Ptinto via the hardware abstraction layer. This is a classical 3T architecture implementation with two general purpose systems: a PDDL planner as the deliberator, and a PLEXIL executor.

Keywords: Autonomous agents, intelligent system, planning and scheduling, intelligent execution, robotics.

1 Introduction

Since the born of robotics to the present-day there has been a continuous progress in software implementation and new hardware architectures. This progress implies, for instance, that most of the production chains in industry are robotized. In addition to that, there are several robots exploring our Solar System as well. In the last decade we can emphasize some examples such as the NASA Mars Exploration Rovers (MER) [1, 2]. This project was launched in 2003 and, with 90 days of lifetime in Mars, it is currently operative.[1]

In the same way, the software to control these complex hardware systems has made a great progress towards more autonomous and reliable systems. Although exploration robots are tele-operated and have a minimum degree of autonomy, there are several systems as: Intelligent Distributed Execution Architecture (IDEA) [3], Teleo-Reactive EXecutive (T-Rex) [4] or LAAS [5] focused on making full autonomous control architectures.

[1] http://marsrover.nasa.gov/home/index.html

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 72–81, 2010.

Space investigation is a very expensive task, so that, before sending an exploration robot to an unknown environment, it has to be tested thoroughly in similar conditions that are expected at its destination. In Spain there is a location: the Tinto river environment, that has been considered one of the few places on Earth which presents ground conditions very similar to the forecast for the Mars planet: is dense, rich in heavy metals and with limited oxygen. To take advantage of this situation and in order to study that special area, the Astrobiology Centre (CAB-INTA) in Spain is working on a terrestrial exploration hexapod robot called Ptinto. This kind of legged robots offer better mobility on difficult terrains and are more capable to overcome obstacles than the traditional rovers [7]. For example, the MER-A Spirit rover is embedded in soft sand since April 23th of this year and it is not expected to be released until next year.[2] Ptinto has six pods with three degrees of freedom each and three extensiometric gauges bridges per pod. Theses bridges are able to detect obstacles by collision and this info represents the only knowledge the robot can get form its environment so far.

This paper is an extension of our previous work [8] where we developed a basic 3T [6] architecture for the control of the locomotion of Ptinto. This time we presents a more complex autonomous architecture using two general purpose systems: a PDDL planer called SGPlan$_6$ [9] as the deliberator system and the couple PLEXIL language and the Universal Executive [10, 11] as the executor. SGplan$_6$ was the winner of the IPC-8[3] competition under the temporal satisficing track. This has been the main motivation to select it as well as its ability to deal with time, resources and metrics (some of the other planners we were testing could not find any solution to the problems we modelled). PLEXIL has been chosen because it is a general purpose language for execution systems and the UE has a general interface module adaptable to communicate the executor system with our own functional layer.

This new version of the architecture is focused on extending the skills of our system using a more complex execution system. PLEXIL and the UE make easier to incorporate the treatment of time (not present in the previous version), run sequences of commands with their respective monitoring and introduce a fault protection ability. We have also extended the PDDL domain and problem using some features of PDDL3 with variable durations on each action and a new 3-D abstraction of the terrain for the problem and subgoals. The previous version only could manage 2-D terrains and there were not durations in the actions.

The paper is structured as follows. Next section describes the deliberator layer. Section 3 presents the executive language and the associated executor. Then, in section 4 the full architecture of the autonomous control system is described. Finally, we outline some future research lines and the conclusions of our work.

[2] Updates on the efforts to free the Spirit rover in:
`http://www.nasa.gov/mission_pages/mer/freespirit.html`

[3] `http://ipc.informatik.uni-freiburg.de`

2 Deliberator Layer: PDDL Based-Planner

In a typical three tier architecture or 3T there is a top level entity typically called deliberator. The deliberator is a high level system designed to have an "intelligent" behaviour. That implies that the traditional programs are not capable enough to be used under this layer. That is the reason why there have been built domain independent planners to be integrated in these type of architectures.

There are several kinds of planners, some of them are specific and others are generic purpose ones. In this case we have chosen a general purpose planner that uses the Planning Domain Definition Language (PDDL) version 2.1 [12] and common features of PDDL3 [13]. The reason to select a general purpose planner with a standard language is the possibility of replacing the deliberator with a more powerful planner with low cost. The same domain/problem files can be used without modification, and, if the planner support new features of PDDL or extensions (like PIPSS [14], which implements resources as a PDDL extension), we will be able to extend the possibilities of the deliberator system in a short period of time.

In these systems time plays a fundamental role. With PDDL 2.1 we can determinate how long each action will take and establish a maximum duration and consistently decide whether a plan is feasible or not. In addition, the time taken for delivering must be minimized in order to respond in bounded time intervals. The treatment of time has been attended on the architectures mentioned in the introduction: IDEA and T-Rex use a timeline based system. This approach defines both a look-ahead window over which to deliberate and a timeline for each reactive component. In that sense, our system is more similar to the LAAS architecture: the deliberator generates a plan in which every action has a duration and the temporal executor decides when to start or stop these actions. In our case the executor does not modify the sequence of actions, but it executes them concurrently when possible.

PDDL planners are systems that use two input files to represent their knowledge base. One of the files contains a description of the actions that represent "what it can/(cannot) be done" and the other file includes the three elements which define the problem: the known objects of the world, the initial state and the goals we want to achieve. With this information, the planner searches a sequence of actions that can reach the goals from the initial state. The optimal solution of the problem is a conjunction of two factors: the metric used and the resolution algorithm. For the current version of our architecture, we have chosen SGPlan version 6 as the deliberator layer. It is important to mention that SGPlan$_6$ [9] uses parallel decomposition to solve the problem.

3 Executor System: PLEXIL and Universal Executive

An executor is a system that runs between the deliberator and the functional layer. Its purpose is to read the high level orders generated by the deliberator and translate them into a sequence of orders that the functional layer can execute.

However, this is only the first step, because each order in the high level can correspond to one or more in the functional level. At the same time, the executor has to supervise the execution of these orders, because if an error arises during the execution of any of them, this error may provoke that the rest of orders not be executed. In these situations the system will most likely fall into a wrong state or, in the worse case, the entire system will fails. Then, when an error happens the executor must have a response to control that event. In some cases it will try to execute an alternative order, repair the plan or use a redundant system. In other cases it will call back the deliberator with the information about the current state so that it can return a new plan.

For the execution system we have used the couple PLan EXecution Interchange Language (PLEXIL) and its associated executor, the Universal Executive (UE). This project[4] has been developed thanks to the efforts of NASA Ames Research Center, NASA Jet Propulsion Laboratory and Carnegie Mellon University and it has been distributed under the Berkeley Software Distribution license. It has been successfully applied to operate prototype planetary rovers and drilling equipments, and demonstrate automation for the International Space Station operations.

PLEXIL is a structured language to make flexible and reliable command execution plans. It is portable (since it uses XML for the encoding), lightweight, deterministic (given the same sequence of events from the external world) and very expressive. UE is the interpreter, designed to facilitate the inter-operability of execution and planning frameworks. It has an interface module to connect the executor with an external system. This interface can be modified for each particular system.

The UE executes PLEXIL's plans that must have been designed to perform one or more tasks. A PLEXIL plan is a tree of nodes which represents a hierarchical breakdown of tasks. Each node has a set of conditions to control its execution and a section called *body* that describes its function (type of node). With all these elements we can express if-then branches, loops and batch processes and use subplans that would be executed only when a special situation occurs. For instance, there is a plan that manage the fault protection system.

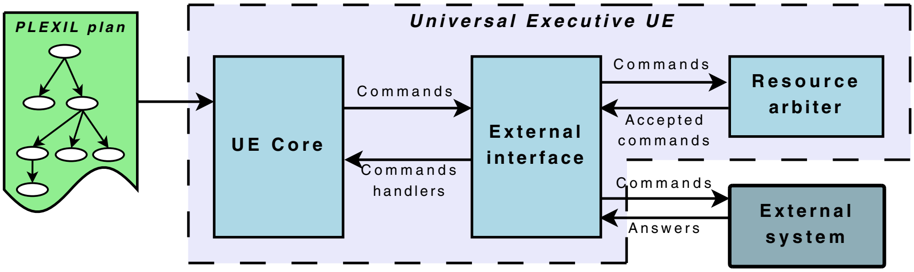


Fig. 1. PLEXIL and the UE

[4] http://plexil.sourceforge.net

The UE executor can be divided into three modules (see Fig. 1). First, the UE core that implements the PLEXIL syntax and the algorithms to execute the plans. Then, there is a module for the management of the resources in case there were conflicts when they were used. Finally, an external interface system to connect the executor with an external system. The UE core algorithm processes each node when it reaches its depth in the tree structure and its starting conditions are fulfilled. When an internal or an external state changes, this change is propagated in cascade until there are no more nodes that can be executed.

4 Control System Architecture

To deal with the Ptinto locomotion control we have implemented a typical 3T architecture. As we have described in the previous sections, the deliberator is a PDDL-like planner (SGPlan$_6$) and the executor is a PLEXIL interpreter (the UE). The functional layer is a C++ specific module for the Ptinto hardware.

These elements are assembled within a main control program written in C++ to interconnect and coordinate all the modules. The flow of the system is expressed below:

1. The control system invokes the PDDL planner with the problem (that contains the model of the ground and the goals to achieve) and the domain files (with the actions that the robot can perform) as inputs. The problem can be manually generated either by a human operator or automatically by a specific system (i.e. an AI vision system). As a result the planner generates a viable plan to reach the objectives avoiding the obstacles.
2. Once the plan has been produced, the executor reads the actions, one by one, and executes them. Each action of the planner can be decomposed in one or more low level commands.
3. Commands are sent from the executor to the functional layer through the executor interface and they are permanently monitored to check whether they are properly executed or not.
4. The functional layer is directly connected to the hardware of the robot and carries out each command issued by the executor. Once the order has been accomplished, the functional layer sends a feedback response to the executor with the outcome of the command.
5. If a command is not properly executed, the executor will try to solve the situation by calling the deliberative layer to replan again. In case this strategy cannot solve the problem, the system would wait for human intervention.

Figure 2 shows the functional view of the architecture, the data and actions involved in each layer and the inputs and outputs that interconnect them. Next subsections explain each one in detail.

4.1 High Tier: PDDL-Like Planner

The high tier is the module in charge of both: generating a plan either when the goals are set for the first time or when the initial plan cannot be reached,

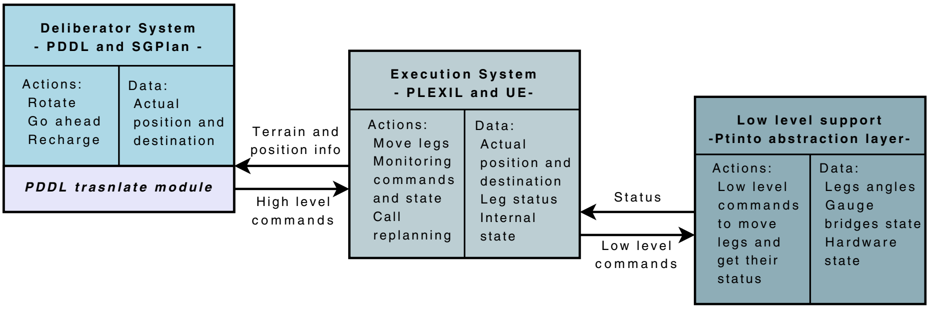


Fig. 2. Functional view of the control system architecture

and updating the problem information. The plan generation is based on the domain and problem definition. The domain definition contains all the actions the robot can perform. This information is static and must be set before Ptinto runs in autonomous mode. The problem definition represents all the information about the world: terrain data (a grid of squares that have aridity and altitude), obstacles, recharge points and data about the state of Ptinto: position, orientation and remaining energy. This information can dynamically change along the time so it is important to keep it updated. The left part of the Fig. 3 shows a conceptual view of a possible scenario. Ptinto is represented as a gray rectangle takin up one square. Each square is defined by its height and degree of aridity. The grid of squares allows us to place elements in the different squares without using a global position coordinates system that would be very complex to be modelled in PDDL. Elements that can be placed in the squares are: obstacles that Ptinto cannot overpass (red elements in the figure), energy recharge points (a blue and white square), points that we want to visit and the goal position (the green arrow). The right part of the Fig. 3 shows also a small portion of the domain and the problem in the PDDL syntax.

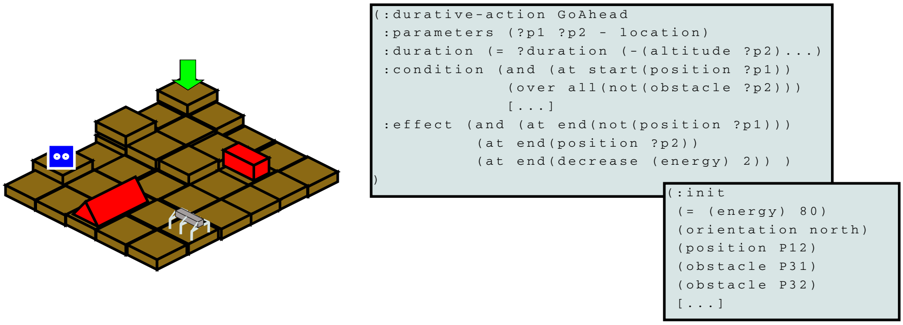
The actions defined in the domain for Ptinto are as follows:

- *Rotate*: changes the orientation of the robot. The possible orientations are: north, south, east and west. The duration of this action is calculated as shown in Eq. 1 where *angle* is the rotation angle in degrees.

$$Duration = \frac{3 \cdot angle}{90} \quad . \tag{1}$$

- *Go ahead*: moves the robot one square in the current direction. This action implies both the difference in height between squares is less than a previously established value and there is no obstacle in the destination square. The cost of this action is a function of the difference in altitude between squares and the aridity of terrain. The right part of Fig. 3 shows how this action is coded in PDDL. The duration for this actions is shown in Eq. 2 where *C1* and *C2* are the origin and destination squares respectively.

$$Duration = (C2\,altitude - C1\,altitude)^2 + C1\,aridity + C2\,aridity + 1 \quad . \tag{2}$$



```
(:durative-action GoAhead
 :parameters (?p1 ?p2 - location)
 :duration (= ?duration (-(altitude ?p2)...)
 :condition (and (at start(position ?p1))
                 (over all(not(obstacle ?p2)))
                 [...]
 :effect (and (at end(not(position ?p1)))
              (at end(position ?p2))
              (at end(decrease (energy) 2)) )
)
```

```
(:init
 (= (energy) 80)
 (orientation north)
 (position P12)
 (obstacle P31)
 (obstacle P32)
 [...]
```

Fig. 3. Graphical representation of how the world has been modelled

- *Recharge*: this action can only be performed if the square has a recharge point. Recharge implies either waiting for a battery recharge or substitution by a human action. How long this action takes is manually added to the problem file.

The output generated by this tier is a sequence of actions that defines the movements that Ptinto has to carry out to accomplish the path between the initial and the goal points. This trajectory will avoid the obstacles and will minimize the energy or the total plan time.

4.2 Middle Tier: PLEXIL Executor

The intelligent executor, or middle tier, is responsible for: reading the high level orders, transforming them into a sequence of low level orders and safely executing them in the correct order. The last part implies that the executor should be monitoring closely whether all the commands are accurately accomplished.

PLEXIL works as follow: first of all we model a PLEXIL plan (a tree of nodes that represent a hierarchy breakdown of tasks) and send that plan to the UE which interprets and executes it. Then, we have to implement the external interface of the UE to link the executor and the functional layer together in order to send commands and get the answers. The connection between the executor and the planner is different. Since the executor cannot read the output of the planner, this information comes through the PDDL translate module. Also, when the executor needs to modify the PDDL problem file (to include both the new ground information and the current position and orientation of the robot), it sends that information to the PDDL translate module and it modifies the archive with the problem accordingly.

The modelled plan should control the moves of the legs, maintain the current position of the robot, and keep the information about the terrain. To accomplish these tasks, the PLEXIL plan is divided in four interconnected plans as seen in Fig. 4:

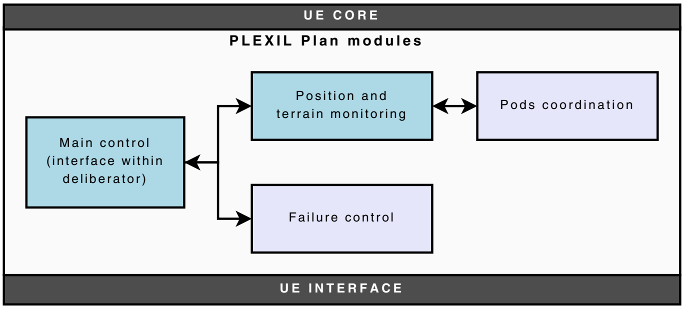

UE CORE
PLEXIL Plan modules
Position and terrain monitoring
Pods coordination
Main control (interface within deliberator)
Failure control
UE INTERFACE

Fig. 4. Modules of PLEXIL plan

- *Main control*: This is the general plan that controls the other subplans. Its function is to read and interpret the high level actions inside the plan generated by the high tier. The necessary information to execute a movement is made up of: the action type (move, rotate or recharge), the origin and destination positions (for moves only), the current and new orientation (for rotations), and how long each action last. With this information the plan controls the execution of the subplans and, when necessary, it calls the high layer to get a new plan according to the new information collected from the terrain and the current position and orientation of the robot.
- *Position and terrain monitoring*: this plan has three functions. (1) To monitor the position of the robot while it is moving. (2) To calculate the abrupt degree of the terrain from the information collected by the gauges and the angle of each pod. This degree sets how safely is going to be the move. (3) When a move cannot be performed, the plan uses some rules to try a new move or determine that there is an obstacle that cut off the present route.
- *Pods coordination*: the coordination of the pods provides the legal moves available for each leg when the robot have to, for instance, make a turn or go forward. One action invoked by this plan is translated into a low level set of commands that will be accepted by the functional layer. This plan is modelled to make concurrent moves when possible. Also, this module receives the results of the executed orders while it is monitoring the correct execution of the movements.
- *Failure control*: when an unknown failure is detected, this plan tries to determine what has failed and try to solve the problem. If there is not routine able to handle the situation, this subplan activates the operator mode and waits for human intervention.

4.3 Low Tier: Ptinto Hardware Control

The Ptinto low tier provides a simple and high level access to the hardware components such as the lineal actuators, the CAN bus or the extensiometric gauges. It also implements some features that guarantee that the access to the hardware is made in safe conditions. This is done with the typical programming techniques such as semaphores and correct synchronization of threads. For example, it checks if two or more move commands are sent to one linear actuator at the same time, or write a new message in the bus when there is another that has not been read.

Another functionality is to provide the commands execution results. This information represents the current state of each leg. That is, the angle of the three lineal actuators and the value of the three extensiometric gauges.

To test the whole architecture we are developing a full environment simulator. At present, the simulator is a basic program that sends the answers through an emulated bus with the angle and gauge values associated to each pod.

5 Conclusions and Future Work

This paper has described the autonomous architecture for the control of the Ptinto robot. The architecture integrates third party software, such as SGPlan$_6$ for the deliberative layer and the PLEXIL system for the executor, into a system that solves a concrete problem: the movility of Ptinto.

The purpose of this work is to integrate a 3T architecture with the possibility to extend the funcionality of the system, if some few changes are needed, and apply it to Ptinto. This means that replacing the PDDL-based planner in the deliberative layer or the PLEXIL executor with a different one, could be done without difficulty. As a consequence, the developed architecture would be fully reusable, and it could be easily adapted to a new layout with lower cost and less time effort than it will take building a new one from scratch.

It is important to mention that some of the decisions performed in the control architecture have been conditioned by the lack of visual sensors in the robot. That is the reason why we have designed a modular architecture that easily will allows us to introduce new developments when any improvement was set up in Ptinto layout. For instance, a vision camera or an ultrasound sensor, will allow us to build an artificial vision system that can automatically generate the terrain information where Ptinto should move. So that, our future work is related to the new hardware that may be included into Ptinto.

Acknowledgements

The authors want to thank Javier Gomez de Elvira, Julio J. Romeral-Planelló and Sara Navarro-López from the CAB-INTA for their support with the Ptinto robot. This work is funded by the Castilla-La Mancha project PEII09-0266-6640.

References

[1] Bresina, J., Jonsson, A., Morris, P., Rajan, K.: Activity Planning for the Mars Exploration Rovers. In: Procs. of the Workshop on Scheduling and Planning Applications of the International Conference on Automated Planning and Scheduling (ICAPS), Monterey, CA, USA (2005)

[2] Bresina, J., Morris, P.: Mission Operations Planning: Beyond MAPGEN. In: Procs. of the Second IEEE International Conference On Space Mission Challenges for Information Technology, Pasadena, CA, USA (2006)

[3] Aschwanden, P., Baskaran, V., Bernardini, S., Fry, C., R-Moreno, M.D., Muscettola, N., Plaunt, C., Rijsman, D., Tompkins, P.: Model-Unified Planning and Execution for Distributed Autonomous System Control. In: Procs. of the AAAI 2006 Fall Symposia. Washington, DC, USA (October 2006)

[4] McGann, C., Py, F., Rajan, K., Thomas, H., Henthorn, R., McEwen, R.: A Deliberative Architecture for AUV Control. In: Procs. of the Conference on Robotics and Automation, Pasadena, CA, USA (2008)

[5] Ingrand, F., Lacroix, S., Lemai-Chenevier, S., Py, F.: Decisional Autonomy of Planetary Rovers. Journal of Field Robotics 24(7), 559–580 (2007)

[6] Gat, E.: Three-Layer Architectures. In: Kortenkamp, D., Bonasso, R.P., Murphy, R. (eds.) Mobile Robots and Artificial Intelligence, pp. 195–210. AAAI Press, Menlo Park (1998)

[7] Smith, T.B., Barreiro, J., Chavez-Clemente, D., Smith, D.E., SunSpiral, V.: ATHLETEs Feet: Multi-Resolution Planning for a Hexapod Robot. In: Procs. of the Workshop on Scheduling and Planning Applications of the International Conference on Automated Planning and Scheduling (ICAPS), Sydney, Australia (September 2008)

[8] Muñoz, P., R-Moreno, M.D., Gómez-de-Elvira, J., Navarro, S., Romeral, J.: An Autonomous System for the Locomotion of a Hexapod Exploration Robot. In: Procs. of the third IEEE International Conference on Space Mission Challenges for Information Technology (SMC-IT 2009), Pasadena, CA, USA (July 2009)

[9] Hsu, C.W., Wah, B.W.: The SGPlan Planning System in IPC-6. In: Sixth International Planning Competition, Sydney, Australia (Sepember 2008)

[10] Universities Space Research Association (USRA). PLEXIL and Universal Executive Reference Manual, Tech. Rep. 0.90, NASA (June 2009)

[11] Verma, V., Jónsson, A., Pasareanu, C., Iatauro, M.: Universal Executive and PLEXIL: Engine and Language for Robust Spacecraft Control and Operations. In: Procs. of the American Institute of Aeronautics and Astronautics Space 2006 Conference, San Jose, CA, USA (September 2006)

[12] Fox, M., Long, D.: PDDL 2.1: An Extension to PDDL for Expressing Temporal Planning Domains. Journal of AI Research 20, 61–124 (2003)

[13] Gerevini, A., Long, D.: Plan Constraints and Preferences in PDDL3. Tech. Rep., Dept. of Electronics for Automation, University of Brescia, Italy (August 2005)

[14] Plaza, J., R-Moreno, M.D., Castaño, B., Carbajo, M., Moreno, A.: PIPSS: Parallel Integrated Planning and Scheduling System. In: Procs. of the 27th Annual Workshop of the UK Planning and Scheduling Special Interest Group (PLANSIG 2008), Edinburgh, UK (December 2008)

Finding the Maximal Pose Error in Robotic Mechanical Systems Using Constraint Programming

Nicolas Berger[1], Ricardo Soto[1,2], Alexandre Goldsztejn[1],
Stéphane Caro[3], and Philippe Cardou[4]

[1] LINA, CNRS, Université de Nantes, France
[2] Escuela de Ingeniería Informática
Pontificia Universidad Católica de Valparaíso, Chile
{nicolas.berger,ricardo.soto,alexandre.goldsztejn}@univ-nantes.fr
[3] IRCCyN, Ecole Centrale de Nantes, France
stephane.caro@irccyn.ec-nantes.fr
[4] Department of Mechanical Engineering, Laval University, Canada
pcardou@gmc.ulaval.ca

Abstract. The position and rotational errors —also called pose errors— of the end-effector of a robotic mechanical system are partly due to its joints clearances, which are the play between their pairing elements. In this paper, we model the prediction of those errors by formulating two continuous constrained optimization problems that turn out to be NP-hard. We show that techniques based on numerical constraint programming can handle globally and rigorously those hard optimization problems. In particular, we present preliminary experiments where our global optimizer is very competitive compared to the best-performing methods presented in the literature, while providing more robust results.

1 Introduction

The accuracy of a robotic mechanical system is a crucial feature for the realization of high-precision operations. However, this accuracy can negatively be affected by position and/or orientation errors, also called pose errors, of the manipulator end-effector. A main source of pose errors is the joints clearances that introduce extra degree-of-freedom displacements between the pairing elements of the manipulator joints. It appears that the lower those displacements, the higher the manufacturing cost and the more difficult the mechanism assembly. Currently, handling this concern is a major research trend in robotics [7,9,14]. A way to deal with joints clearances is to predict their impact on the pose errors. To this end, the involved displacements together with a set of system parameters can be modeled as two continuous constrained optimization problems, whose global maximum characterize the maximum pose error. It is therefore mandatory to obtain a rigorous upper bound of this global maximum, which disqualifies the usage of local optimizers. This model can be seen as a sequence of variables

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 82–91, 2010.

lying in a continuous domain, a set of constraints, and an objective function. It turns out that such a model is computationally hard to solve: Computing time may increase exponentially with respect to the problem size, which in this case depends on the number of manipulator joints.

There exists some research works dealing with joint clearances modeling and error prediction due to joint clearances [7,9]. However, few of them deal with the solving techniques [14], and no work exists related to the use of constraint programming, a powerful modeling and solving approach. In this paper, we investigate the use of constraint programming for solving such problems. In particular, we combine the classic branch-and-bound algorithm with interval analysis and powerful pruning techniques. The branch-and-bound algorithm allows us to handle the optimization part of the problem, while computing time can be reduced thanks to the pruning techniques. Moreover, the reliability of the process is guaranteed by the use of interval analysis, which is mandatory for the application presented in this paper. The experiments demonstrate the efficiency of the proposed approach w.r.t. state-of-the-art solvers GAMS/BARON [1] and ECL^{i}PS^e [18].

The remaining of this paper is organized as follows. Section 2 presents an example of robot manipulator with the associated model for predicting the impact of the joints clearances on the end-effector pose. A presentation of constraint programming including the implemented approach is given in Sect. 3. The experiments are presented in Sect. 4, followed by the conclusion and future work.

2 Optimization Problem Formulation

In the scope of this paper, we consider serial manipulators composed of n revolute joints, n links and an end-effector. Figure 1 illustrates such a manipulator composed of two joints, named RR-manipulator, as well as a clearance-affected revolute joint. Let us assume that joint clearances appear due to manufacturing errors. Accordingly, the optimization problem aims to find the maximal positional and rotational errors of the manipulator end-effector for a given manipulator configuration.

2.1 End-Effector Pose without Joint Clearance

In order to describe uniquely the manipulator architecture, i.e. the relative location and orientation of its neighboring joints axes, the Denavit-Hartenberg nomenclature is used [6]. To this end, links are numbered 0, 1, ..., n, the jth joint being defined as that coupling the $(j-1)$st link with the jth link. Hence, the manipulator is assumed to be composed of $n+1$ links and n joints; where 0 is the fixed base, while link n is the end-effector. Next, a coordinate frame $\mathcal{F}_j$ is defined with origin O_j and axes X_j, Y_j, Z_j. This frame is attached to the $(j-1)$st link for $j = 1, \ldots, n+1$. The following screw takes $\mathcal{F}_j$ onto $\mathcal{F}_{j+1}$:

$$\mathbf{S}_j = \begin{bmatrix} \mathbf{R}_j & \mathbf{t}_j \\ \mathbf{0}_3^T & 1 \end{bmatrix}, \tag{1}$$

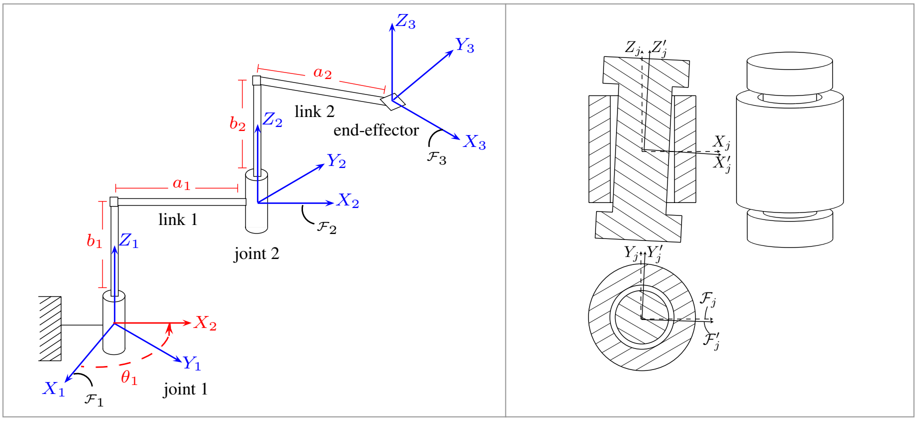


Fig. 1. Left: A serial manipulator composed of two revolute joints. Right: Clearance-affected revolute joint.

where $\mathbf{R}_j$ is a 3×3 rotation matrix; $\mathbf{t}_j \in \mathbb{R}^3$ points from the origin of $\mathcal{F}_j$ to that of $\mathcal{F}_{j+1}$; and $\mathbf{0}_3$ is the three-dimensional zero vector. Moreover, $\mathbf{S}_j$ may be expressed as:

$$\mathbf{S}_j = \begin{bmatrix} \cos\theta_j & -\sin\theta_j \cos\alpha_j & \sin\theta_j \sin\alpha_j & a_j \cos\theta_j \\ \sin\theta_j & \cos\theta_j \cos\alpha_j & -\cos\theta_j \sin\alpha_j & a_j \sin\theta_j \\ 0 & \sin\alpha_j & \cos\alpha_j & b_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2)$$

where α_j, a_j, b_j and θ_j represent respectively the link twist, the link length, the link offset, and the joint angle. Let us notice that a_1, b_1, a_2 and b_2 are depicted in Fig. 1 for the corresponding manipulator whereas α_1 and α_2 are null as the revolute joints axes are parallel.

Provided that the joints are perfectly rigid in all directions but one, that there is no joint clearance, that the links are perfectly rigid and that the geometry of the robotic manipulator is known exactly, the pose of the end-effector with respect to the fixed frame $\mathcal{F}_1$ is expressed as:

$$\mathbf{P} = \prod_{j=1}^{n} \mathbf{S}_j, \qquad (3)$$

However, if we consider joint clearances, we must include small errors in Eq. (3).

2.2 Joint-Clearance Errors

Taking into account joint clearances, the frame $\mathcal{F}_j$ associated with link $j-1$ is shifted to $\mathcal{F}'_j$. Provided it is small, this error on the pose of joint j with respect to joint $j-1$ may be represented by the small-displacement screw depicted in Eq. (4):

$$\delta \mathbf{s}_j \equiv \begin{bmatrix} \delta \mathbf{r}_j \\ \delta \mathbf{t}_j \end{bmatrix} \in \mathbb{R}^6, \quad (4) \qquad \delta \mathbf{S}_j = \begin{bmatrix} \delta \mathbf{R}_j & \delta \mathbf{t}_j \\ \mathbf{0}_3^T & 0 \end{bmatrix}, \quad (5)$$

where $\delta \mathbf{r}_j \in \mathbb{R}^3$ represents the small rotation taking frame $\mathcal{F}_j$ onto $\mathcal{F}'_j$, while $\delta \mathbf{t}_j \in \mathbb{R}^3$ points from the origin of $\mathcal{F}_j$ to that of $\mathcal{F}'_j$. It will be useful to represent $\delta \mathbf{s}_j$ as the 4×4 matrix given with Eq. (5) where $\delta \mathbf{R}_j \equiv \partial(\delta \mathbf{r}_j \times \mathbf{x})/\partial \mathbf{x}$ is the cross-product matrix of $\delta \mathbf{r}_j$. Intuitively, clearances in a joint are best modeled by bounding its associated errors below and above. Assuming that the lower and upper bounds are the same, this generally yields six parameters that bound the error screw $\delta \mathbf{s}_j$. Accordingly, the error bounds are written as:

$$\delta r_{j,X}^2 + \delta r_{j,Y}^2 \leq \delta \beta_{j,XY}^2, \quad (6) \qquad \delta t_{j,X}^2 + \delta t_{j,Y}^2 \leq \delta b_{j,XY}^2, \quad (8)$$

$$\delta r_{j,Z}^2 \leq \delta \beta_{j,Z}^2, \quad (7) \qquad \delta t_{j,Z}^2 \leq \delta b_{j,Z}^2, \quad (9)$$

where $\delta \mathbf{r}_j \equiv [\delta r_{j,X}\ \delta r_{j,Y}\ \delta r_{j,Z}]^T$ and $\delta \mathbf{t}_j \equiv [\delta t_{j,X}\ \delta t_{j,Y}\ \delta t_{j,Z}]^T$.

2.3 End-Effector Pose with Joint Clearances

Because of joints clearances, the end-effector frame $\mathcal{F}_{n+1}$ is shifted to $\mathcal{F}'_{n+1}$. From [16], the displacement taking frame $\mathcal{F}_j$ onto $\mathcal{F}'_j$ is given by the matrix exponential of $\delta \mathbf{S}_j$, $e^{\delta \mathbf{S}_j}$. As a result, the screw that represents the pose of the shifted end-effector may be computed through the kinematic chain as:

$$\mathbf{P}' = \prod_{j=1}^{n} e^{\delta \mathbf{S}_j} \mathbf{S}_j, \qquad (10)$$

where screw $\mathbf{P}'$ takes frame $\mathcal{F}_1$ onto $\mathcal{F}'_{n+1}$ when taking errors into account.

2.4 The End-Effector Pose Error Modeling

In order to measure the error on the pose of the moving platform, we compute the screw $\Delta \mathbf{P}$ that takes its nominal pose $\mathcal{F}_{n+1}$ onto its shifted pose $\mathcal{F}'_{n+1}$ through the kinematic chain, namely:

$$\Delta \mathbf{P} = \mathbf{P}^{-1} \mathbf{P}' = \prod_{j=n}^{1} \mathbf{S}_j^{-1} \prod_{j=1}^{n} \left(e^{\delta \mathbf{S}_j} \mathbf{S}_j \right). \qquad (11)$$

From [4], it turns out that $\Delta \mathbf{P}$ may as well be represented as a small-displacement screw $\delta \mathbf{p}$ in a vector form, namely,

$$\delta \mathbf{p} = \sum_{j=1}^{n} \left(\prod_{k=n}^{j} \operatorname{adj}(\mathbf{S}_k)^{-1} \right) \delta \mathbf{s}_j, \quad \text{with} \quad \operatorname{adj}(\mathbf{S}_j) \equiv \begin{bmatrix} \mathbf{R}_j & \mathbf{O}_{3\times 3} \\ \mathbf{T}_j \mathbf{R}_j & \mathbf{R}_j \end{bmatrix}$$

being the adjoint map of screw $\mathbf{S}_j$ and $\mathbf{T}_j \equiv \partial(\mathbf{t}_j \times \mathbf{x})/\partial \mathbf{x}$ the cross-product matrix of $\mathbf{t}_j$.

2.5 The Maximum End-Effector Pose Error

Let $\delta\mathbf{p}$ be expressed as $[\delta\mathbf{p}_r\ \delta\mathbf{p}_t]^T$ with $\delta\mathbf{p}_r \equiv [\delta p_{r,X}\ \delta p_{r,Y}\ \delta p_{r,Z}]^T$ and $\delta\mathbf{p}_t \equiv [\delta p_{t,X}\ \delta p_{t,Y}\ \delta p_{t,Z}]^T$ characterizing the rotational and translational errors of the manipulator end-effector, respectively. In order to find the maximal pose errors of the end-effector for a given manipulator configuration, we need to solve two optimization problems:

$$\begin{aligned} \mathbf{max} \quad & \|\delta\mathbf{p}_r\|_2, \\ \mathbf{s.t.} \quad & \delta r_{j,X}^2 + \delta r_{j,Y}^2 - \delta\beta_{j,XY}^2 \leq 0, \\ & \delta r_{j,Z}^2 - \delta\beta_{j,Z}^2 \leq 0, \\ & j = 1, \ldots, n \end{aligned} \tag{12}$$

$$\begin{aligned} \mathbf{max} \quad & \|\delta\mathbf{p}_t\|_2, \\ \mathbf{s.t.} \quad & \delta r_{j,X}^2 + \delta r_{j,Y}^2 - \delta\beta_{j,XY}^2 \leq 0, \\ & \delta r_{j,Z}^2 - \delta\beta_{j,Z}^2 \leq 0, \\ & \delta t_{j,X}^2 + \delta t_{j,Y}^2 - \delta b_{j,XY}^2 \leq 0, \\ & \delta t_{j,Z}^2 - \delta b_{j,Z}^2 \leq 0, \\ & j = 1, \ldots, n \end{aligned} \tag{13}$$

where $\|.\|_2$ denotes the 2-norm. The maximum rotational error due to joint clearances is obtained by solving problem (12). Likewise, the maximum point-displacement due to joint clearances is obtained by solving problem (13). It is noteworthy that the constraints of the foregoing problems are defined with Eqs. (6)–(9). Moreover, it appears that those problems are nonconvex quadratically constrained quadratic (QCQPs). Although their feasible sets are convex—all the constraints of both problems are convex—their objectives are convex, making the computation of their global maximum NP-Hard.

3 Constraint Programming

3.1 Definitions

Constraint Programming (CP) is a programming paradigm that allows one to solve problems by formulating them as a Constraint Satisfaction Problem (CSP). We are mainly interested in Numerical CSP (NCSP) whose variables belong to continuous domains. This formulation consists of a sequence of variables lying in a domain and a set of constraints that restrict the values that the variables can take. The goal is to find a variable-value assignment that satisfies the whole set of constraints. Formally, a NCSP $\mathcal{P}$ is defined by a triple $\mathcal{P} = \langle \mathcal{X}, [\mathbf{x}], \mathcal{C} \rangle$ where:

- $\mathcal{X}$ is a vector of variables $(x_1, x_2, \ldots, x_n)$.
- $[\mathbf{x}]$ is a vector of real intervals $([x_1], [x_2], \ldots, [x_n])$ such that $[x_i]$ is the domain of x_i.
- $\mathcal{C}$ is a set of constraints $\{c_1, c_2, \ldots, c_m\}$. In the scope of this paper, we focus on inequality constraints, i.e., $c(\mathbf{x}) \iff g(\mathbf{x}) \leq 0$ where $g : \mathbb{R}^n \to \mathbb{R}$ is assumed to be a differentiable function.

A solution of a CSP is a real vector $\mathbf{x} \in [\mathbf{x}]$ that satisfies each constraint, i.e. $\forall c \in \mathcal{C}, c(\mathbf{x})$. The set of solutions of the CSP $\mathcal{P}$ is denoted by $\mathrm{sol}(\mathcal{P})$. This definition can be extended to support optimization problems by also considering a cost function f. Then we want to find the element of $\mathrm{sol}(\mathcal{P})$ that minimizes or maximizes the cost function.

3.2 Interval Analysis

The modern interval analysis was born in the 60's with [15] (see [17,10] and references therein). Since, it has been widely developed and is today one central tool in the rigorous resolution of NCSPs (see [3] and extensive references).

Intervals, interval vectors and interval matrices are denoted using brackets. Their sets are denoted respectively by $\mathbb{IR}$, $\mathbb{IR}^n$ and $\mathbb{IR}^{n\times m}$. The elementary functions are extended to intervals in the following way: Let $\circ \in \{+,-,\times,/\}$ then $[x]\circ[y] = \{x \circ y : x \in [x], y \in [y]\}$ (division is defined only for denominators that do not contain zero). E.g. $[a,b]+[c,d] = [a+c, b+d]$. Also, continuous functions $f(x)$ with one variable are extended to intervals using the same definition: $f([x]) = \{f(x) : x \in [x]\}$, which turns to be an interval as f is continuous. When numbers are represented with a finite precision, the previous operations cannot be computed in general. The outer rounding is then used so as to keep valid the interpretations. For example, $[1,2]+[2,3]$ would be equal to $[2.999, 5.001]$ if rounded with a three decimal accuracy.

Then, an expression which contains intervals can be evaluated using this interval arithmetic. The main property of interval analysis is that such an interval evaluation gives rise to a superset of the image through the function of the interval arguments. For example, the interval evaluation of expression $x(y-x)$ is $[x]\times([y]-[x])$ and contains $\{x(y-x) : x \in [x], y \in [y]\}$. In some cases (e.g. when the expression contains only one occurrence of each variable), this enclosure is optimal.

Given an n-ary constraint c and a box $[\mathbf{x}] \in \mathbb{R}^n$, a contractor for c will contract the box $[\mathbf{x}]$ without losing any solution of c. Some widely used contractors are based on the 2B-consistency (also called hull-consistency) or the box consistency [12,2], which are pruning methods similar to the well known arc-consistency [13] in the context of discrete CSPs. They are both applied to one constraint at a time, hence suffering of the usual drawbacks of the locality of their application. We encapsulate this notion in the function $\mathrm{Contract}_{\mathcal{C}}([\mathbf{x}])$ which uses the constraints $\mathcal{C}$ to prune the box $[\mathbf{x}]$. Thus, the result is a new box $[\mathbf{x}'] \subseteq [\mathbf{x}]$ that satisfies $\mathbf{x} \in [\mathbf{x}] \wedge (\forall c \in \mathcal{C}, c(\mathbf{x})) \Rightarrow \mathbf{x} \in [\mathbf{x}']$, i.e. no solution was lost during the pruning. This property, rigorously achieved thanks to the correct rounding of interval arithmetic, allows the CP framework to provide rigorous proofs of mathematical statements.

3.3 The Branch and Bound Algorithm

NCSPs are usually solved using a branch and prune algorithm. A basic branch and prune algorithm is described by Algorithm 1. Its input is a set of constraints and an initial box domain. It interleaves pruning (Line 4) and branching (Line 5) to output a set of boxes that sharply covers the solution set: Due to the property satisfied by the function $\text{Contract}_{\mathcal{C}}([\mathbf{x}])$, Algorithm 1 obviously maintain the property $\mathbf{x} \in [\mathbf{x}] \wedge (\forall c \in \mathcal{C}, c(\mathbf{x})) \Rightarrow \mathbf{x} \in \cup\mathcal{L}$. The stopping criterion is usually the size of the boxes in $\mathcal{L}$, the algorithm stopping when every box got smaller than a fixed precision. We have used the branch and prune algorithm implemented in RealPaver [8] which basically proceeds as Algorithm 1 does.

A branch and prune algorithm can be modified to a branch and bound algorithm that handles minimization problems (maximization problems are handled similarly). Such a simple branch and bound algorithm is described by Algorithm 2. The cost function is an additional input, and maintains an upper bound on the global minimum in the variable m (which is initialized to $+\infty$ at the beginning of the search at Line 2). This upper bound is used to discard parts of the search space whose cost is larger than it by adding the constraint $f(\mathbf{x}) \leq m$ to the set of constraints (Line 5). Finally, the upper bound is updated at Line 6 by searching for feasible points inside the current box $[\mathbf{x}]$ and when such points are found, by evaluating the cost function at these points. In our current implementation, some random points are generated inside $[\mathbf{x}]$ and the constraints are rigorously checked using interval arithmetic before updating the upper bound[1]. The branch and bound algorithm maintains an enclosure of the global minimum, that converges to the global minimum provided that feasible points are found during the search (which is guaranteed in the case of inequality constraints that are not singular).

Algorithm 1

```
Input: $\mathcal{C} = \{c_1, \ldots, c_m\}$, $[\mathbf{x}]$
1 $\mathcal{L} \leftarrow \{[\mathbf{x}]\}$
2 While $\mathcal{L} \neq \emptyset$ and $\neg$stop_criteria do
3     $([\mathbf{x}], \mathcal{L}) \leftarrow$Extract$(\mathcal{L})$
4     $[\mathbf{x}] \leftarrow$Contract$_{\mathcal{C}}([\mathbf{x}])$
5     $\{[\mathbf{x}'], [\mathbf{x}'']\} \leftarrow$Split$([\mathbf{x}])$
6     $\mathcal{L} = \mathcal{L} \cup \{[\mathbf{x}'], [\mathbf{x}'']\}$
7 End While
8 Return$(\mathcal{L})$
```

Algorithm 2

```
Input: $f$, $\mathcal{C} = \{c_1, \ldots, c_m\}$, $[\mathbf{x}]$
1  $\mathcal{L} \leftarrow \{[\mathbf{x}]\}$
2  $m \leftarrow +\infty$
3  While $\mathcal{L} \neq \emptyset$ and $\neg$stop_criteria do
4      $([\mathbf{x}], \mathcal{L}) \leftarrow$Extract$(\mathcal{L})$
5      $[\mathbf{x}] \leftarrow$Contract$_{\mathcal{C} \cup \{f(\mathbf{x}) \leq m\}}([\mathbf{x}])$
6      $m \leftarrow$Update$([\mathbf{x}], f)$
7      $\{[\mathbf{x}'], [\mathbf{x}'']\} \leftarrow$Split$([\mathbf{x}])$
8      $\mathcal{L} = \mathcal{L} \cup \{[\mathbf{x}'], [\mathbf{x}'']\}$
9  End While
10 Return$(\mathcal{L}, m)$
```

Fig. 2. Left: The branch and prune algorithm. Right: The branch and bound algorithm.

An efficient implementation of the branch and bound algorithm requires the list of boxes $\mathcal{L}$ to be carefully handled: It has to be maintained sorted w.r.t. a lower bound of the objective evaluated for each box. So, each time a box $[\mathbf{x}]$ is inserted into $\mathcal{L}$, the interval evaluation of the objective $f([\mathbf{x}])$ is computed and the lower bound of this interval evaluation is used to maintain the list sorted.

[1] More elaborated branch and prune algorithms usually use a local search to find good feasible points, but preliminary experiments presented in Section 4 and performed using this simple random generation of potential feasible points already showed good performances.

Then, Line 4 extracts the first box of $\mathcal{L}$ so that most promising regions of the search space are explored first, leading to drastic improvements. Another advantage of maintaining $\mathcal{L}$ sorted is that its first box contains the lowest objective value over all boxes. Therefore, we use this value and check its distance to the current upper bound m and stop the algorithm when the absolute precision of the global minimum has reached a prescribed value.

The branch and bound algorithm described above has been implemented on top of RealPaver [8]. This implementation is used for the experiments presented in the next section.

4 Experiments

We have performed a set of experiments to analyze the performance of our approach solving the pose error problem. We compare it with GAMS/BARON [1] and ECL^{i}PS^e [18]. GAMS/BARON is a widely used system for mathematical programming and optimization[2] and ECL^{i}PS^e is one of the few CP systems supporting continuous optimization problems.

We have tested 8 models, 4 translations (T) and 4 rotation (R) models. For both types of model we consider from 2 to 5 joints. Table 1 depicts the results obtained. Columns 1 and 2 show the number of joints and the type of problem. Columns 3 to 5 depict relevant data about the problem size (number of variables, number of constraints, and number of arithmetic operations in the objective function). Columns 5 to 10 depict the solving times using GAMS, RealPaver (including 2 different filtering techniques), and ECL^{i}PS^e. Experiments were run on a 3 Ghz Pentium D with 2 GB of RAM running Ubuntu 9.04. All solving times are the best of five runs.

The results show that our approach is faster in almost all cases. In smaller problems such as 2R, 2T, 3R and 3T, RealPaver exhibits great performance. It can even be 100 times faster than GAMS. Such a faster convergence can be explained on one hand by the efficient work done by the filtering techniques HC4 and BC4 (based on hull and box consistencies respectively) and on the other hand by the fact that there is no need for an accurate computation for this particular problem. In fact, we do not have to go beyond a precision of 10^{-2}, making the interval computations less costly than usual. Moreover, the impact on the solving time of the used consistency is not surprising: using HC4 is faster than BC4, as expected when dealing with constraints having variables not occurring multiple times [5].

In bigger problems such as 4R and 5R, RealPaver remains faster. However, when the problem size arises, in particular when the number of variables exceeds 20, the convergence starts to be slow. This is a common phenomenon we can explain thanks both to the exponential in problem size complexity of this kind of branch-and-bound solving algorithms and to the fact that the objective function itself grows exponentially with the number of variables it involves. Nevertheless, we believe that solving times are reasonable with regard to the complexity and

[2] In our version, LP and NLP solving are respectively done by CPLEX and MINOS.

Table 1. Solving times (seconds)

#joints	Type	Problem Size			GAMS	RealPaver		ECL^{i}PS^e
		#var	#ctr	#op		HC4	BC4	
2	T	12	8	28	0.08	0.004	0.004	>60
2	R	6	4	18	0.124	0.004	0.004	>60
3	T	18	12	135	0.124	0.008	0.016	t.o.
3	R	9	6	90	0.952	0.004	0.008	t.o.
4	T	24	16	374	0.144	0.152	0.168	t.o.
4	R	12	8	205	2.584	0.02	0.02	t.o.
5	T	30	20	1073	0.708	>60	>60	t.o.
5	R	15	10	480	9.241	0.26	0.436	t.o.

size of the problems as well as the techniques used. But solving times may actually not be the point to emphasize here: on the contrary, it may rather be important to discuss the reliability of the solutions given by GAMS/BARON. It was already noted in [11] that BARON is not reliable in general. We have also verified using RealPaver that the optimum enclosure computed by BARON is unfeasible.

Finally, it is worth noticing that although RealPaver as well as ECL^{i}PS^e use CP techniques, ECL^{i}PS^e is completely outperformed even for the smallest of our instances. It is not surprising, given that RealPaver was designed to solve problems of this kind whereas ECL^{i}PS^e is a more generic and extensible tool.

5 Conclusion and Future Work

In this paper we have shown how to efficiently solve the pose error problem by using constraint programming. We have modeled the problem as a NCSP, and then shown how to turn a CP classical branch-and-prune resolution scheme into a branch-and-bound algorithm, dedicated to the solving of continuous constrained optimization problems. Finally, we presented experimental results for several instances of the problem, and showed that our approach was very competitive with respect to some state-of-the-art CP and optimization tools.

It is important to notice that this approach is not specific to robotics, it is indeed applicable to any problem where rigorous computation and numerical reliability are required. Moreover, in the future we should be able to design new pruning criteria in order to accelerate the convergence of the optimization process. This way, we should tackle the scalability issue and thus be able to solve faster the large instances of the problem.

References

1. GAMS (Visited October 2009), http://www.gams.com/
2. Benhamou, F., Mc Allester, D., Van Hentenryck, P.: CLP (Intervals) Revisited. In: Proceedings of ILPS, pp. 124–138. MIT Press, Cambridge (1994)

3. Benhamou, F., Older, W.J.: Applying Interval Arithmetic to Real, Integer and Boolean Constraints. Journal of Logic Programming 32(1), 1–24 (1997)
4. Cardou, P., Caro, S.: The Kinematic Sensitivity of Robotic Manipulators to Manufacturing Errors. Technical report, IRCCyN, Internal Report No RI2009_4 (2009)
5. Collavizza, H., Delobel, F., Rueher, M.: Comparing Partial Consistencies. Reliable Computing 5(3), 213–228 (1999)
6. Denavit, J., Hartenberg, R.S.: A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices. ASME J. Appl. Mech. 23, 215–221 (1955)
7. Fogarasy, A., Smith, M.: The Influence of Manufacturing Tolerances on the Kinematic Performance of Mechanisms. Proc. Inst. Mech. Eng., Part C: J. Mech. Eng. Sci., 35–47 (1998)
8. Granvilliers, L., Benhamou, F.: Algorithm 852: RealPaver: an Interval Solver Using Constraint Satisfaction Techniques. ACM Trans. Math. Softw. 32(1), 138–156 (2006)
9. Innocenti, C.: Kinematic Clearance Sensitivity Analysis of Spatial Structures With Revolute. ASME J. Mech. Des. 124, 52–57 (2002)
10. Jaulin, L., Kieffer, M., Didrit, O., Walter, E.: Applied Interval Analysis with Examples in Parameter and State Estimation, Robust Control and Robotics. Springer, Heidelberg (2001)
11. Lebbah, Y., Michel, C., Rueher, M.: An Efficient and Safe Framework for Solving Optimization Problems. Journal of Computational and Applied Mathematics 199(2), 372–377 (2007); Special Issue on Scientific Computing, Computer Arithmetic, and Validated Numerics (SCAN 2004)
12. Lhomme, O.: Consistency Techniques for Numeric CSPs. In: Proceedings of IJCAI, pp. 232–238 (1993)
13. Mackworth, A.: Consistency in Networks of Relations. Artificial Intelligence 8(1), 99–118 (1977)
14. Meng, J., Zhang, D., Li, Z.: Accuracy Analysis of Parallel Manipulators With Joint Clearance. ASME J. Mech. Des. 131 (2009)
15. Moore, R.: Interval Analysis. Prentice-Hall, Englewood Cliffs (1966)
16. Murray, R., Li, Z., Sastry, S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton (1994)
17. Neumaier, A.: Interval Methods for Systems of Equations. Cambridge Univ. Press, Cambridge (1990)
18. Wallace, M., Novello, S., Schimpf, J.: ECLiPSe: A Platform for Constraint Logic Programming. Technical report, IC-Parc, Imperial College, London (1997)

Down-Up-Down Behavior Generation for Interactive Robots

Yasser Mohammad[1] and Toyoaki Nishida[2]

[1] Department of Electrical Engineering, Faculty of Engineering,
Assiut University, Egypt
yasserm@aun.edu.eg

[2] Department of Intelligence Science and Technology,
Graduate School of Informatics, Kyoto University, Japan
nishida@i.kyoto-u.ac.jp

Abstract. Behavior generation in humans and animals usually employs a combination of bottom-up and top-down patterns. Most available robotic architectures utilize either bottom-up or top-down activation including hybrid architectures. In this paper, we propose a behavior generation mechanism that can seamlessly combine these two strategies. One of the main advantages of the proposed approach is that it can naturally combine both bottom-up and top-down behavior generation mechanisms which can produce more natural behavior. This is achieved by utilizing results from the theory of simulation in neuroscience which tries to model the mechanism used in human infants to develop a theory of mind. The proposed approach was tested in modeling spontaneous gaze control during natural face to face interactions and provided more natural, human-like behavior compared with a state-of-the-art gaze controller that utilized a bottom-up approach.

1 Introduction

Hierarchies of behaviors are ubiquitous in modern robotic architectures. Both reactive [1], deliberative and hybrid [2] architectures tend to utilize hierarchies of behaviors to generate complex external behavior.

In reactive architectures like the subsumption architecture [1], activation of the control processes (behaviors) in some level of the hierarchy will most likely be triggered by some events in lower layers of the hierarchy leading to a bottom-up behavior generation mechanism.

In the other end of the spectrum, architectures that utilize symbolic reasoning and representation like the BDI (Belief-Desire-Intention) architecture tend to utilize the hierarchical approach in order to divide the behavior generation problem into smaller easier problems. In this case the activation of processes in lower levels is usually triggered by processes in higher levels (e.g. goals trigger subgoals etc). This leads to a top-down behavior generation mechanism.

More recent robotic architectures are usually hybrid behavioral ones that utilize both reaction and deliberation. The direction of behavior generation in these architectures vary. For example, Arkin et al. [2] utilized an ethological approach

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 92–101, 2010.

in which a motivational system is used to guide the behavior of the robot. This architecture utilized a rich ethogram (categorization of behavior patterns) that was based on the research done by researchers in animal behavior. The low level behaviors were implemented using reactive processes and deliberation was used to select among these reactive processes based on the activation levels of *higher-level* motivational subgoals. This led to a top-down behavior generation mechanism similar to the ones found in purely deliberative systems. Most hybird reactive-deliberative architecture use a similar technique by having the deliberative layer dictates which reactive processes to run [3]. An exception of this is the system proposed in [4] in which the reactive layer works as an advisor for the deliberation layer making behavior generation go from bottom-up.

Humans and animals, on the other hand, utilize both top-down goal directed and bottom-up situation aware approaches for behavior generation. For example, Wohlschlager, Gattis, and Bekkering have shown that even during imitation, a goal directed approach is more appropriate than a traditional direct coupling approach [5]. Their model called GOADI (GOAl Directed Imitation) was shown to correspond more accurately to observed behavior of infants and animals during learning compared with traditional sensorimotor direct coupling approaches like Meltzoff's AIM (Active Intermodal Mapping) theory [6]. For this goal directed coupling to happen, the agent must be able to utilize both action to goal inference (bottom-up activation) as well as goal to action generation (top-down activation) and to combine effectively these two directions for the final behavior generation. This is in contrast to most robotic architectures that utilize only one of these two behavior generation approaches as previous paragraphs have shown.

Our goal is to design a behavior generation mechanism that can have similar fluency in achieving natural behavior. Moreover, we are interested in interactive social situations in which the robot is interacting with untrained humans using natural communication mechanisms that we call the *interaction protocol* in this paper. This is a challenging problem for many reasons. Firstly, each computational process may have conflicting activation levels in the top-down and bottom-up paths and a principled way is needed to select the final activation level based on these possibly conflicting values. Secondly, The model of the partner used to understand his/her behavior should also incorporate these two directions of activation which can greatly complicate the design of the robot. Thirdly, the computational processes (behaviors) at various levels should be learnable while taking into account the dependency of each of them on the processes both under and over it in the behavioral hierarchy.

This paper presents and evaluates a behavior generation mechanism called the DUD (Down-Up-Down) as applied to the L_iEICA (Embodied Interactive Control Architecture) proposed by the authors [7]. This behavior generation mechanism can meet the three challenges mentioned in the previous paragraph utilizing the simulation theory of mind.

The rest of this paper is organized as follows: Section 2 briefly describes the theory of simulation upon which the proposed approach is based. Section 3 describes the main components of the architecture for which the DUD behavior generation mechanism was designed. Section 4 describes the proposed behavior generation mechanism. Section 5 reports an experiment to compare the performance of the proposed behavior generation mechanism and a state-of-the-art bottom-up approach in the context of gaze control during face to fact interactions. The paper is then concluded.

2 Simulation Theory

To understand the intentions of other people, humans develop a theory of mind that tries to understand the actions of interacting partners in a goal directed manner. Failure to develop this theory of mind is hypothesized to be a major factor in developing autism and other interaction disorders. Two major theories are competing to explain how humans learn and encode the theory of mind namely the theory of theory and the theory of simulations [8]. The theory of theory hypothesizes that a separate recognition mechanism is available that can decode the partner's behavior while the theory of simulation suggests that the same neuronal circuitry is used for both generation of actions and recognition of those actions when performed by others [9,10]. The discovery of mirror neurons in the F5 premotor cortex area of monkeys and recent evidence of there existence in humans [11] support the theory of simulation although the possibility of the existence of other separate recognition mechanism is far from being ruled out.

If the assumption of the theory of simulation are right then, taking it to its limits, there should be no difference in the subconscious level between behaving in one role while understanding the behavior of other partners (in all other roles) and behaving in any other role of the interaction. For example, in the explanation scenario in which an instructor is teaching a listener how to achieve some task, there should be no internal difference in the lower levels of the architecture between acting as a listener and as an instructor. The only difference should be in how the final behavior commands are transferred to the actuators and how the sensor data are interpreted.

The proposed architecture follows this assumption and arranges its computational components so that internally there is no difference between acting in any role of the interaction (e.g. listener, instructor, bypasser, etc). This property combined with simultaneous learning of all interaction roles are unique features of this architecture that cannot be found in available HRI architectures [7]. As will be shown later, these features enables the DUD mechanism to naturally combine both top-down and bottom-up behavior generation approaches.

3 Architecture components

The proposed system provides a simulation theoretic mechanism for recognizing interaction acts of the partner in a goal directed manner. This mechanism is augmented by a separate recognition mechanism to enable learning the interaction protocol using the interaction itself as will be explained later in this section.

The most important parts of the architecture for the purposes of this paper are:

Forward Basic Interaction Acts (FBIAs). The basic interactive acts that the agent is capable of. In this work we use Recurrent Neural Networks for these processes.

Reverse Basic Interaction Acts (RBIAs). Every FBIA has a reverse version that detects the probability of its execution in the signals perceived by the robot.

Interactive Control Processes (ICPs). These processes constitute the higher interactive control layers. Every interactive control process consists of two twin processes in the same way as FBIAs and RBIAs.

The following section explains the mechanism used to calculate the activation levels of each of these processes as well as its inputs and outputs.

4 Down-Up-Down Behavior Generation (DUD)

This section presents the behavior generation mechanism used to generate the final behavior. Algorithm 1 provides an overview of the proposed behavior generation mechanism. It should be clear that DUD is not a *serial algorithm* in the traditional sense but it describes how each process *asynchronously* and *independently* calculates its outputs and how each one of them is activated.

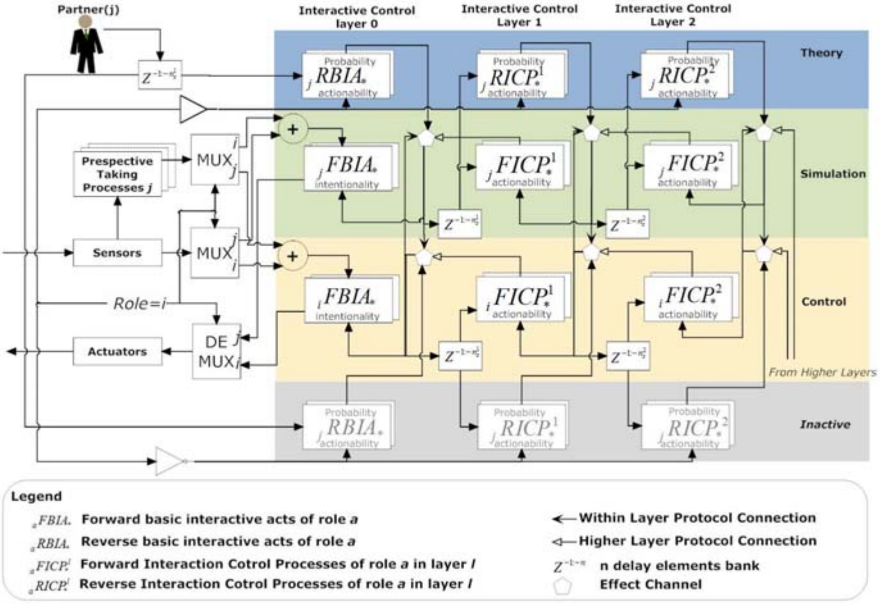

Partner(j)
Interactive Control layer 0
Interactive Control Layer 1
Interactive Control Layer 2
Theory
Simulation
Control
Inactive
From Higher Layers
Probability
actionability
intentionality
RBIA
RICP
FBIA
FICP
Prespective Taking Processes j
Sensors
MUX
DE MUX
Role=i
Actuators
Legend
Forward basic interactive acts of role a
Reverse basic interactive acts of role a
Forward Interaction Cotrol Processes of role a in layer l
Reverse Interaction Cotrol Processes of role a in layer l
Within Layer Protocol Connection
Higher Layer Protocol Connection
n delay elements bank
Effect Channel

Fig. 1. The three information flow paths during DUD behavior generation

Fig. 1 shows the three information paths during behavior generation called the *theory*, *simulation* and *control* paths.

The *theory* path represents current understanding of partner's behavior using current protocol and is composed from reverse interactive control processes (and reverse basic interactive acts). All processes in this path are activated during the whole interaction. The information passes bottom-up in this path. Notice that only the reverse processes corresponding to partner roles are activated while the reverse processes corresponding to agent's role are disabled using the inverter controlled by the role variable shown in Fig. 1. Activation is going bottom-up in this path.

The *simulation* path represents a simulation of partner's behavior assuming its current perceived situation as encoded through perspective taking processes. This path is represented by the forward interactive acts of all roles except the role currently taken by the robot/agent. Corresponding forward interactive acts are shielded from the actual actuators of the robot through the DEMUX (Demultiplexer) shown in Fig. 1 which is controlled using the role variable. Activation is going top-down in this path.

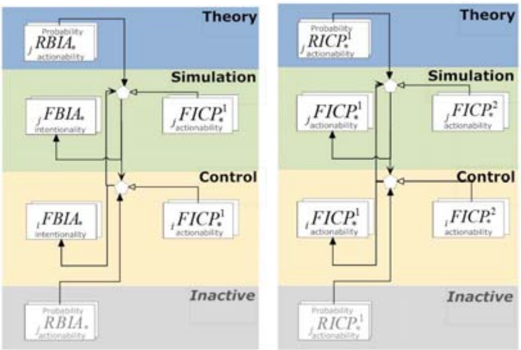
The *control* path represents the current controller that is allowed to access the robot sensors and actuators directly through the lower MUX and DEMUX in Fig. 1. These MUX and DEMUX are controlled using the role variable. Activation in this layer is also going from top downward.

The three paths are interacting to give the final behavior of the robot.

Algorithm 1. DUD Behavior Generation Mechanism

1. Set *role* variable to the current role in the interaction.
2. Use perspective taking processes to generate input to all RBIAs and propagate the calculation of all RICPs of all layers.
3. Asynchronously calculate intentionality of all FBIAs using combination of within-layer protocol factor, inter-layer protocol factor, and theory factor.
4. Asynchronously calculate actionability of all FICPs using combination of within-layer protocol factor, inter-layer protocol factor, and theory factor.
5. Asynchronously update inputs and calculate outputs of FBIAs and propagate the calculations to all FICPs.
6. Repeat this steps until *role* changes or all activation levels become zero (end of interaction).

Fig. 2 shows the factors affecting the activation levels of FICPs and FBIAs. As clearly visible by comparing Fig. 2-a and Fig. 2-b, the only difference between FICPs and FBIAs in this regard is that for FBIAs the activation level is translated to intentionality value in L_0EICA terminology while for FICPs it is translated to actionability. The actionability of FBIAs is connected always to 1 as long as this interaction protocol is activated by higher level deliberation layers which are not a part of L_iEICA. For simplicity we assume that the robot has a single interaction protocol implemented. For all other purposes FBIA and FICP activation is the same so we will speak about FICPs only in the rest of this



(a) BIAs (b) ICPs

Fig. 2. The factors affecting the activation level of basic interactive acts and interaction control processes

section but it should be clear that concerning activation level, the same rules apply for FBIAs.

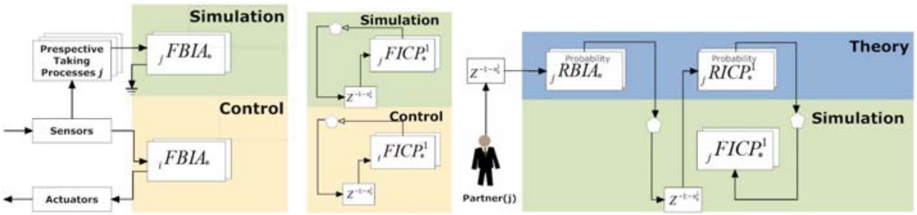
As Fig. 2-b shows, the activation level of each FICP is controlled through three factors:

1. The activation level of the corresponding RICP from the theory or inactive paths. This is called the theory factor. In case of the ICPs representing the current role, the theory factor is zero because the activation level of all inactive RICPs is set to zero (see Fig. 1).
2. The activation level of other FICPs corresponding to other roles in the interaction (from the control or simulation paths). This is called the within-layer protocol factor. For the ICPs corresponding to the current role this within-layer protocol factor is coming from ICPs in the simulation path while for ICPs representing other agents in the interaction, this within-layer protocol factor comes from ICPs in the control path. This within-layer protocol factor represents the protocol at one time scale.
3. The output of higher level ICPs corresponding to the self for the control path and to the same other agent in the simulation path. This is called the inter-layer protocol factor. This factor represents the top-down activation representing the relation between the interaction protocols at different time scales.

The architecture itself does not specify how to combine these factors. In most of our implementations though we simply average the three contributions using the activation levels of their sources as weights.

The reverse version of FBIAs and ICPs (RBIAs and RICPs) are activated based on the current role as shown in Fig. 1. All reverse processes corresponding to current role are deactivated (this is the *inactive* path) while all reverse processes corresponding to other roles are activated (this is the *theory* path).

After understanding how each ICP and BIA is activated we turn our attention to the meaning of the inputs and outputs of them. The input-output mapping



(a) FBIAs (b) FICPs (c) FBIAs & FICPs

Fig. 3. Input/Output mapping implemented by FBIAs, FICPs, and reverse processes

implemented by BIAs and ICPs is quite different and for this reason we will treat each of them separately. Moreover reverse and forward versions are also treated differently.

Fig. 3(a) shows the inputs and outputs of forward an example basic interactive act. The input for the FBIAs representing the current role (in the control path) are taken directly from the sensors while its output go to the actuators. The inputs for the FBIAs representing all other roles in the interaction (the simulation path) are taken from perspective taking processes and the output is simply discarded. In actual implementation the FBIAs corresponding to other agents are disabled by having their activation set to zero to save their computation time. Even though in this figure it seems that there is a difference in the wiring between FBIAs corresponding to current role and other FBIAs which does not agree with the cognitive indistinguishability rule described in section 2, this difference is just an effect of the simplification in drawing the figure. Using the set of MUXs and the DEMUX shown if Fig. 1 the wiring of all FBIAs is identical and the role variable controls at run time which FBIAs are given the sensor data directly and which are given the output of perspective taking processes. Also the same variable controls which FBIAs are connected to the actuators of the robot and which are not.

Fig. 3(b) shows the inputs and outputs of an example forward interactive control process. The figure shows explicitly that the FICPs in control and simulation paths have exactly the same inputs and outputs. This feature is in agreement with the cognitive indistinguishability suggested by the theory of simulation. The inputs of the FICP are the last n_z activation levels of the FICPs in the same path but the lower interaction control layer (the output of their effect channels connected to their activation level). n_z controls the relative speed of the FICPs running in this layer and the lower layer. In general each FICP can use a different n_z value. The output of each FICP is the inter-layer protocol factor affecting the FICPs of the lower layer in the same path. This means that FICPs of layer i implement a higher level slower protocol on top of the protocol implemented with the FICPs in layer $i-1$. In [12] we described how the number of layers, the number of FICPs in each layer and the strengths of the inter-layer protocol factors are learned.

Fig. 3(c) shows the inputs and outputs of both reverse basic interactive acts and reverse interaction control processes. The figure shows that in general the mappings implemented by these two types of processes are very similar. keeping in mind that all RBIAs and RICPs in the inactive path are disabled, we will describe only the operation of the RBIAs and RICPs in the theory path. It should be clear that the reverse processes in the inactive path have exactly the same wirings but with the control layer and they are not considered here because they are always disabled (when they are enabled the role is reversed and the control and simulation paths are also reversed and in this case the reverse processes currently in the inactive path become the theory path and vice versa). The inputs to RBIAs in the theory path are the last n_z values of all the sensors sensing the behavior of the partner. The output is the probability that the corresponding FBIAs are active in the partner control architecture assuming that the partner has the same control architecture (the central hypothesis of the theory of simulation). This output is the theory factor affecting the activation level of the corresponding FBIA. It is also used for adapting the parameters of this FBIA. RICPs are doing exactly the same job with relation to FICPs. The only difference is that because of the within-layer and theory factors the RICPs have a harder job detecting the activation of their corresponding FICPs compared with RBIAs. To reduce this effect, the weight assigned to the RICPs is in general less than that assigned to RBIAs in the effect channels connected to their respective forward processes.

5 Evaluation

In this section we provide the results of an experiment to compare the performance of a gaze controller that uses the proposed DUD behavior generation mechanism with the dynamic structure gaze controller developed using L_0EICA architecture which is a state-of-the-art gaze controller able of achieving human-like behavior [13].

In this experiment we used the explanation scenario in which a human is explaining to the robot how to assemble and use a Polymate device as our object of explanation. The device is disassembled and put in a table between the listener and the instructor The instructor explains to the listener how to use the device for measuring different kinds of physiological signals.

The experiment was designed as an internet poll to measure third person subjective evaluation. Subjects where recruited from university students and staff. 38 subjects participated in the poll but the data of 3 of them was corrupted so we used only 35 subjects (ages between 24 and 43 including 8 females). The procedure of the session was as follows: Firstly, the subject is informed about the procedure. Secondly, the subject answers the six questions to measure background information. After that the subject watches two videos: one showing the dynamic L_0EICA based gaze controller ([13]) and the other shows the controller controlled by the proposed DUD mechanism. After each session the subject ranks the robot from 1 (worst) to 7 (best) in the following evaluation dimensions: Attention, Naturalness, Understanding instructor's explanation, Human-likeness, Instructor's comfort, and Complexity of underlying algorithm. Each subject is then asked to select his/her preferred robot from the two videos.

Table 1 shows the mean and standard deviation of each evaluation dimension for the two controllers as well as the total statistics. The table shows that both controllers achieved quit acceptable performance with an average rate of 4.94 for the DUD based controller and 4.49 for L_0EICA controller. The improvement achieved by the proposed controller is statistically significant according to two-samples t-test with p-value equal to 0.0243 for the total score. Naturalness, Human-likeness, and Comfort of the instructor dimensions all showed statistically significant differences all in the favor of the proposed approach.

Table 1. Comparison between DUD based and L_0EICA Gaze Controllers. Dimensions showing statistically significant differences are shown in bold.

	DUD		L_0EICA		Total	
Dimension	Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Attention	4.66	1.33	4.63	1.68	4.64	1.50
Naturalness	5.40	1.63	4.11	1.69	4.76	1.77
Understanding	3.63	1.70	3.71	2.00	3.67	1.85
Human-Likeness	5.26	1.48	4.49	1.69	4.87	1.62
Comfort	5.66	1.16	4.86	1.35	5.26	1.32
Complexity	5.06	2.00	5.11	2.00	5.09	1.98
Total	4.94	0.78	4.49	0.89	4.71	0.86

21 subjects selected DUD based controller as their preferred controller compared with 14 subjects for L_0EICA. This again supports the superiority of the proposed approach over the bottom-up approach used in L_0EICA controller.

To confirm that this improvement in the behavior is resulting from the DUD algorithm rather than the differences in the behaviors themselves between the two controllers, the DUD controller was compared to the situation in which either the Top-down or the Bottom-up directions are disabled (called D and U controllers later). Six sessions in which the instructor interacted with a human listener were conducted and the gaze direction of the listener was logged. The average Levenshtein distance between the human behavior and the three controllers was calculated. DUD achieved an average distance of 6 compared with 31 for D and 27 for U controllers. The differences were both statistically significant with p-values less than 0.01. This improvement of the DUD algorithm may be explained by its ability to combine both both directed activation in the Top-down path with situation aware activation in the Bottom-up path.

6 Conclusion

This paper presented a novel asynchronous behavior generation mechanism called Down-Up-Down (DUD) that can naturally combine bottom-up and top-down approaches for behavior generation. The proposed mechanism is based on the simulation theory for acquiring a theory of mind in humans.

The proposed mechanism works on a hierarchy of interacting behaviors called interaction control processes and provides the means for determining the activation level of each one of these processes. The proposed system was compared to a traditional bottom-up action generation mechanism in gaze control and statistical analysis of the results showed that the proposed approach provide more natural, comforting, and human-like behavior. In the future an objective physiological based evaluation will be conducted to support this result.

References

1. Brooks, R.A.: A robust layered control system for a mobile robot. IEEE Journal of Robotics and Automation 2(1), 14–23 (1986)
2. Arkin, R.C., Masahiro Fujita, T.T., Hasegawa, R.: An ethological and emotional basis for human-robot interaction, robotics and autonomous systems. Robotics and Autonomous Systems 42(3-4), 191–201 (2003)
3. Yang, L., Yue, J., Zhang, X.: Hybrid control architecture in bio-mimetic robot, pp. 5699–5703 (June 2008)
4. Ulam, P., Arkin, R.: Biasing behavioral activation with intent for an entertainment robot. Intelligent Service Robotics 1(3), 195–209 (2008)
5. Wohlschlager, A., Gattis, M., Bekkering, H.: Action generation and action perception in imitation:an instance of the ideomotor principle. Phil. Trans. R. Soc. Lond. (358), 501–515 (2003)
6. Meltzoff, A., Moore, M.: Imitation, memory and the representation of persons. Infant Behav. Dev. (17), 83–99 (1994)
7. Mohammad, Y., Nishida, T.: Toward combining autonomy and interactivity for social robots. AI & Society (in press, 2008)
8. Breazeal, C.: Socially intelligent robots. Interactions 12(2), 19–22 (2005)
9. Davies, M.: Stone: Mental Simulation. Blackwell Publishers, Oxford (1995)
10. Sabbagh, M.A.: Understanding orbitofrontal contributions to theory-of-mind reasoning: Implications for autism. Brain and Cognition (55), 209–219 (2004)
11. Oberman, L.M., McCleery, J.P., Ramachandran, V.S., Pineda, J.A.: Eeg evidence for mirror neuron activity during the observation of human and robot actions: Toward an analysis of the human qualities of interactive robots. Neurocomput. 70(13-15), 2194–2203 (2007)
12. Mohammad, Y., Nishida, T.: Learning Interaction Structure using a Hierarchy of Dynamical Systems. Studies in Computational Intelligence. In: Opportunities and Challenges for Next-Generation Applied Intelligence, pp. 253–258. Springer, Heidelberg (2009)
13. Mohammad, Y., Nishida, T.: Controlling gaze with an embodied interactive control architecture. In: Applied Intelligence (2009) (Published online May 30)

Music-Ensemble Robot That Is Capable of Playing the Theremin While Listening to the Accompanied Music

Takuma Otsuka[1], Takeshi Mizumoto[1], Kazuhiro Nakadai[2], Toru Takahashi[1], Kazunori Komatani[1], Tetsuya Ogata[1], and Hiroshi G. Okuno[1]

[1] Graduate School of Informatics, Kyoto University, Kyoto, Japan
{ohtsuka,mizumoto,tall,komatani,ogata,okuno}@kuis.kyoto-u.ac.jp
[2] Honda Research Institute Japan, Co., Ltd., Saitama, Japan
nakadai@jp.honda-ri.com

Abstract. Our goal is to achieve a musical ensemble among a robot and human musicians where the robot listens to the music with its own microphones. The main issues are (1) robust beat-tracking since the robot hears its own generated sounds in addition to the accompanied music, and (2) robust synchronizing its performance with the accompanied music even if humans' musical performance fluctuates. This paper presents a music-ensemble Thereminist robot implemented on the humanoid HRP-2 with the following three functions: (1) self-generated Theremin sound suppression by semi-blind Independent Component Analysis, (2) beat tracking robust against tempo fluctuation in humans' performance, and (3) feedforward control of Theremin pitch. Experimental results with a human drummer show the capability of this robot for the adaptation to the temporal fluctuation in his performance.

Index Terms: Music robot, Musical human-robot interaction, Beat tracking, Theremin.

1 Introduction

To realize a joyful human-robot interaction and make robots more friendly, music is a promising medium for interactions between humans and robots. This is because music has been an essential and common factor in most human cultures. Even people who do not share a language can share a friendly and joyful time through music although natural communications by other means are difficult. Therefore, *music robots* that can interact with humans through music are expected to play an important role in natural and successful human-robot interactions.

Our goal is to achieve a musical human-robot ensemble, or *music-ensemble robot*, by using the robot's microphones instead of using symbolic musical representations such as MIDI signals. "Hearing" music directly with the robot's "ears", i.e. the microphones, like humans do is important for naturalness in the musical interaction because it enables us to share the acoustic sensation.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 102–112, 2010.

We envision a robot capable of playing a musical instrument and of synchronizing its performance with the human's accompaniment. However, the difficulty resides in that human's musical performance often includes many kinds of fluctuations and variety of music sounds. For example, we play a musical instrument with a temporal fluctuation, or when we sing a song, the pitch often vibrates.

Several music ensemble robots have been presented in a robotics field. However, the ensemble capability of these robots remains immature in some ways. A. Alford *et al.* developed a robot that plays the theremin [1], however, this robot is intended to play the theremin without any accompaniments. Petersen *et al.* reported a musical ensemble between a flutist robot, WF-4RIV, and a human saxophone player [2]. However, this robot takes turns with the human playing musical phrases. Therefore, the musical ensemble in a sense of performing simultaneously is yet to be achieved. Weinberg *et al.* developed a percussionist robot called "Haile" that improvises its drum with human drum players [3],[4]. The adaptiveness in this robot to the variety of human's performance is limited. For example, this robot allows for little tempo fluctuation in the human performance, or it assumes a specific musical instrument to listen to.

This paper presents a robot capable of music ensemble with a human musician by playing the electric musical instrument called theremin [5]. This robot plays the theremin while it listens to the music and estimates the tempo of the human accompaniment by adaptive beat tracking method. The experiment confirms our robot's adaptiveness to the tempo fluctuation by the live accompaniment.

2 Beat Tracking-Based Theremin Playing Robot

2.1 Essential Functions for Music Robots

Three functions are essential to the envisioned music ensemble robot: (1) listening to the music, (2) synchronizing its performance with the music, and (3) expressing the music in accordance with the synchronization.

The first function works as a preprocessing of the music signal such that the following synchronization with the music is facilitated. Especially, self-generated sound such as robot's own voice that is mixed into the music sound robot hears has been revealed to affect the quality of subsequent synchronization algorithm [6][7].

The second function extracts some information necessary for the robot's accompaniment to the human's performance. The tempo, the speed of the music, and the time of beat onsets, where you step or count the music, are the most important pieces of information to achieve a musical ensemble. Furthermore, the robot has to be able to predict the coming beat onset times for natural synchronization because it takes the robot some time to move its body to play the musical instrument.

The third function determines the behavior of the robot based on the output of the preceding synchronization process. In the case of playing the musical instrument, the motion is generated so that the robot can play a desired phrase at the predicted time.

2.2 System Architecture

Figure 1 outlines our beat tracking-based theremin player robot. This robot has three functions introduced in Section 2.1. The robot acquires a mixture sound of human's music signal and the theremin sound and plays the theremin in synchronization with the input music.

For the first listening function, independent component analysis (ICA)-based self-generated sound suppression [8] is applied to suppress the theremin sound from the sound that the robot hears. The inputs for this suppression method are the mixture sound and the clean signal of the theremin. The clean signal from the theremin is easily acquired because theremin directly generates an electric waveform.

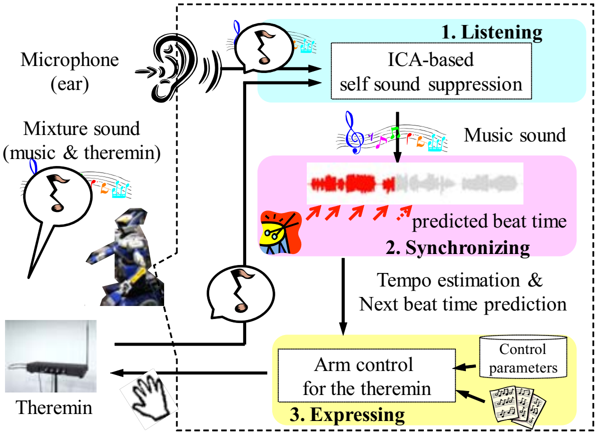


Fig. 1. The architecture of our theremin ensemble robot

The second synchronization function is a tempo-adaptive beat tracking called spectro-temporal pattern matching [7]. The algorithm consists of tempo estimation, beat detection, and beat time prediction. Some formulations are introduced in Section 3.1.

The third expression function is a the pitch and timing control of the theremin. A theremin has two antennae: vertical one for pitch control and horizontal one for volume control. The most important feature of a theremin is a **proximity control**: without touching it, we can control a theremin's pitch and volume [5]. As a robot's arm gets closer to the pitch-control antenna, the theremin's pitch increases monotonically and nonlinearly. The robot plays a given musical score in synchronization with the human's accompaniment.

3 Algorithm

3.1 Beat Tracking Based on Spectro-temporal Pattern Matching

This beat tracking algorithm has three phases: (1) tempo estimation, (2) beat detection, and (3) beat time prediction. The input is a music signal of human performance after the self-generated sound suppression.

Tempo estimation. Let $P(t,f)$ be the mel-scale power spectrogram of the given music signal where t is time index and f is mel-filter bank bin. We use 64 banks, therefore $f = 0, 1, ..., 63$. Then, Sobel filtering is applied to $P(t,f)$ and the onset belief $d_{inc}(t,f)$ is derived.

$$d_{inc}(t,f) = \begin{cases} d(t,f) \text{ if } \ d(t,f) > 0, \\ 0 \qquad\qquad \text{otherwise,} \end{cases} \tag{1}$$

$$\begin{aligned} d(t,f) = & -P(t-1,f+1) + P(t+1,f+1) \\ & -2P(t-1,f) + 2P(t+1,f) \\ & -P(t-1,f-1) + P(t+1,f-1), \end{aligned} \tag{2}$$

where $f = 1, 2, ..., 62$. Equation (2) shows the Sobel filter.

The tempo is defined as the interval of two neighboring beats. This is estimated through Normalized Cross Correlation (NCC) as Eq. (3).

$$R(t,i) = \frac{\displaystyle\sum_{f=1}^{62}\sum_{k=0}^{W-1} d_{inc}(t-k,f)d_{inc}(t-i-k,f)}{\sqrt{\displaystyle\sum_{f=1}^{62}\sum_{k=0}^{W-1} d_{inc}(t-k,f)^2 \cdot \sum_{f=1}^{62}\sum_{k=0}^{W-1} d_{inc}(t-i-k,f)^2}} \tag{3}$$

where W is a window length for tempo estimation and i is a shift offset. W is set to be 3 [sec]. To stabilize the tempo estimation, the local peak of $R(t,i)$ is derived as

$$R_p(t,i) = \begin{cases} R(t,i) \text{ if } \ R(t,i-1) < R(t,i) \text{ and } R(t,i+1) < R(t,i) \\ 0 \qquad\qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{4}$$

For each time t, the beat interval $I(t)$ is determined based on $R_p(t,i)$ in Eq. (4). The beat interval is an inverse number of the musical tempo. Basically, $I(t)$ is chosen as $I(t) = \underset{i}{\text{argmax}}\, R_p(t,i)$. However, when a complicated drum pattern is performed in the music signal, the estimated tempo will fluctuate rapidly.

To avoid the mis-estimation of the beat interval, $I(t)$ is derived as Eq. (5). Let I_1 and I_2 be the first and the second peak in $R_p(t,i)$ when moving i.

$$I(t) = \begin{cases} 2\|I_1 - I_2\| \text{ if } (\|I_{n2} - I_1\| < \delta \ \text{ or } \|I_{n2} - I_2\| < \delta) \\ 3\|I_1 - I_2\| \text{ if } (\|I_{n3} - I_1\| < \delta \ \text{ or } \|I_{n3} - I_2\| < \delta) \\ I_1 \qquad\qquad\qquad\qquad\qquad \text{otherwise,} \end{cases} \tag{5}$$

where $I_{n2} = 2\|I_1 - I_2\|$ and $I_{n3} = 3\|I_1 - I_2\|$. δ is an error margin parameter.

The beat interval $I(t)$ is confined to the range between 61 – 120 beats per minute (bpm). This is because this range is suitable for the robot's arm control.

Beat detection. Each beat time is estimated using the onset belief $d_{inc}(t, f)$ and the beat interval $I(t)$. Two kinds of beat reliabilities are defined: neighboring beat reliability and continuous beat reliability. Neighboring beat reliability $S_n(t, i)$ defined as Eq. (6) is a belief that the adjacent beat lies at $I(t)$ interval.

$$S_n(t,i) = \begin{cases} \sum_{f=1}^{62} d_{inc}(t-i,f) + \sum_{f=1}^{62} d_{inc}(t-i-I(t),f) & \text{if } (i \le I(t)), \\ 0 & \text{if } (i > I(t)) \end{cases} \tag{6}$$

Continuous beat reliability $S_c(t, i)$ defined as Eq. (7) is a belief that the sequence of musical beats lies at the estimated beat intervals.

$$S_c(t,i) = \sum_{m=0}^{N_{beats}} S_n(T_p(t,m), i), \tag{7}$$

$$T_p(t,m) = \begin{cases} t - I(t) & \text{if } (m = 0), \\ T_p(t, m-1) - I(T_p(t,m)) & \text{if } (m \ge 1), \end{cases}$$

where $T_p(t, m)$ is the m-th previous beat time at time t, and N_{beats} is the number of beats to calculate the Continuous beat reliability.

Then, these two reliabilities are integrated into beat reliability $S(t)$ as

$$S(t) = \sum_i S_n(t-i,i) \cdot S_c(t-i,i). \tag{8}$$

The latest beat time $T(n + 1)$ is one of the peak in $S(t)$ that is closest to $T(n) + I(t)$, where $T(n)$ the n-th beat time.

Beat time prediction. Predicted beat time T' is obtained by extrapolation using the latest beat time $T(n)$ and the current beat interval $I(t)$.

$$T' = \begin{cases} T_{tmp} & \text{if } \; T_{tmp} \ge \frac{3}{2}I(t) + t, \\ T_{tmp} + I(t) & \quad\text{otherwise}, \end{cases} \tag{9}$$

$$T_{tmp} = T(n) + I(t) + (t - T(n)) - \{(t - T(m)) \mod I(t)\} \tag{10}$$

3.2 Theremin Pitch Control by Regression Parameter Estimation

We have proposed a theremin's model-based feedforward pitch control method for a thereminist robot in our previous work [9]. We introduce our method in the following order: model formulation, parameter estimation and feedforward arm control.

Arm-position to pitch model. We constructed a model that represents a relationship between a theremin's pitch and a robot's arm position. According to the fact that a theremin's pitch increases monotonically and nonlinearly, we formulated our model as follows:

$$\hat{p} = M_p(x_p; \boldsymbol{\theta}) = \frac{\theta_2}{(\theta_0 - x_p)^{\theta_1}} + \theta_3 \tag{11}$$

where, $M_p(x_p; \boldsymbol{\theta})$ denotes our pitch model, x_p denotes a pitch-control arm, $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3)$ denotes model parameters, $\hat{p}$ denotes an estimated pitch using a pitch model ([Hz]).

Parameter estimation for theremin pitch control. To estimate model parameters, $\boldsymbol{\theta}$, we obtain a set of learning data as following procedure: at first, we equally divide a range of robot's arm into N pieces (we set $N = 15$). For each boundary of divided pieces, we extract a theremin's corresponding pitch. Then, we can obtain a set of learning data, i.e., pairs of pitch-control arm positions ($x_{pi}, i = 0 \cdots N$) and corresponding theremin's pitches ($p_i, i = 0 \cdots N$). Using the data, we estimate model parameters with Levenberg-Marquardt (LM) method, which is one of a nonlinear optimization method. As an evaluation function, we use a difference between measured pitch, p_i, and estimated pitch, $M_p(x_{pi}, \boldsymbol{\theta})$.

Feedforward arm control. Feedforward arm control has two aspects: arm-position control and timing control. A musical score is prepared for our robot to play the melody. The musical score consists of two elements: the note name that determines the pitch and the note length that relates to the timing control.

To play the musical notes in a correct pitch, a musical score is converted into a sequence of arm position. We first convert musical notes (e.g., $C4$, $D5$, ..., where the number means the octave of each note) into a sequence of corresponding pitches based on equal-temperament:

$$p = 440 \cdot 2^{o-4} \sqrt[12]{2^{n-9}}, \tag{12}$$

where p is the pitch for the musical note, o is the octave number. The variable n in Eq. (12) represents the pitch class where $n = 0, 1, ..., 11$ correspond to $C, C\sharp, ..., B$ notes, respectively. Then, we give the pitch sequence to our inverse pitch model:

$$\hat{x_p} = M_p^{-1}(p, \boldsymbol{\theta}) = \theta_0 - \left(\frac{\theta_2}{p - \theta_3}\right)^{1/\theta_1} \tag{13}$$

where, $\hat{x}_p$ denotes an estimated robot's arm position. Finally, we obtain a sequence of target arm positions. By connecting these target positions linearly, the trajectory for a thereminist robot is generated.

The timing of each note onset, the beginning of a musical note, is controlled using the predicted beat time T' in Eq. (9) and the current beat interval $I(t)$ in Eq. (5). When T' and $I(t)$ are updated, the arm controller adjusts the timing such that the next beat comes at time T', and the time duration of each note is calculated by multiplying relative note length such as quarter notes by $I(t)$.

4 Experimental Evaluation

This section presents the experimental results of our beat tracking-based thereminist robot. Our experiments consist of two parts. The first experiment proves our robot's capability of quick adaptation to tempo change and robustness against the variety of musical instruments. The second experiment shows that our robot is able to play the theremin with a little error even when fluctuations in human's performance are observed.

4.1 Implementation on Humanoid Robot HRP-2

We implemented our system on a humanoid robot HRP-2 in Fig. 3 [10]. Our system consists of two PCs. The ICA-based self-generated sound suppression and the beat tracking system is implemented by C++ on MacOSX. The arm control for the theremin performance is implemented by Python on Linux Ubuntu 8.04. The predicted beat time T' and the beat interval $I(t)$ are sent to the arm controller through socket communication at time $T' - \Delta t$, where Δt is the delay in the arm control. Δt is set 40 [msec] empirically.

The settings for the beat tracking is as follows: the sampling rate is 44100 [Hz], the window size for fast Fourier transform is 4096 [pt], and the hop size of the window is 512 [pt]. For the acoustic input, a monaural microphone is attached to the HRP-2's head as indicated in Fig. 3.

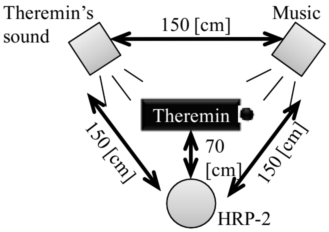


Fig. 2. Experimental setup

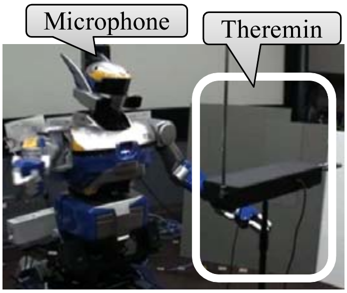


Fig. 3. Humanoid robot HRP-2

4.2 Experiment 1: The Influence of the Theremin on Beat Tracking

Figure 2 shows the experimental setup for the experiment 1. The aim of this experiment is to reveal the influence of theremin's sound on the beat tracking algorithm. Music sound comes out of the right loudspeaker while the robot is playing the theremin and its sound comes out of the left loudspeaker.

The music signal used in the experiment is three minutes long that is the excerpts from three popular music songs in RWC music database (RWC-MDB-P-2001) developed by Goto *et al* [11]. These three songs are No. 11, No. 18,

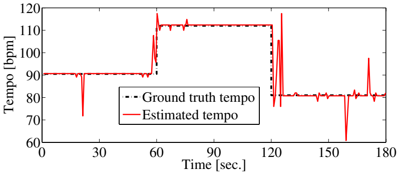


Fig. 4. Tempo estimation result w/ self-generated sound suppression

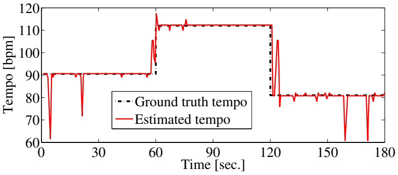


Fig. 5. Tempo estimation result w/o self-generated sound suppression

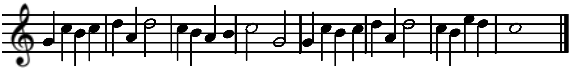


Fig. 6. The musical score of Aura Lee

No. 62. The tempo for each song is 90, 112, 81 [bpm], respectively. One-minute excerpts are concatenated to make the three-minute music signal.

Figure 4 and 5 are the tempo estimation results. The self-generated sound suppression is active in Fig. 4 while it is disabled in Fig. 5. The black line shows the ground truth tempo, and the red line shows the estimated tempo.

These results prove prompt adaptation to the tempo change and robustness against the variety of the musical instruments used in these music tunes. On the other hand, a little influence of the theremin sound on the beat tracking algorithm is observed. This is because theremin's sound does not have impulsive characteristics that mainly affect the beat tracking results. Though the sound of theremin has little influence on the beat tracking, self-generated sound suppression is generally necessary.

4.3 Experiment 2: Theremin Ensemble with a Human Drummer

In this experiment, a human drummer stands in the position of the right loudspeaker in Fig. 2. At first, the drummer beats the drum slowly, then he hastes the drum beating. The robot plays the first part of "Aura Lee," American folk song. The musical score is shown in Fig. 6.

Figure 7 and 8 show the ensemble of the thereminist robot and the human drummer. Top plots indicate the tempo of human's drumming and estimated tempo by the system. Middle plots are the theremin's pitch trajectory in a red line and human's drum-beat timings in black dotted lines. The bottom plots show the onset error between human's drum onsets and the theremin's note onsets. Positive error means the theremin onset is earlier than the drum onset. The pitch trajectories of the theremin are rounded off to the closest musical note on a logarithmic frequency axis.

The top tempo trajectory shows that the robot successfully tracked the tempo fluctuation in the human's performance whether the self-generated sound suppression because the tempo, or the beat interval, is estimated after a beat is

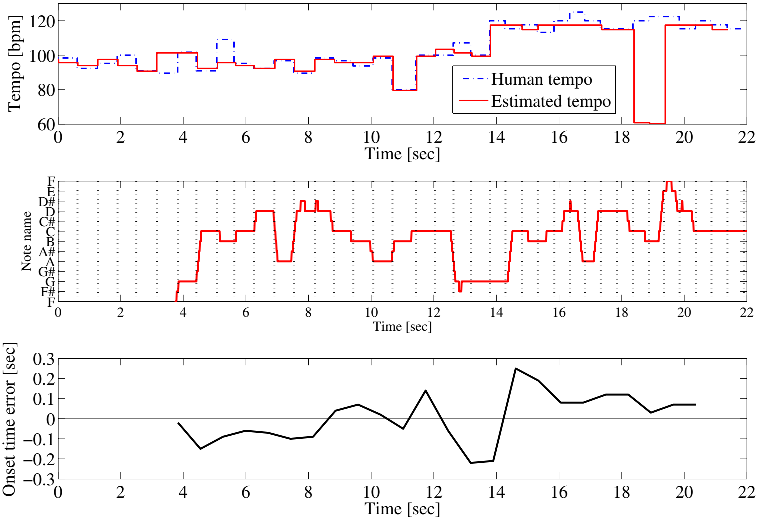


Fig. 7. Theremin ensemble with human drum w/ self-generated sound suppression Top: tempo trajectory, Mid: theremin pitch trajectory, Bottom: Onset time error

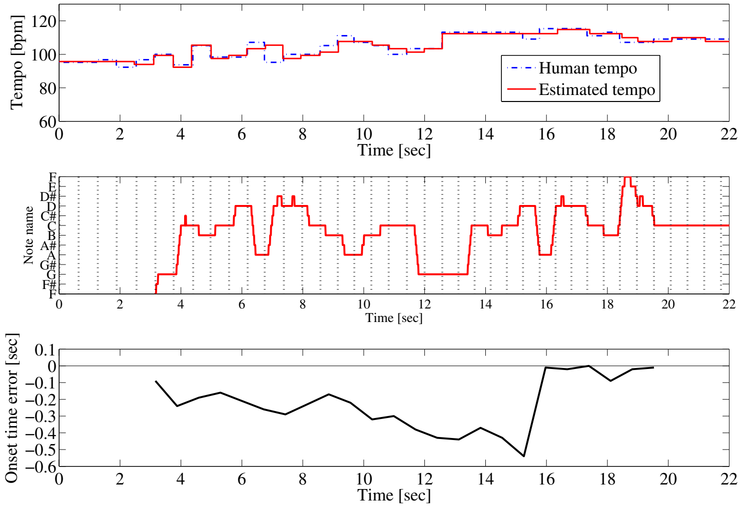


Fig. 8. Theremin ensemble with human drum w/o self-generated sound suppression Top: tempo trajectory, Mid: theremin pitch trajectory, Bottom: Onset time error

observed. However, some error was observed between the human drum onsets and theremin pitch onsets especially around 13 [sec], where the human player hastes the tempo. The error was then relaxed from 16 [sec], about 6 beat onsets after the tempo change.

The error at the bottom plot of Fig. 8 first gradually increased toward a negative value. This is because the human drummer hasted its performance gradually, therefore, the robot did not catch up the speed and produced an increasing negative error value. The error at the bottom plot of Fig. 7 went zigzag because both the human and the robot tried to synchronize their own performance with the other's. The mean and standard deviation of the error for Fig. 7 and 8 were 6.7 ± 120.2 [msec] and -225.8 ± 157.0 [msec], respectively. It took 3–4 [sec] the robot before it starts playing the theremin because this time is necessary to estimate the tempo with stability.

5 Conclusion

This paper presented a robot capable of playing the theremin with human's accompaniment. This robot has three functions for the ensemble: (1) the ICA-based self-generated sound suppression for the listening function, (2) the beat tracking algorithm for the synchronization function, (3) the arm control to play the theremin in a correct pitch for the expression function.

The experimental results revealed our robot's capability of adaptiveness to the tempo fluctuation and of robustness against the variety of musical instruments. The results also suggest that the synchronization error increases when the human player gradually changes his tempo.

The future works are as follows: First, this robot currently considers the beat in the music. For richer musical interaction, the robot should allow for the pitch information in the human's performance. Audio to score alignment [12] is promising technique to achieve a pitch-based musical ensemble. Second, the ensemble with multiple humans is a challenging task because the synchronization becomes even harder when all members try to adapt to another member. Third, this robot requires some time before it joins the ensemble or the ending of the ensemble is still awkward. To start and conclude the ensemble, quicker adaptation is preferred.

Acknowledgments. A part of this study was supported by Grant-in-Aid for Scientific Research (S) and Global COE Program.

References

1. Alford, A., et al.: A music playing robot. In: FSR 1999, pp. 29–31 (1999)
2. Petersen, K., Solis, J.: Development of a Aural Real-Time Rhythmical and Harmonic Tracking to Enable the Musical Interaction with the Waseda Flutist Robot. In: Proc. of IEEE/RSJ Int'l Conference on Intelligent Robots and Systems (IROS), pp. 2303–2308 (2009)
3. Weinberg, G., Driscoll, S.: Toward Robotic Musicianship. Computer Music Journal 30(4), 28–45 (2006)
4. Weinberg, G., Driscoll, S.: The interactive robotic percussionist: new developments in form, mechanics, perception and interaction design. In: Proc. of the ACM/IEEE Int'l Conf. on Human-robot interaction, pp. 97–104 (2007)

5. Glinsky, A.V.: The Theremin in the Emergence of Electronic Music. PhD thesis, New York University (1992)
6. Mizumoto, T., Takeda, R., Yoshii, K., Komatani, K., Ogata, T., Okuno, H.G.: A Robot Listens to Music and Counts Its Beats Aloud by Separating Music from Counting Voice. In: IROS, pp. 1538–1543 (2008)
7. Murata, K., Nakadai, K., Yoshii, K., Takeda, R., Torii, T., Okuno, H.G., Hasegawa, Y., Tsujino, H.: A Robot Uses Its Own Microphone to Synchronize Its Steps to Musical Beats While Scatting and Singing. In: IROS, pp. 2459–2464 (2008)
8. 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: IROS, pp. 1718–1723 (2008)
9. Mizumoto, T., Tsujino, H., Takahashi, T., Ogata, T., Okuno, H.G.: Thereminist Robot: Development of a Robot Theremin Player with Feedforward and Feedback Arm Control based on a Theremin's Pitch Model. In: IROS, pp. 2297–2302 (2009)
10. Kaneko, K., Kanehiro, F., Kajita, S., Hirukawa, H., Kawasaki, T., Hirata, M., Akachi, K., Isozumi, T.: Humanoid robot HRP-2. In: Proc. of IEEE Int'l Conference on Robotics and Automation (ICRA), vol. 2, pp. 1083–1090 (2004)
11. Goto, M., Hashiguchi, H., Nishimura, T., Oka, R.: RWC Music Database: Popular Music Database and Royalty-Free Music Database. IPSJ Sig. Notes 2001(103), 35–42 (2001)
12. Dannenberg, R., Raphael, C.: Music Score Alignment and Computer Accompaniment. Communications of the ACM 49(8), 38–43 (2006)

Self-organisation of an 802.11 Mesh Network

John Debenham and Ante Prodan

Centre for Quantum Computation & Intelligent Systems,
University of Technology, Sydney, Australia
debenham@it.uts.edu.au

Abstract. The self-organisation of telecommunications networks has to confront the two challenges of the scalability and the stability of the solution. This paper describes a distributed, co-operative multiagent system in which agents make decisions based only on local knowledge — that guarantees scalability. Extensive simulations indicate that stability is ensured by the agent's making improvements to the network settings that improve the social performance for all agents in a two-hop range. Our overall goal is simply to reduce maintenance costs for such networks by removing the need for humans to tune the network settings.

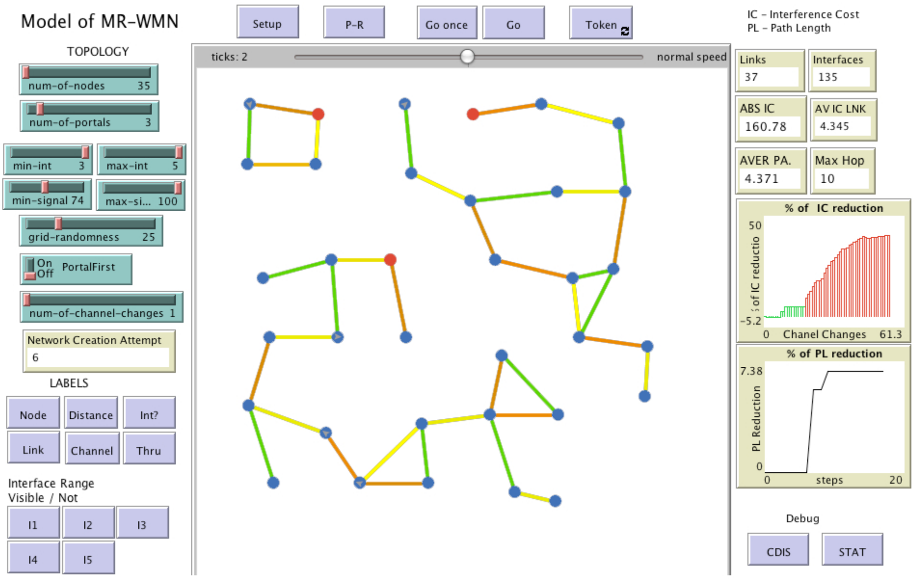
1 Introduction

Recent work on 802.11 Mesh Networks, such as [1], is predicated on a network whose prime purpose is to route traffic to and from nodes connected to the wired network — in which case there is assumed to be no traffic between end-user nodes. This introduces the conceptual simplification that mesh nodes can be seen as being grouped into clusters around a wired node where each cluster has a tree-like structure, rooted at a wired node, that supports the traffic. This is the prime purpose of 802.11 Mesh Networks in practice. Where possible, we move away from any assumptions concerning tree-like structures with the aim of designing algorithms for the more general classes of "wireless ad-hoc networks" or "wireless mesh networks". This paper is based on previous work in the area of mesh networking, and in particular in distributed algorithms at Columbia University, Microsoft Research, University of Maryland and Georgia Institute of Technology. See also: [2], [3], [4] and [5].

There are three principal inputs that we assume are available to the methods described: a load model, a load-balancing algorithm and an interference model. The work described below makes no restrictions on these three inputs other than that they are available to every node in the mesh. The load model, and so too the load balancing algorithm, will only be of value to a method for self-organisation if together they enable future load to be predicted with some certainty. We assume that the load is predictable. We assume that the external demands on a set of nodes S are known and that there is a *load balancing algorithm* — that may or may not be intelligent — that determines how the load is routed through S. We assume that the load balancing algorithm will determine how the load is allocated to each link in the mesh.

The measurement of interference cost is discussed in Section 2. Methods for adjusting the channels in a multi-radio mesh networks for predictable load are described

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 113–122, 2010.



Model of MR-WMN

TOPOLOGY

num-of-nodes 35

num-of-portals 3

min-int 3

max-int 5

min-signal 74

max-si... 100

grid-randomness 25

On Off PortalFirst

num-of-channel-changes 1

Network Creation Attempt 6

ticks: 2

normal speed

IC - Interference Cost
PL - Path Length

Links 37

Interfaces 135

ABS IC 160.78

AV IC LNK 4.345

AVER PA. 4.371

Max Hop 10

Fig. 1. The implementation of the algorithms

in Section 3, and for adjusting the links in Section 4. Future plans are described in Section 5.

2 Measuring Interference Cost

Suppose that during some time interval Δt two interfaces a and b are transmitting and receiving on channels Γ_a and Γ_b. During Δt, the *interference limit* that interface x imposes on interface y, $\tau_{y|x}$, is a ratio being the loss of traffic volume that interface y could receive if interface x were to transmit persistently divided by the volume of traffic that interface y could receive if interface x was silent:

$$\tau_{y|x} = \frac{(m_y \mid \text{ interface } x \text{ silent}) - (m_y \mid \text{ interface } x \text{ persistent})}{m_y \mid \text{ interface } x \text{ silent}}$$

where m_y is the mean SNIR (i.e. Signal to Noise Plus Interference Ratio) observed by interface y whilst listening on channel Γ_y, where as many measurements are made as is expedient in the calculation of this mean[1]. The *interference load* of each interface, v_a and v_b, is measured as a proportion, or percentage, of some time interval during which that interface is transmitting. Then the *observed interference* caused by interface

[1] For $\tau_{y|x}$ to have the desired meaning, m_y should be a measurement of *link throughput*. However, link throughput and SNIR are approximately proportional — see [6].

b transmitting on channel Γ_b as experienced by interface a listening on channel Γ_a is: $\tau_{a|b} \times v_b$, and the *observed interference cost* to interface a is[2]:

$$f(a \mid b) \triangleq \tau_{a|b} \times v_b \times (1 - v_a)$$

and so to interface b:

$$f(b \mid a) = \tau_{b|a} \times v_a \times (1 - v_b)$$

Now consider the interference between one interface a and two other interfaces c and d. Following the argument above, the *observed interference* caused by interfaces c and d as experienced by interface a is[3]: $\tau_{a|c} \times v_c + \tau_{a|d} \times v_d - \tau_{a|\{c,d\}} \times v_c \times v_d$. The observed interference cost to interface a is:

$$f(a \mid \{c,d\}) = (1 - v_a) \times \left(\tau_{a|c} \times v_c + \tau_{a|d} \times v_d - \tau_{a|\{c,d\}} \times v_c \times v_d\right)$$

If interfaces at agents c and d are linked then they will transmit on the same channel Γ_β, and we ignore the possibility of them both transmitting at the same time[4]. Further suppose that v_β is the proportion of Δt for which either interface c or interface d is transmitting. Then for some κ_β, $0 \leq \kappa_\beta \leq 1$: $v_c = \kappa_\beta \times v_\beta$, and $v_d = (1 - \kappa_\beta) \times v_\beta$. Thus:

$$f(a \mid \beta) = (1 - v_a) \times v_\beta \times \left(\tau_{a|c} \times \kappa_\beta + \tau_{a|d} \times (1 - \kappa_\beta)\right)$$

Now suppose that interfaces a and b are linked, and that v_α is the proportion of Δt for which either interface a or interface b is transmitting. Then for some κ_α, $0 \leq \kappa_\alpha \leq 1$: $v_a = \kappa_\alpha \times v_\alpha$, $v_b = (1 - \kappa_\alpha) \times v_\alpha$. Then as a will only receive interference when it is listening to b transmitting:

$$f(a \mid \beta) = v_b \times v_\beta \times \left(\tau_{a|c} \times \kappa_\beta + \tau_{a|d} \times (1 - \kappa_\beta)\right)$$

and so:

$$\begin{aligned} f(\alpha \mid \beta) = (1 - \kappa_\alpha) \times v_\alpha \times v_\beta \times \left(\tau_{a|c} \times \kappa_\beta + \tau_{a|d} \times (1 - \kappa_\beta)\right) \\ + \kappa_\alpha \times v_\alpha \times v_\beta \times \left(\tau_{b|c} \times \kappa_\beta + \tau_{b|d} \times (1 - \kappa_\beta)\right) \end{aligned} \tag{1}$$

Note that v_α, v_β, κ_α and κ_β are provided by the load model, and the $\tau_{x|y}$ are provided by the interference model.

3 Adjusting the Channels

Each node is seen as an agent. The multiagent system exploits the distinction between proactive and reactive reasoning [10]. Proactive reasoning is concerned with planning

[2] We assume here that whether or not interfaces a and b are transmitting are independent random events [7]. Then the probability that a is transmitting at any moment is v_a, and the probability that b is transmitting and a is listening at any moment is: $(1 - v_a) \times v_b$.

[3] That is, the interference caused by either interface c or interface d.

[4] The probability of two linked interfaces transmitting at the some time on an 802.11 mesh network can be as high as 7% — see [8], [9].

to reach some goal. Reactive reasoning is concerned with dealing with unexpected changes in the agent's environment. The reactive logic provides an "immediate fix" to serious problems. The proactive logic, that involves deliberation and co-operation of nearby nodes, is a much slower process.

An agent (i.e. a node, or physically a router) with omnidirectional interfaces has three parameters to set for each interface: 1) The channel that is assigned to that interface; 2) The interfaces that that interface is linked to, and 3) The power level of the interface's transmission. Methods are describe for these parameters in the following sections. The methods all assume that there is a load balancing algorithm.

Informally the proactive logic uses the following procedure:

- *Elect* a node a that will manage the process
- *Choose* a link α from a to another node — precisely a trigger criterion (see below) permits node a to attempt to improve the performance of one of its links $\alpha \ni a$ with a certain priority level.
- *Measure* the interference
- *Change* the channel setting if this leads to a mean improvement for all the agents in a's interference range

The following is a development of the ideas in [2].

```
choose node $a$ at time $t-2$;
set $V_a = \cup_{n \in S_a} S_n$;
$\forall x \in V_a$ transmit "propose organise$[a,x,p]$";
unless $\exists x \in V_a$ receive "overrule organise$[a,x,q]$" in
                $[t-2,t-1]$ where $q > p$ do {
    $\forall x \in V_a$ transmit "propose lock$[a,x,t,t+1]$";
    if $\forall x \in V_a$ receive "accept lock$[a,x,t,t+1]$" in $[t-1,t]$
    then {
        unless $\exists x \in V_a$ receive "reject lock$[a,x,t,t+1]$"
            do {improve $a$;}
    }
}
where: improve $a = \{$
    choose link $\alpha \ni a$ on channel $\Gamma^t_\alpha$;
    set $B \leftarrow \sum_{\beta \in S_\alpha} f(\alpha \mid \beta) + \sum_{\beta \in S_\alpha} f(\beta \mid \alpha)$;
    if (feasible) re-route $\alpha$'s traffic;
    for $\Gamma_\alpha = 1, \ldots, K, \Gamma_\alpha \neq \Gamma^t_\alpha$ do{
        if $\sum_{\beta \in S_\alpha} f(\alpha \mid \beta) + \sum_{\beta \in S_\alpha} f(\beta \mid \alpha) < B \times \varepsilon$ then{
            $\Gamma^{t+1}_\alpha \leftarrow \Gamma_\alpha$;
            selflock node $a$ in $[t+1, t+k]$;
            break;
        };
    };
    $\forall x \in V_a$ transmit "$\alpha$'s interference test signals";
    apply load balancing algorithm to $S_a$;
}
```

The statement **selflock** is to prevent a from having to activate the method too frequently. The constant $\varepsilon < 1$ requires that the improvement be 'significant' both for node a and for the set of nodes S_a. The stability of this procedure follows from the fact that it produces a net improvement of the interference cost within S_a. If a change of channel is effected then there will be no resulting change in interference outside S_a.

The above method reduces the net observed inference cost in the region V_a. It does so using values for the variables that appear on the right-hand side of Equation 1. If those values are fixed then the method will converge. The method above suggests the possibility that traffic is re-routed during the reassignment calculation — this is not essential.

The multiagent system described in the paper has been simulated; the simulation is available[5] on the World Wide Web. A screen shot is shown in Figure 1. The "Setup" button establishes a random topology in line with the TOPOLOGY settings on the left side. The "P-R" button reduces the path length, and the "Go" button reduces the interference cost using the algorithms described in this paper. The colours on the arcs denote which of the eleven 802.11 channels is used. The work has been conducted in collaboration with Alcatel-Lucent, Bell Labs, Paris.

3.1 Results and Discussion

The interference cost reduction for a link discussed herein is measured as the difference between absolute interference (AI) values obtained before the channel assignment process and after the channel assignment process. For example, if $AI_{before} = 5$ and $AI_{after} = 4$ the absolute difference is $AD = 1$ which is 20% decrease in the absolute interference. Consequently, the performance is always expressed as a percentage of the decrease. Our simulation studies consider realistic scenarios of different node densities and topologies in a typical wireless mesh network hence are more reflective of evaluating the true performance of the algorithm. In these studies the mean of interference cost (IC) reduction across all topologies and network (node) densities obtained is 36.7.

Impact of typical topologies on the interference cost. Figure 2(a) shows the variation in the interference cost reduction as a function of network topology across different node densities. It can be deduced that the impact of the topologies on the performance of the algorithm (i.e. in terms of interference cost reduction) is insignificant. The mean of IC reduction calculated from the data obtained shows that the topology with the smallest average IC reduction is the completely random with a mean of 36.02 and topology with the most IC reduction is the random grid with a mean of 37.12. The difference in performance between best and worst case is just 1.1 which confirms that the performance of the algorithm is almost completely independent of the type of topology.

Performance Comparison across the Network. In this study, we obtained interference cost (IC) in different regions of the MR-WMN (Multi Radio Wireless Mesh Network) for the same set of links before and after the self-organisation algorithm is invoked. Comparison of the results obtained is shown in Figure 2(b) where the Interference cost is on the X-axis. From Figure 2(b) we can see that there were no nodes (red dots) that

[5] `http://www-staff.it.uts.edu.au/~debenham/homepage/holomas/`

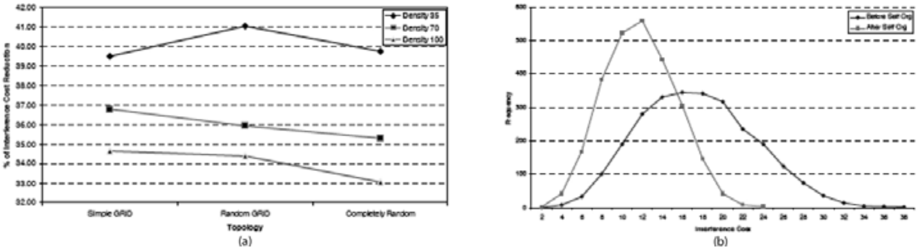


Fig. 2. (a) Interference cost reduction as a function of topologies. (b) Comparison of *IC* across the network before (blue) and after (red) self-organisation.

caused more interference after the self-organisation than it had caused before (blue dots) the self-organisation was invoked.

4 Adjusting the Links

The first path-reduction algorithm is precisely the same as the algorithm in Section 3 but with the following 'improve' methods.

Link adjustment with known traffic load. Suppose that node a has interference range S_a. Let M_a be the set of nodes in S_a excluding node a. Then use the method in Section 3 with the following 'improve' method:

```
improve $a = \{$
   **for** link $\alpha \ni a$, where $\alpha = [a,b]$
   suppose $\alpha$ is on channel $\Gamma^t_\alpha$;
   **set** $B \leftarrow \sum_{\beta \in S_\alpha} f(\alpha \mid \beta) + \sum_{\beta \in S_\alpha} f(\beta \mid \alpha)$;
   **if** (feasible) **re-route** $\alpha$'s traffic;
   **set** $\gamma \leftarrow \alpha$;
   **for** $y \in M_a$ **do** $\{$
      **for** $\Gamma_{[a,y]} = 1, \ldots, K$, **do** $\{$
         **if** $\sum_{\beta \in S_a} f([a,y] \mid \beta) + \sum_{\beta \in S_a} f(\beta \mid [a,y]) < B \times \varepsilon$
          **then** $\{$
               **set** $\gamma \leftarrow [a,y]$;
               **selflock** node $a$ **in** $[t+1, t+k]$;
               **break**;
         $\}$;
      $\}$;
   $\}$;
   $\forall x \in V_a$ **transmit** "$\gamma$'s interference test signals";
   **apply** load balancing algorithm to $S_a$;
$\}$
```

Trigger for attempting to adjust a link with known traffic load. Consider a mesh with known traffic load. Suppose that the load balancing algorithm has allocated load to links on the mesh, and let link $(a,b) = \arg\max_{x \in N_a^t} \rho(x)$. If replacing (a,b) with (a,x) would mean that there exists a cut through the mesh that traverses (a,x) and that all other links on that cut have a load $< \rho(a,b)$ then let node a initiate the link adjusting procedure. Likewise if replacing (a,b) with (y,b).

4.1 Reactive Logic

The relationship between the reactive and proactive logics is determined by:

```
if event [link $\alpha$ is broken] then {
  activate [activate the Reactive Method for link $\alpha$];
  $\forall x \in \alpha$ if state [node $x$ locked by "accept lock[$a,x,s,t$]"
  then {transmit "reject lock[$a,x,s,t$]";}
}
```

where the *Reactive Method* is as follows; it simply fixes disasters as they occur possibly with a configuration that is less satisfactory than the prior. It has no implications for neighbouring interfaces, and so it presents no instability issues.

Link adjustment with unknown traffic load. Suppose that node a has interference range S_a. Let M_a be the set of nodes in S_a excluding node a. For nodes $x,y \in S_a$, let $c(x,y)$ denote the cost[6] of the least cost path that connects x and y. We assume that: $(\forall x,y) c(x,y) = c(y,x)$, and that if the least cost path between nodes u and v is a subset of the least cost path between x and y then $c(u,v) \leq c(x,y)$. Let N_a^t be the set of links in S_a at time t, and $N_a^t(\ominus[a,x], \oplus[a,y])$ denotes the network configuration with link $[a,x]$ replaced by $[a,y]$. Let $C(N_a^t)$ denote the cost of the path of greatest cost in S_a: $C(N_a^t) \triangleq \max_{x,y \in S_a} c(x,y)$. Choose the pair of nodes b and c by:

$$(b,c) = \arg \min_{(x,y) \mid [a,x] \in N_a^t, y \in M_a} C(N_a^t(\ominus[a,x], \oplus[a,y]))$$

and swap link $[a,b]$ for link $[a,c]$ if:

$$C(N_a^t(\ominus[a,b], \oplus[a,c])) < C(N_a^t) \times \varepsilon$$

where $\varepsilon < 1$ is a threshold constant [11].

4.2 Second Path Reduction Algorithm

In our second path reduction algorithm only link substitution method is used which makes it different from our previous PR algorithm in which link substitution is used

[6] The precise meaning of this cost function does not matter. It could be simply the number of hops, or some more complex measure involving load and/or interference.

```
For node $n_x \in N$ select one of its interfaces $i_x$
that is free and has the strongest signal of all
its free interfaces;
if $i_x = \varnothing$
        for $n_x$ set blockFree ← bc;
        end;
else
        set blockBusy ← bc;
set $I_{free}$ ← communicationRange($i_x$ , I) ;
if $I_{free} = \varnothing$
        endAction.
set $i_y$ ← bestSNR($I_{free}$)
if $i_y = \varnothing$
        endAction.
$l_{i_x i_y}$ ← createLink($i_x$ , $i_y$ )
for $n_x$ linkCounter  linkSubCounter + 1;
remove ($l_{i_u i_v}$ );
reset ($i_u$ , $i_v$);
set p ← newShortestPath($n_x$)
end.
```

N	a set of all nodes
I	a set of all interfaces
L	a set of all links
I_{free}	a subset of interfaces
i_x	an available interface.
n_x	a node which contains i_x.
i_y	an available interface.
n_y	a node which contains i_y.
i_u	an interface with shortest path on n_x
n_v	a node that contains $i_{v.}$
i_u	an interface on n_z that is linked to i_u
p	a shortest path for a n_x
$l_{i_u u_v}$	a link between i_u *and* i_v
bc	a blocking constant

```
endAction
        for $n_x$
                set blockBusy ← 0;
                set blockFree ← bc;
end.
```

Fig. 3. Second path reduction algorithm

along with link addition. Another distinction is that previously we selected only those substituted links that would not increase the IC whereas in the current algorithm we select links irrespective of their effect on the IC.

The outline of our algorithm is succinctly given here: An initiator node selects one of its available radio interfaces on the basis of strongest transmission power, the selected interface creates a list of available interfaces in its communication range (locality principle), from these interfaces those that have a path length to the portal node longer than the shortest path are filtered out. The remaining ones are short listed and an interface from these that offers the best SNIR is selected; the new link between the two interfaces is created and the previous shortest path link is switched off and its interfaces operational attributes are reset to their default values. This process occurs simultaneously across the multiagent system. A more formal description of the algorithm is given in Figure 3. The functions in the algorithm are:

- communicationRange(interface, set of interfaces): This function selects all interfaces from the set of interfaces that are free and within the range of interface.
- shortestPath(p, interface, set of interfaces): This function selects a set of interfaces that has a shortest path from the interface to the set of interfaces. If the path between the interface with a shortest path and the interface is not shorter than path p the function returns ∅. Otherwise it returns a set of interfaces with a shortest path.

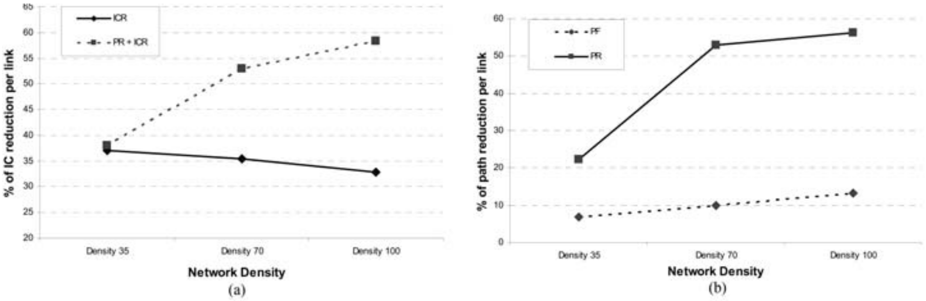


Fig. 4. (a) Result (comparative study) showing the increase in IC reduction in comparison to ICR only algorithm. (b) First versus the second algorithm for path reduction through link substitution.

- bestSNIR(set of interfaces): This function selects an interface with a best SNIR from the set of interfaces. If there is more than one such interface this function randomly selects one.
- createLink(interface *A*, interface *B*): This function creates a link between agent *A* and agent *B* and returns a newly created link.
- remove(link): This function removes a link from the set *L*.
- reset(interface *A*, interface *B*): This function resets attributes of agent *A* and agent *B* to default attributes.
- newShortestPath(node): This function returns a shortest path value for the node.
- A node is either *locked* or *unlocked*. A locked node is either locked because it has committed to lock itself for a period of time on request from another node, or it is ‘self-locked’ because it has recently instigated one of the self organisation procedures. A locked node is only locked for a “very short” period. This is simply to ensure that no more than one alteration is made during any one period which is necessary to ensure the stability of the procedures.

4.3 Results and Discussion

We present below some of the key results that we have obtained to illustrate the performance of the second path reduction algorithm. Figure 4(a) shows a graph of ICR vs. network density when just the algorithm for ICR has been invoked and both the path reduction and ICR algorithms has been invoked. This comparative study clearly shows that without the invocation of the path reduction algorithm the effect of ICR process results in much higher IC. A reason for this is twofold. Firstly, because of an increased number of interfaces that the initiator node can use and secondly, because of the additional mechanism that selects the interface based on the best SNIR value. Furthermore, as the network density increases the performance of the path reduction followed by ICR significantly increases whereas just the performance of the ICR algorithm on its own slightly decreases. Figure 4(b) compares the path length reduction that is achieved by using the first algorithm and the second path length reduction algorithm. The second algorithm is much more effective than the first by a factor of 2 to 5 times.

5 Conclusion and Future Work

We have described a multiagent system based self-organising algorithm for multi-radio wireless mesh networks (MR-WMN) that can operate on any radio technology. The multiagent system ensures scalability by progressively assigning the channels to nodes in clusters during the WMN system start up phase. The stability is offered by means of the proactive and reactive logic of the system. These attributes were validated through analysis and simulation. Through the work described in this report we have examined motivation and developed a multiagent system for the topological control of MR-WMN. The goal of this algorithm is to increase the number of shortest paths to the portal nodes without adversely effecting interference cost. In addition to interference cost reduction implementation of this algorithm on MR-WMN further improves the system capacity.

Our future work will be focused on the development of our Java framework that is multi threaded so each node is represented as an independent thread. We believe that this will enable us to develop algorithms for tuning the capacity of the network links according to fluctuations in demand by mobile users.

References

1. Raniwala, A., Chiueh, T.c.: Architecture and Algorithms for an IEEE 802.11-based Multi-channel Wireless Mesh Network. In: Proceedings IEEE Infocom 2005. IEEE Computer Society, Los Alamitos (2005)
2. Ko, B.J., Misra, V., Padhye, J., Rubenstein, D.: Distributed Channel Assignment in Multi-Radio 802.11 Mesh Networks. Technical report, Columbia University (2006)
3. Mishra, A., Rozner, E., Banerjee, S., Arbaugh, W.: Exploiting partially overlapping channels in wireless networks: Turning a peril into an advantage. In: ACM/USENIX Internet Measurement Conference (2005)
4. Mishra, A., Shrivastava, V., Banerjee, S.: Partially Overlapped Channels Not Considered Harmful. In: SIGMetrics/Performance (2006)
5. Akyildiz, I.F., Wang, X., Wang, W.: Wireless mesh netowrks: a survey. Computer Networks, 445–487 (2005)
6. Vasudevan, S.: A Simulator for analyzing the throughput of IEEE 802.11b Wireless LAN Systems. Master's thesis, Virginia Polytechnic Institute and State University (2005)
7. Leith, D., Clifford, P.: A self-managed distributed channel selection algorithm for wlans. In: Proceedings of RAWNET, Boston, MA, USA, pp. 1–9 (2006)
8. Duffy, K., Malone, D., Leith, D.: Modeling the 802.11 Distributed Coordination Function in Non-saturated Conditions. IEEE Communication Letters 9, 715–717 (2005)
9. Tourrilhes, J.: Robust Broadcast: Improving the reliability of broadcast transmissions on CSMA/CA. In: Proceedings of PIMRC 1998, pp. 1111–1115 (1998)
10. Fischer, K., Schillo, M., Siekmann, J.: Holonic multiagent systems: The foundation for the organization of multiagent systems. In: Mařík, V., McFarlane, D.C., Valckenaers, P. (eds.) HoloMAS 2003. LNCS (LNAI), vol. 2744, pp. 71–80. Springer, Heidelberg (2003)
11. Ramachandran, K., Belding, E., Almeroth, K., Buddhikot, M.: Interference-aware channel assignment in multi-radio wireless mesh networks. In: Proceedings of Infocom 2006, Barcelona, Spain, pp. 1–12 (2006)

Anomaly Detection in Noisy and Irregular Time Series: The "Turbodiesel Charging Pressure" Case Study*

Anahì Balbi[1], Michael Provost[2], and Armando Tacchella[3]

[1] Centro Biotecnologie Avanzate - Largo R. Benzi 10, 16132 - Genova, Italy
anahi.patrizia.balbi@unige.it
[2] Bombardier Transportation UK - Litchurch Lane, Derby, UK
mike.provost@uk.transport.bombardier.com
[3] Università degli Studi di Genova, DIST - Viale F. Causa 13, 16145 - Genova, Italy
armando.tacchella@unige.it

Abstract. In this paper we consider the problem of detecting anomalies in sample series obtained from critical train subsystems. Our study is the analysis of charging pressure in turbodiesel engines powering a fleet of passenger trains. We describe an automated methodology for (i) labelling time series samples as normal, abnormal or noisy, (ii) training supervised classifiers with labeled historical data, and (iii) combining classifiers to filter new data. We provide experimental evidence that our methodology yields error rates comparable to those of an equivalent manual process.

1 Introduction

Our work is meaningful in the context of data-driven prognostics (DDP), a process whereby data from in-service equipment is acquired with the final goal to predict when maintenance should be performed, see, e.g., [1]. In this paper, we are concerned with the analysis of time series obtained by monitoring relevant features in a fleet of passenger trains. These are diesel multiple units wherein each car has it own turbodiesel engine, but cars are capable of working in multiple as a single unit. We focus on turbocharger exit pressure time series, i.e., samples corresponding to the air pressure value measured in proximity of the engine air intake – hence the name "Turbodiesel Charging Pressure" given to our case study. We chose charging pressure since it is a relevant performance parameter in a turbodiesel engine, one whose anomalous behaviors may indicate several possible faults, including, e.g., turbocharger malfunction, or leaking ducts. The fact that cars work in multiple aggravates the potential problem, because a fault occurring in a single car may go unnoticed until it either causes significant damage, or another failure occurs that renders the whole unit inoperative.

More to the point, the fleet-wide analysis of charging pressure time series is made nontrivial by two facts. The first is that charging pressure is sampled by legacy diagnostic units, and thus its value is not monitored continuously, but only when specific

* At the time of this work, A. Balbi was partially supported by a PhD grant from Bombardier Transportation Italy. A. Tacchella is partially supported by a research grant from Bombardier Transportation Italy. Part of this work was carried out while A. Balbi was visiting the Predictive Services Engineering Group of Bombardier Transportation UK Ltd.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 123–132, 2010.

diagnostic events occur – gear shifting in our case. This means that the corresponding time series are irregular, i.e., the sampling interval is not constant, not predictable in advance and also widely spaced in time. Indeed, it is not uncommon to have gaps of *minutes* between consecutive samples. The second is that the engine idles during gear shifting which may cause a sudden drop in charging pressure. The corresponding time series contain samples which deviate from the standard value but do not imply a fault condition. However, unlike noise due to the measurement process, this kind of noise cannot be interpreted using standard models, e.g., true signal plus random disturbances. Noisy samples, in our case, must be identified and filtered away in order to enable the correct interpretation of anomalies in the corresponding time series.

Our problem is thus to detect anomalous behaviors in time series with the above characteristics using an automated method which is (i) simple, (ii) reliable, and (iii) efficient at the same time. Simplicity calls for approaches which are easy to implement and whose internals can be understood, even by personnel without a specific background in Computer Science or Artificial Intelligence, such as design and service engineers. Reliability is required to build trust in the method which requires very low error probabilities. Finally, identification of anomalies must be efficient, because it has to be performed on-line, while data for several different engines are collected.

The rest of the paper is structured as follows. In Section 2 we introduce the case study and we describe how we preprocessed and organized the data. In Section 3 we introduce automated labeling of time series. In Section 4 we describe the on-line classification part of our system and we present a quantitative performance evaluation of the overall system. We conclude the paper in section 5 with a discussion about our work.

2 The Case Study

Our case study focuses on charging pressure time series obtained from diagnostic data of turbodiesel engines mounted on a fleet of passenger trains. Each unit in the fleet consists of three cars, and each car is fitted with an engine which works in multiple with the other ones. An example of a time series corresponding to charging pressure samples downloaded from a specific car of the fleet is plotted in Figure 1. The y-axis of the plot is the charging pressure value, and the x-axis corresponds to the sequential sample number. Because of irregular sampling, equally spaced samples may not correspond to the same time interval, the only relationship with time being that the $(i + 1)$-th sample always occurs after the i-th one.

As we can see from the plot in Figure 1, the samples in charging pressure time series can be roughly distinguished in *normal*, *noisy* and *abnormal*. In particular, normal samples are those fluctuating around a stationary mean towards the top of the plot, i.e., the standard level of the feature. Noisy samples are scattered points occurring in isolation below the normal level. Indeed, they represent a physiological condition – pressure occasionally drops because the engine is idling during gear shifting – but their value can be several standard deviations apart from the mean normal level. Finally, abnormal samples are those lumping together below the normal level, and they may indicate the presence of an incipient or actual fault. While the standard approach in system monitoring is fixing thresholds and generating alarms when sample values are outside the

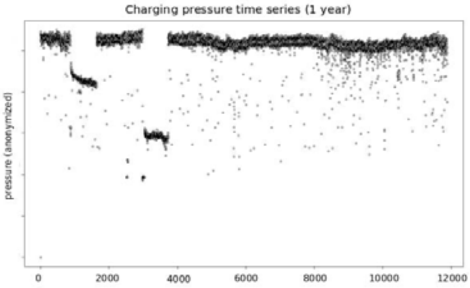


Fig. 1. Example of charging pressure time series (one year of samples from one car). The x-axis is sample sequence and the y-axis is charging pressure in mbar. Actual values on the y-axis have been anonimized.

allowed range [2], applying this method to our case could result in noisy samples to be mistakenly identified as abnormal. Moreover, since sampling is heavily irregular, we do not have an associated frequency domain in which noise could be filtered out, e.g., by removing the high frequency components of the time series.

We collected charging pressure data from a fleet of 16 units – 48 cars – run during 2007. The median number of samples in the resulting time series – one for each car – is 7591, while the maximum is 11899. We further divided each time series in 6 datasets for ease of management, yielding a maximum of 2000 samples for each dataset. We normalized the charging pressure values in order to make them comparable across the fleet. Indeed, minimum and maximum values from pressure sensors can be different for each car, so we normalize all pressure samples between 0 and 1 in a non-parametric way. We discarded datasets in which samples are either affected by heavy noise, or in which there are less than one thousand samples. The former datasets could result from a pressure sensor not working properly, while the latter do not bring much information and could possibly have a negative impact on the performances of our system. In the end, we consider 270 out of 288 datasets with an average number of about 1200 samples each. All the datasets are affected by noise, and 106 datasets are also characterized by one or more anomalies, i.e., lumps of abnormal samples.

3 Off-Line Labeling

Labeling samples as normal, noisy or abnormal reduces to a problem of unsupervised pattern classification, see, e.g., [3]. While several off-the-shelf algorithms exist for finding groups in data, see, e.g., [4], to the extent of our knowledge there is no such algorithm that could group noisy samples in our case. This is because clustering algorithms aim to find compact clusters of samples, whereas we experience noise scattered throughout the time series. In order to cluster the data we proceed as follows. We consider charging pressure – the original feature – as well as an additional feature that we call “neighbor distance”. Neighbor distance is computed as the average distance among a sample point and K points preceding it and K points following it, where K

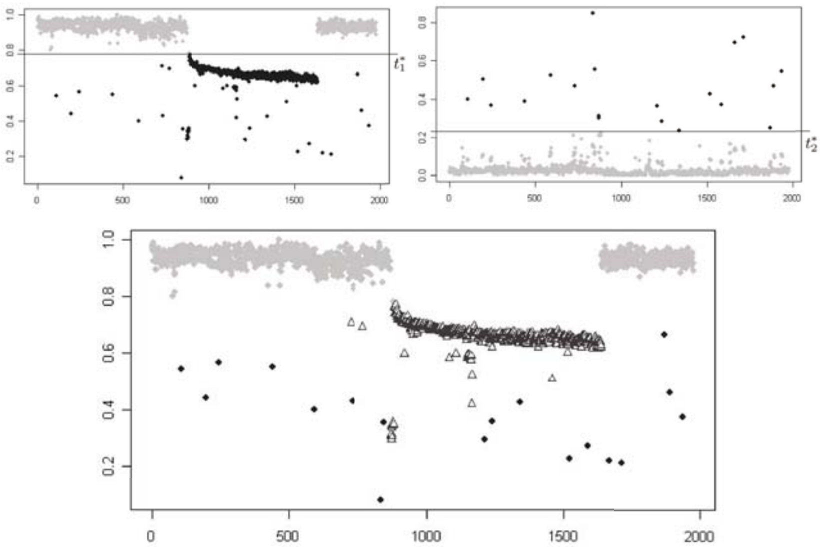


Fig. 2. Automated labeling by adaptive threshold computation: charging pressure (top-left), neighbor distance (top-right), and combined classification (bottom). Each plot features sample sequence on the x-axis and normalized charging pressure on the y-axis.

is an integer parameter such that $K \geq 1$. More precisely, if $v = \{v_1, v_2, \dots v_n\}$ is the vector of samples corresponding to charging pressure, neighbor distance is a vector $d = \{d_1, d_2 \dots, d_n\}$ where each d_i is computed according to

$$d_i = \begin{cases} \frac{1}{2K} \sum_{j=i-K}^{i+K} |v_i - v_j| \text{ if } K < i \leq n - K \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{1}$$

Intuitively, if the average distance between each of the K samples preceding v_i and each of the K samples following v_i is relatively large, then there is more chance for v_i to be scatter noise. On the other hand, consistent behavior of any kind - be it normal or abnormal - will result in a relatively small value of d_i. Notice that the value of d_i is 0 for the indices corresponding to the first and the last K samples of a time series. In an off-line setting, this is not problematic since we always choose $K \ll n$, and we can simply discard the first and the last K samples in v after computing d, so v and d will have the same number of “significant” entries. In an on-line setting, leveraging neighbor distance implies that we are not able to perform any classification unless we have seen at least K samples. Again, this is not a major issue, as long as $K \ll n$. In particular, for all the experiments that we show in this paper we fixed $K = 3$.

As we can see in Figure 2 (top-left), in the space of charging pressure it is relatively easy to separate normal samples (grey dots) from abnormal and noisy samples (black dots): a threshold value t_1^* (plotted as a straight line) is sufficient to accomplish the task. On the other hand, in Figure 2 (top-right), we can see that in the space of neighbor distance a second threshold value t_2^* can be computed to separate noisy samples (black dots - large values of neighbor distance) from normal and abnormal samples (grey dots - small values of neighbor distance). Now we have a way to discriminate between normal, noisy and abnormal points, by observing that noisy samples are always below t_1^* and above t_2^*, whereas abnormal samples are also below t_1^*, but they are below t_2^* as well. All the remaining samples can be classified as normal. The plot in Figure 2 (bottom) shows the final labeling obtained by combining the results of the partial classifications shown in Figure 2 (top). In the plot, grey dots stand for normal samples, black dots for noisy samples and triangle-shaped dots stand for abnormal samples. This convention is used throughout the rest of the paper. From a qualitative point of view, we can see that the combined classification yields the expected result. Indeed, some noisy samples are labeled as abnormal, but this happens only when the anomaly is occurring, which is reasonable.

We compute t_1^* and t_2^* by optimizing a coefficient of clustering quality. In particular, we consider the silhouette index [4] which is a numerical index in the range $[-1, 1]$ computed for each sample by considering the distance between the sample and the elements in its own cluster (intraclass distance), and comparing it with the distance between the elements in different cluster (interclass distance). Intuitively, if the intraclass distance is small with respect to the interclass distance, the sample is correctly classified, and silhouette index approaches 1. Otherwise, if the interclass distance is small with respect to the intraclass distance, the sample is misclassified, and silhouette index approaches -1. Silhouette index values close to 0 indicate borderline classifications. The average silhouette index over all the samples in a dataset is a well-established measure of clustering quality, see, e.g. [4]. The specific setting of t_1^* and t_2^* is adapted to each of the 270 datasets that we consider as follows. Let v be the sample vector, t be the threshold value and $\sigma(t)$ be the average silhouette index obtained considering two clusters: $A = \{v_i \mid v_i \geq t\}$, i.e., the samples above the threshold, and $B = \{v_i \mid v_i < t\}$, i.e., the samples below the threshold. We start by setting $t = \overline{v}$, i.e., we initialize the threshold to the mean value of the samples. Then we optimize, e.g., by iterative improvement, $\sigma(t)$. The optimal threshold is thus $t^* = argmax_t\{\sigma(t)\}$. In Figure 2 (top, middle) the values of t_1^* and t_2^* are obtained by maximizing $\sigma(t_1)$ and $\sigma(t_2)$, respectively. As we can see, computing an adaptive threshold using the average silhouette index corresponds to the classification that an human expert could perform by looking at the data.

4 On-Line Filtering

Once datasets are labeled, we can use them as training sets to yield pattern classifiers which analyze new data on-line, i.e., sample by sample. We start by labelling all the available datasets using the algorithm shown in Section 3, and then we implement the following steps:

1. A number q of datasets is extracted, uniform at random with replacement, from labeled datasets.
2. A supervised classifier M is trained on the datasets considering both charging pressure and neighbor distance.

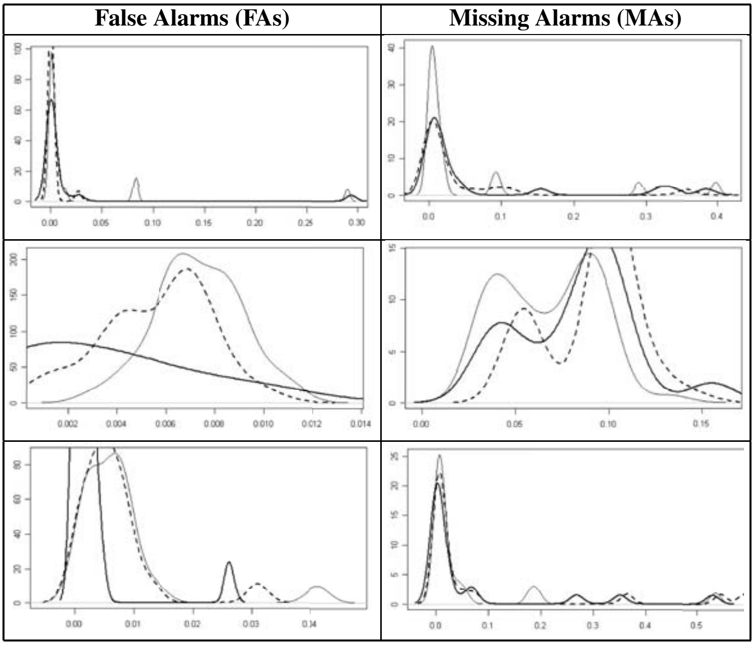
For each learning algorithm, steps (1) and (2) are iterated r times, so that we obtain $M = \{M_1, M_2, \ldots M_r\}$ classifiers. On-line data can be filtered by combining the result of the classifiers using simple majority voting. The motivation behind the use of voting is that automated labeling can introduce *labeling noise*, i.e., samples of one kind mistakenly identified as samples of another one. Therefore, random selection of training datasets is usually not sufficient to make the final classifier robust enough for our purposes, and we need to combine the results of several classifiers to make them fit. In all the following experiments, we set $q = 20$, i.e., less than 10% of the datasets available to us, and $r = 50$. For the sake of comparison, the model M is computed according to three well-established supervised learning algorithms, namely k Nearest Neighbor (k-NN) [5], Support Vector Machines (SVMs) [6], and Classification Trees (CARTs) [7]. The reason of our choice is twofold. First, the three algorithms are paradigmatic of different classes of pattern classifiers. k-NN is the reference instance-based learning algorithm. SVMs and CARTs, on the other hand, build a model which goes beyond storing training instances and classifying by lookup, but they are paradigmatic of functional and symbolic models, respectively. The second motivation, that holds for all the chosen algorithms, is that they are non-parametric, i.e., knowing the specific shape of the statistical relationship between input and output is not required.

In order to evaluate the performances of our methodology, we consider a subset of 20 automatically labeled time series, and we manually edit them to correct any misclassification error. This task, which would be too time consuming to apply routinely, gives us a set of *reference time series* that we can use for validation purposes. In particular, we are interested in bounding the effect of labeling noise on false and missing alarm rates. To this purpose we consider reference time series as the input of two distinct processes:

- *Cross validation on reference time series*. Reference time series are leveraged to obtain an estimate of the generalization error of supervised learning algorithms This is done by using a standard "leave one out" cross-validation scheme (see, e.g., [3]), wherein all but one reference time series are used for training a classifier, and the remaining one is used to estimate the error.
- *Validation of voted classification*. Reference time series are given as input to the voted classifier obtained by combing the results of several classifiers trained on random selections of automatically labeled samples. This yields an estimate of the total error of the automated approach. Again, for each reference time series we obtain an error estimate, and we consider the distribution over 20 values.

In the plots of Figure 3 we compare, from a qualitative point of view, the results obtained on reference time series with those that can be obtained by the automated approach. The plots in Figure 3 are non-parametric density estimates using gaussian kernels (see, e.g., [3]) of false alarm (left column) and missing alarm (right column) error rates obtained considering three cases:

1. Cross-validation on reference time series – Figure 3 (top); this represents the best performance we can expect to obtain using on-line supervised pattern classification;
2. A *single* classifier trained on a subset of automatically labeled time series and tested on the whole set of reference time series – Figure 3 (middle); since we trained $r = 50$ different classifiers, and each classifier exhibits a different performance


False Alarms (FAs)

Missing Alarms (MAs)

	FAs (median)			MAs (median)		
Setting	k-NN	SMVs	CARTs	k-NN	SVMs	CARTs
Reference	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	0.01	0.01	0.01
Single	0.01	0.01	0.01	0.07	0.09	0.09
Voted	0.01	0.01	0.01	0.01	0.01	0.01

	FAs (IQ-range)			MAs (IQ-range)		
Setting	k-NN	SMVs	CARTs	k-NN	SVMs	CARTs
Reference	0.01	$< 10^{-3}$	$< 10^{-3}$	0.01	0.03	0.03
Single	0.01	0.01	0.01	0.05	0.03	0.04
Voted	0.01	0.01	0.01	0.02	0.03	0.03

Fig. 3. Distribution of false and missing alarm rates (left and right columns respectively) in three different settings: cross-validation of reference time series (top), single random training set and reference test set (middle), voted classification (bottom). The thin line corresponds to k-NN. the dotted line to SVMs and the thick line to CARTs. The table at the bottom reports median and interquartile values of each distribution.

depending on the specific training set, the resulting distribution is a reasonable estimate of the effects of labeling noise on the performances of a pattern classifier.

3. The voted classifier – Figure 3 (bottom);

We included case (2) above in order to highlight the effect of combining several pattern classifiers using a voting schema. From the plots in Figure 3 (top) we can see that, for all the supervised learning algorithms that we consider, training on reference time series yields very good performances. This is not an obvious result, and it proves that

Table 1. Comparing error distributions: p-values of the WRS tests

	False alarms (p-value)		Missing alarms (p-value)	
Algorithm	Reference vs. Single	Reference vs. Voted	Reference Vs Single	Reference Vs Voted
k-NN	0.002	0.03	$< 10^{-3}$	0.3
SVMs	$< 10^{-3}$	0.03	$< 10^{-3}$	0.2
CARTs	0.001	0.02	$< 10^{-3}$	0.4

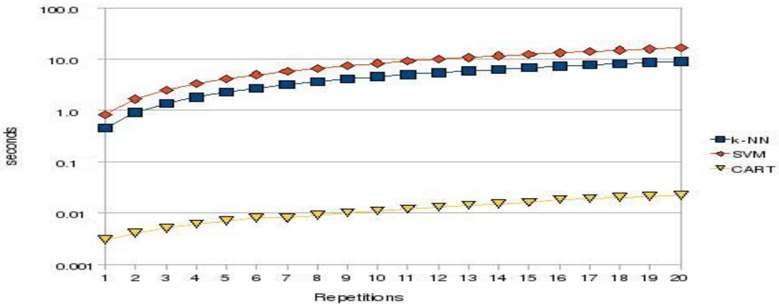


Fig. 4. Performances of time series classification with k-NN (boxes), SVMs (diamonds) and CARTs (triangles). The x-axis is the number of repetitions of the testing dataset and the y-axis is CPU time in seconds (logarithmic scale).

our choice of features – charging pressure and neighbor distance – is adequate for the problem at hand, and that it does not consistently privilege one classifier over another. In Figure 3 (middle) we can see that, on the other hand, training on datasets affected by labeling noise can degrade the performances of the supervised algorithms. Not only mean error rates tend to be higher than training on reference time series, but also the variance is substantially increased. Finally, from the plots in Figure 3 we see that performances can be boosted by combining several randomly trained classifiers using a voting schema.

We compared the location shift of various error distributions to see whether there are significant statistical differences from setting to setting and from algorithm to algorithm. To this purpose we considered the Wilcoxon Rank Sum (WRS) test [8], a non-parametric test of location shift which compares two distributions against the null hypothesis that the shift is zero. In Table 1 we show the p-values of the test, i.e., the probability to observe the same difference as the one observed in the data if the null hypothesis were true. We set a confidence level of $\alpha = 0.01$ (99%), and thus we reject the null hypothesis whenever the p-value is less than α. In Table 1, we can see that the null hypothesis can be rejected with a 99% confidence in all the cases of "Reference vs. Single" setting. On the other hand, looking at the column "Reference vs. Voted", we can see that the null hypothesis *cannot* be rejected, i.e., the location of the distributions is statistically the same in all the cases.

In our experiments we have also tracked efficiency of the the three main phases: (i) automatic labeling as described in Section 3, (ii) training of supervised classifiers, and (iii) time series filtering. [1] Considering phase (i), we report a total CPU time of 570s (resp. 816s) to optimize thresholds on charging pressure (resp. neighbor distance) for 339251 samples in the datasets that we examined. Overall, this means that automated labeling costs less than one hundredth CPU second per sample, while labeling a single dataset (between one and two thousands samples) costs less than 5 seconds on average. As for phase (ii), we report data about the time taken to classify 20 reference time series. The whole cross validation process, costs 10s, 1468s and 23s for k-NN, SVMs and CARTs, respectively. Here we can see that SVMs are substantially less efficient than k-NN and CARTs, and this is particularly true of the training phase. Indeed, most of the CPU time reported above is related to the testing phase for k-NN and CARTs, while it is related to the training phase for SVMs. Training on 50 randomly assembled ensembles of datasets, as it is required in order to provide input to the voted classification, involves a substantial amount of time for SVMs (about 20 hours), but less than 20 minutes for k-NN and CARTs. If off-line training time comes at a premium, we can safely opt for the latter algorithms, as they guarantee accuracies that are comparable to, or even better than, SVMs. Finally, for phase (iii), we consider the following experiment, whose result is presented in Figure 4. We train the algorithms on 19-out-of-20 reference time series. Then we construct a test set by collating r repetitions of the reference time series that was left out for testing. In this way, as the number of repetitions grows, we get longer and longer time series. The purpose of this experiment is to see how the classification algorithms scale with the size of incoming data. In Figure 4 we show the results of k-NN, SVMs and CARTs with $r = 1, 2, \ldots, 20$, where $r = 20$ roughly corresponds to two years worth of samples (notice the logarithmic scale on the y-axis of the plot). As we can see from the plot, the classification time scales gracefully with the size of the time series for all the algorithms considered. Indeed, classifying time series with more than 10^3 samples can be always done in less than 20 seconds even by the least efficient algorithm, namely SVMs.

5 Conclusions and Related Work

Our initial goal was to devise a methodology that can detect anomalies in noisy and irregular time series which is simple, reliable and efficient at the same time. We argue that the work presented in this paper fulfills the initial goals. As we have shown with extensive analysis on the "Turbodiesel Charging Pressure" case study, our methodology yields error rates that are very close to those that could be obtained by labeling the time series manually. In particular, we obtain very low error probabilities (in the order of 10^{-3}) for false alarms which are always expensive and tend to create distrust in the system (see, e.g., [1]). Our method is efficient since even a far-from-optimized

[1] All the CPU times that we consider in the following have been obtained using a proof-of-concept implementation of our methodology using the R environment [9]. We benchmarked the off-line part on a 1.86Ghz Pentium IV laptop with 1GB of RAM running Windows XP, and the on-line part on a 2.3GHz Pentium IV workstation with 3GB of RAM running Linux.

implementation in a high-level interpreted language such as the one provided by R environment, can classify a fairly large amount of data in a matter of seconds. Moreover, since we use relatively small chunks of time series for training, the most expensive part of the process is independent from the scale of the deployment. On the other hand, classification is extremely efficient as it scales linearly with the size of the dataset.

There is some literature about nonuniformly sampled time series like the ones we deal with. In particular, in [10] a statistical approach to the problem is sought, which provides a spectral analysis method based on least square approximation. Albeit sophisticated, this method cannot provide an automatic filtering procedure, in particular when sampling is affected by constant offsets. A number of contributions – see, e.g., [11] for a survey – tackle the problem of irregular sampling with interpolation. These solutions, in our case, can lead to misleading results because there is no "physical law" binding the values between two samples. More recently, in [12], several approaches that parallel the uniformly-sampled cases are introduced. Some of them could be interesting also in our case and we are currently performing an extensive comparison about these methods and ours. Our future work includes also testing the approach with different features, and combining the results obtained in a multi-feature setting to improve the accuracy of classification.

References

1. Vachtsevanos, G., Lewis, F.L., Roemer, M., Hess, A., Wu, B.: Intelligent Fault Diagnosis and Prognosis for Engineering Systems. Wiley, Chichester (2006)
2. Isermann, R.: Fault Diagnosis Systems. Springer, Heidelberg (2006)
3. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, Chichester (2000)
4. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data. Wiley, Chichester (1990)
5. Aha, D., Kibler, D.: Instance-based learning algorithms. Machine Learning, 37–66 (1991)
6. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
7. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and regression trees. Wadsworth Inc. (1984)
8. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics 1 (1945)
9. R Development Core Team. R: A Language and Environment for Statistical Computing, Version 2.6.2 (2008-02-08), http://cran.r-project.org/
10. Mathias, A., Grond, F., Guardans, R., Seese, D., Canela, M., Diebner, H.H.: Algorithms for spectral analysis of irregularly sampled time series. Journal of Statistical Software 11(2), 1–30 (2004)
11. Adorf, H.M.: Interpolation of Irregularly Sampled Data Series—A Survey. In: Astronomical Data Analysis Software and Systems IV, vol. 77, p. 460 (1995)
12. Stoica, P., Sandgren, N.: Spectral analysis of irregularly-sampled data: Paralleling the regularly-sampled data approaches. Digit. Signal Process. 16(6), 712–734 (2006)

Safe Learning with Real-Time Constraints: A Case Study*

Giorgio Metta[1,2], Lorenzo Natale[2], Shashank Pathak[1],
Luca Pulina[1,**], and Armando Tacchella[1]

[1] DIST, Università di Genova, Viale Causa, 13 – 16145 Genova, Italy
[2] Italian Institute of Technology, Via Morego 30 – 16163 Genova, Italy

Abstract. Aim of this work is to study the problem of ensuring safety and effectiveness of a multi-agent robot control system with real-time constraints in the case of learning components usage. Our case study focuses on a robot playing the air hockey game against a human opponent, where the robot has to learn how to minimize opponent's goals. This case study is paradigmatic since the robot must act in real-time, but, at the same time, it must learn and guarantee that the control system is safe throughout the process. We propose a solution using automata-theoretic formalisms and associated verification tools, showing experimentally that our approach can yield safety without heavily compromising effectiveness.

1 Introduction

A key feature of state of the art robotic development architectures is real-time multi-agent control systems [1] support. Such architectures are of paramount importance in robotics, because they offer a scalable solution for a number of operations, ranging from the computation of inputs coming from several sensors, to the execution of complex cognitive functions. These systems are complex, and such complexity is motivated from, for instance, the parallel execution of many interacting components, or real-time constraints to satisfy during the execution of a given task. This complexity is increased by learning agents which enable robots to perform various tasks efficiently [2] on one hand, but which may conceal errors which can hinder the correctness of the control system on the other. This is mainly due to the fact that learning agents change their internal parameters over time, making it difficult, if not impossible, to anticipate all the possible behaviors of the control system as a whole.

Aim of this paper is to prove the correctness of a real-time robot control system embedding learning agents with respect to safety properties, i.e., properties requiring that a control configuration that may cause damage to the environment is never reached. Our notion of safety is the one used in Formal Methods, i.e., given mathematical models of the system and some desired properties that it must satisfy, such methods return formal

[*] This research has received funding from the European Community's Information and Communication Technologies Seventh Framework Programme [FP7/2007-2013] under grant agreement n. [215805], the CHRIS project.

[**] Corresponding author. Email: luca.pulina@unige.it

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 133–142, 2010.

proofs that given properties hold during the evolution of the system. Such approach is known as Model Checking [3], i.e., an automated approach wherein the model of the system and the properties are input to a verification software which statically checks execution traces. Model Checking literature reports success stories concerning the debugging of integrated circuits [4], software [5] and, more recently, complex control systems [6].

The main contribution of this work is to formally prove safety properties of a multi-agent control system that includes learning components in a challenging setup with hard real-time constraints. Our case study involves learning to play the air hockey game. Air hockey is a game played by two players. They use round paddles (mallets) to hit a flat round puck across a table. At each end of the table there is a goal area. The objective of the game is to hit the puck so that it goes into the opponent's goal area (offense play), and to prevent it from going into your own goal area (defense play). Air hockey has already been explored as a benchmark task for humanoid robots, see, e.g., [7]. In the words of [7], air hockey is challenging because it is fast, demanding, and complex, once the various elements of the physical setup are taken into account. In our case study, a multi-agent control system manages such a robotic setup. A *vision* agent is devoted to visual perception; a *motion control* agent sends position commands to the manipulator; finally, a *coordination* agent converts the stimuli perceived by the vision agent into commands for motion control, and its internal parameters change over time for the effects of learning. In particular, the goal of learning is to fine tune coordination in order to be able to intercept the puck when it approaches the robot's goal area (defense play). In order to have an effective accomplishment of this task, we need a fast, real-time coordination.

Concerning safety, our target is to prove that our real-time multi-agent control system with learning is not able to reach unsafe positions, e.g., moving too close to the table's edges. In order to apply the Model Checking technique to our setting, we design and model each agent as an automaton, and, given the peculiarity of our case study, we formalize that as a hybrid automaton [8]. We check execution traces for safety in a static way, by feeding the automata and the property statements to a model checker, which can find bugs by exploring traces of increasing length. In our experiment we used the hybrid model checker HYSAT [9]. We preserve safety at all times by keeping safe – and possibly ineffective – parameters of the coordination agent in place, until we have a more effective – and definitely safe – setting. While the proposed approach is expressive enough to allow for various setups to be analyzed, our experimental analysis in the air hockey setup shows that it can yield safety without heavily compromising on effectiveness. All other things being equal, the accuracy of a safety-conscious control system is very close – albeit inferior – to the one of a safety-oblivious control system.

The paper is structured as follows. In Section 2 we introduce to the formal language and the algorithms to verify safety. In Section 3 we describe in detail both the setup and the formal model of the robot control system. In Section 4 we comment the results of our experiments, while in Section 5 we conclude the paper discussing related work.

2 Hybrid Automata and Safety Verification

In order to model the real-time multi-agent control system related to our case study, we resort to the formalism of *Hybrid Automata* [8]. For our purposes, a hybrid automaton can be defined as a tuple $A = (X, V, flow, inv, init, E, jump)$ consisting of the following components. *Variables* are a finite ordered set $X = \{x_1, x_2, \dots x_n\}$ of real-valued variables, representing the continuous component of the system's state. *Control modes* are a finite set V, representing the discrete component of the system's state. *Flow conditions* are expressed with a labeling function $flow$ that assigns a condition to each control mode $v \in V$. The flow condition $flow(v)$ is a predicate over the variables in $X \cup \dot{X}$, where $\dot{X} = \{\dot{x}_1, \dot{x}_2, \dots, \dot{x}_n\}$. The dotted variable $\dot{x}_i$ for $1 \leq i \leq n$ refers to the first derivative of x_i with respect to time, i.e., $\dot{x}_i = \frac{dx}{dt}$. *Invariant conditions* determine the constraints of each control mode with the labeling function inv, and *initial conditions* are denoted with the function $init$. *Control switches* are a finite multiset $E \in V \times V$. Each control switch (v, v') is a directed edge between a source mode $v \in V$ and a target mode $v' \in V$. *Jump conditions* are expressed with a labeling function $jump$ that assigns a jump condition to each control switch $e \in E$. The jump condition $jump(e)$ is a predicate over the variables in $X \cup X'$, where $X' = \{x'_1, x'_2, \dots x'_n\}$. The unprimed symbol x_i, for $1 \leq i \leq n$, refers to the value of the variable x_i before the control switch, and the primed symbol x'_i refers to the value of x_i after the control switch. Thus, a jump condition relates the values of the variables before a control switch to the possible values after the control switch.

Intuitively, checking that a hybrid automaton is safe amounts to checking that no execution reaches an unwanted condition. In order to frame this concept precisely we define a *state* as a pair $(v, \mathbf{a})$ consisting of a control mode $v \in V$ and a vector $\mathbf{a} = (a_1, \dots, a_n)$ that represents a value $a_i \in \mathbb{R}$ for each variable $x_i \in X$. The state $(v, \mathbf{a})$ is *admissible* if the predicate $inv(v)$ is true when each variable x_i is replaced by the value a_i. The state $(v, \mathbf{a})$ is *initial* if the predicate $init(v)$ is true when each x_i is replaced by a_i. Consider a pair of admissible states $q = (v, \mathbf{a})$ and $q' = (v', \mathbf{a}')$. The pair (q, q') is a *jump* of A if there is a control switch $e \in E$ with source mode v and target mode v' such that the predicate $jump(e)$ is true when each variable x_i is replaced by the value a_i, and each primed variable x'_i is replaced by the value a'_i. The pair (q, q') is a *flow* of A if $v = v'$ and there is a non-negative real $\delta \in \mathbb{R}_{\geq 0}$ – the duration of the flow – and a differentiable function $\rho : [0, \delta] \rightarrow \mathbb{R}^n$ – the curve of the flow – such that (i) $\rho(0) = \mathbf{a}$ and $\rho(\delta) = \mathbf{a}'$; (ii) for all time instants $t \in (0, \delta)$ the state $(v, \rho(t))$ is admissible; and (iii) for all time instants $t \in (0, \delta)$, the predicate $flow(v)$ is true when each variable x_i is replaced by the i-th coordinate of the vector $\rho(t)$, and each $\dot{x}_i$ is replaced by the i-th coordinate of $\dot{\rho}(t)$ – where $\dot{\rho} = \frac{d\rho}{dt}$. In words, jumps define the behavior of the automaton when switching from one control mode to another, whereas flows describe the behavior of the automaton inside the control mode. With the concepts above, we can now define executions of the automaton as *trajectories*, i.e., finite sequences $q_0, q_1, \dots, q_k$ of admissible states q_j such that (i) the first state q_0 of the sequence is an initial state of A, and (ii) each pair $(q_j, q_j + 1)$ of consecutive states in the sequence is either a jump of A or a flow of A. A state of A is *reachable* if it is the last state of some trajectory. Let us assume that a safety requirement can be specified by defining the "unsafe" values and value combinations of the system variables. A state is thus safe if it does not entail such values, and the whole system is safe exactly when all reachable states are safe. Safety verification, therefore, amounts to computing the set of reachable states.

Given the expressiveness of the above formalism, it is no surprise that checking safety is, in its most general form, an undecidable problem in hybrid automata [8]. Even if the air hockey control system can be modeled in terms of a *linear hybrid automaton*, i.e., a hybrid automaton where the dynamics of the continuous variables are defined by linear differential inequalities, there is still no guarantee that the exploration of the set of reachable states terminates. The method is still of practical interest, however, because terminations can be enforced by considering the behavior of the system over a bounded interval of time. This technique, known as Bounded Model Checking – see, e.g., [9] – involves the exploration of increasingly long trajectories until either an unsafe state is reached, or resources (CPU time, memory) are exhausted. Technically, we no longer speak of verification, which is untenable in an infinite state space, but of falsification. Whenever an unsafe state is found, then we have a trajectory witnessing the bug. On the other hand, the unfruitful exploration of increasingly long trajectories is considered an empirical guarantee of safety.

3 Modeling

3.1 Robotic Setup and Software Agents

The robotic setup concerning our case study consists of a commercial air hockey table, that is approximately 90cm wide and 180cm long. For vision, we used a CCD DragonFly2 camera wired to a standard PC by means of an IEEE-1394 interface. For manipulation, we used a 6 d.o.f. Unimation PUMA 260 industrial robot mounted upside down on a metallic scaffolding to match human-like arm position and movement. Four standard PCs are used to provide elaboration. The whole robot control system is built on top of YARP (Yet Another Robotic Platform) [10], a middleware devoted to the integration of devices for complex robotic architectures. In our setup YARP provides seamless connection between the modules, and it enables us to distribute the components across different PCs connected to a LAN.

Concerning the software agents, a vision module is dedicated to track the puck on the table and send puck positions via YARP. However, the puck moves fast – average speeds of 800 pixels/s (about 2.5m/s) are typical – and real-time detection requires a tight compromise between speed and complexity. Our tracking algorithm is accurate enough for our purposes, and fast enough to acquire some (at least two) frames containing the puck inside a predefined “perception zone” of the table. As soon as the puck leaves such zone, the grabbed positions are sent to the coordination module. In our experiments, the perception zone starts at 1.1 m ($\rho\_begin\_perception$) from the robot’s world origin, and ends at 0.6 m ($\rho\_end\_perception$).

A motion control module is dedicated to the low-level control of the manipulator. For the sake of speed and simplicity, control is organized around five primitives. *Home* returns the arm to its reference position. *Left* and *right*, respectively, rotate left and right the waist joint of the manipulator, so that the paddle follows a circular trajectory on the table centered in the robot's world origin. Finally, *forward* and *backward*, respectively, coordinate shoulder, elbow and the wrist bend joints of the manipulator, so that the paddle follows a linear trajectory on the table, going either outward or inward w.r.t. the robot's world origin.

A coordination module is in charge of (i) reading the puck positions from vision, (ii) estimating reasonable target positions for the robot's paddle and (iii) send the estimated position to the motion controller. The estimated target position of the robot's paddle (ρ_{ee}, θ_{ee}) is the result of the following calculations:

$$\begin{aligned} \rho_{ee} &= p_1 + p_2\rho_1 + p_3\theta_1 + p_4\rho_2 + p_5\theta_2 \\ \theta_{ee} &= p_6 + p_7\rho_1 + p_8\theta_1 + p_9\rho_2 + p_{10}\theta_2 \end{aligned} \tag{1}$$

where (ρ_1, θ_1) and (ρ_2, θ_2) correspond to two puck positions detected by vision, and $\mathbf{p} = \{p_1, p_2, \ldots, p_{10}\}$ are parameters learned using the WEKA [11] implementation of the Sequential Minimal Optimization for Support Vector Machines regression algorithm (SMOREG) [12]. Notice that coordination requires at least two points from vision in order to predict (ρ_{ee}, θ_{ee}).

The parameters $\mathbf{p}$ can be learned in two ways:

- **Off-line** by collecting a number of shots and related target positions, and using them as a train set to obtain the value of $\mathbf{p}$ which is unchanged thereafter.
- **On-line** by starting with some bootstrap value $\mathbf{p}_0$ and then recording new shots/target positions while playing so that new and improved values can be computed.

In our case, on-line learning amounts to adding new shots/target positions to a data set, and train SMOREG from scratch.

3.2 Control Agents as Hybrid Automata

In order to check the safety of the control system, we model each agent as a hybrid automaton, and then we check the composition of the agent's automata. All the automata that we describe in the following are depicted in Figure 1 (see the legend for graphical conventions).

The vision automaton – Figure 1 (top-left) – has three control modes: WAIT_V, SEND_V, and RECEIVE_V. The real variables are ¡(ρ_{1v}, θ_{1v}) and (ρ_{2v}, θ_{2v}) denoting the two positions perceived along the puck trajectory by the camera. The Boolean flags *ack_v* and *ack_r* are used for communications with the other agents. Initially, the automaton is in the control mode WAIT_V until at least two points are perceived along the trajectory of a shot directed towards the robot's goal area. When the two points are collected, the control mode SEND_V is reached, and the automaton loops there while the *ack_v* signal is false. When the coordination automaton acknowledges receipt of the coordinates, *ack_v* becomes true, and the mode RECEIVE_V is entered. The automaton resets to WAIT_V as soon as the motion controller also receives its target position and *ack_r* becomes true.

The coordination automaton – Figure 1 (top-right) – has three control modes: WAIT_C, PREDICT_C, and SEND_C. The real variables are (ρ_{1c}, θ_{1c}) and (ρ_{2c}, θ_{2c}), corresponding

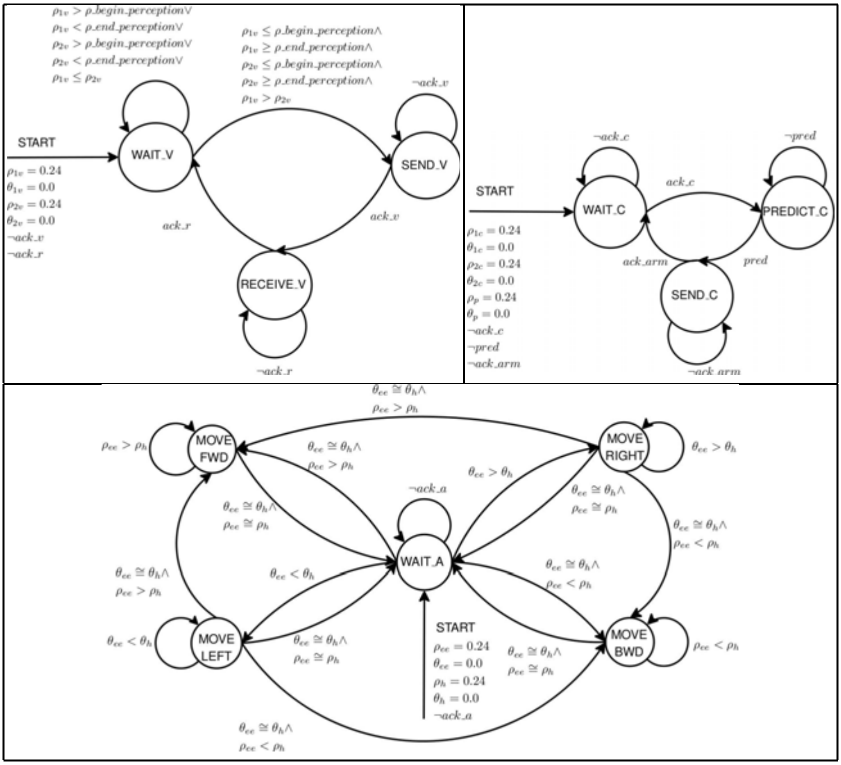


Fig. 1. Vision (top-left), coordination (top-right), and motion (bottom) automata. Control modes are vertexes, and directed edges are control switches whose labels represent jump conditions. Looping edges are labeled with invariant conditions. Initial conditions and initial values of the variables are shown as labels of incoming arrows (START). Symbols have the usual meaning, with the exception of "=" which indicates assignment, and "$\cong$" which indicates approximate equality testing: $x \cong y$ iff $x = y \pm \delta$, where δ is a small positive constant.

to the puck coordinates received by the vision automaton; the variables (ρ_p, θ_p) represent the result of applying equation (1) where $(\rho_1, \theta_1) \equiv (\rho_{1c}, \theta_{1c})$ and $(\rho_2, \theta_2) \equiv (\rho_{2c}, \theta_{2c})$. The Boolean flag $pred$ denotes the end of the computation, and the Boolean flags ack_c, and ack_arm are used for communication with other agents. Initially, the automaton is in the control mode WAIT_C until the puck coordinates are received from the vision automaton. In this case, the control mode PREDICT_C is reached, and the variables ρ_p, θ_p are computed according to equation (1). Once this operation is complete, there is a jump to the control mode SEND_C, in which the coordinates (ρ_p, θ_p) are sent to the motion controller automaton. The automaton loops in SEND_C until motion control acknowledges, and then it resets to WAIT_C. Notice that we do not model the learning component of coordination as an automaton, and we consider the parameters $\mathbf{p}$ as constants. However, we can keep into account changes in $\mathbf{p}$ by scheduling (re)verification of the control system automaton each time $\mathbf{p}$ changes because of learning. In this way, the internals of the learning algorithm need not to be modeled, but it is still possible to ensure safety of the control system, as long as only safe choices of the parameters $\mathbf{p}$ are used in the coordination module.

The motion controller automaton – (Figure 1, bottom) – has five control modes: WAIT_A, MOVE BWD, MOVE FWD, MOVE LEFT, and MOVE RIGHT, corresponding to the control primitives. The real variables (ρ_{ee}, θ_{ee}) are the target coordinates of the end effector; (ρ_h, θ_h) denote the current position of the end effector. The Boolean flag ack_a is devoted to communications with other agents. Initially, the automaton is in control mode WAIT_A until the coordinates (ρ_{ee}, θ_{ee}) are received from the coordination automaton. If ρ_h or θ_h are different from ρ_{ee} and θ_{ee}, respectively, a jump condition for the control modes representing the motion primitives is satisfied, and the automaton evolves until the target position is reached. Inside the primitives, the evolution of (ρ_h, θ_h) is modeled using linear differential equations expressing constant radial and angular velocities control, respectively.

4 Empirical Analysis

In this section we show empirical results supporting the claim that our multi agent control system can learn from experience while maintaining both safety and the real-time constraints imposed by the dynamic of the air hockey game. Our safety target requires that the angle θ_{ee} is in the range $[-50, +50]$ (degrees) and ρ_{ee} is the range $[0.10, 0.27]$ (meters). These boundaries ensure that the paddle will never hit the borders of the table. Considering that the robot moves, e.g., left and right, at an average speed of 5.2 rad/s (298 degrees/s) hitting the borders would surely damage the table or the paddle. In our experiments this is prevented by a low-level emergency protection that avoids dangerous commands to be executed by the robot. In the following, when we say that the control system reached an unsafe state during experimental plays, it means that the low-level protection intervened to avoid damage. In order to check for safety we used the tool HySAT, a satisfiability checker for Boolean combinations of arithmetic constraints over real- and integer-valued variables which can also be used as a bounded model checker for hybrid systems. HySAT takes as input a textual description of the automata in Figure 1, and it tries to find a violation of the stated safety properties by exploring trajectories of increasing bounded length k.

Data was collected by having the robot play a series of games against ten different human players and using three different settings of the coordination module. In the first setting (off-line), the parameters $\mathbf{p}$ are learned off-line using a controlled training set of 150 shots performed by one of us, of which 50 are straight shots, and 100 are single-bounce shots; this setting is not checked for safety. In the second setting (on-line), the parameters $\mathbf{p}$ are learned on-line; the bootstrap parameters $\mathbf{p}_0$ correspond to a hand-made setting which is checked off-line for safety. In the third setting (safe on-line), each time a new set of parameters is learned, it is plugged into the model which is then checked for safety; the new parameters are used thereafter only if HySAT could not find a safety violation within 30 CPU seconds. Notice that in the two on-line settings we keep learning across different players, so the more games are played, the more effective the robot becomes. Because of this, the training set accumulates the shots/target positions recorded during all the playing history. The choice of 30 CPU

Table 1. Unsafe predictions during the experiments. The table is organized as follows: The first column contains the name of the two settings (OFF-LINE and ON-LINE). It is followed by ten columns, one for each player. Each cell contains two values: the leftmost one denotes the number of shots detected for each player, while the rightmost one (inside the parenthesis) denotes the number of shots that triggered the low level protection. A "–" indicates that no shots were detected.

	# 1	# 2	# 3	# 4	# 5	# 6	# 7	# 8	# 9	# 10
OFF-LINE	103 (3)	46 (–)	58 (1)	59 (–)	56 (–)	48 (–)	61 (–)	99 (–)	84 (1)	44 (–)
ON-LINE	112 (–)	39 (1)	80 (–)	55 (–)	72 (–)	69 (–)	46 (–)	86 (3)	76 (4)	84 (–)

seconds as a resource limit for HYSAT reflects a tradeoff between the accuracy of the safety check, and the performances of safe learning. In our application, 30 CPU seconds are sufficient to perform a number of iterations k in the order of hundreds. Considering that most bugs can be found withing a limited number of iterations – k in the order of tens, running time of a few seconds – this provides a sufficient empirical guarantee of safety.

The conclusions of our experiments are that (i) the off-line setting is unsafe; (ii) online learning, even starting from a safe choice of parameters, improves performances over time, but it hinders safety; and (iii) safe on-line learning can be almost as effective as its counterpart, and it guarantees safety.

In more detail, considering the parameters $\mathbf{p}$ used in the off-line setting, HYSAT is able to detect an unsafe state within 3.29 CPU seconds and $k = 13$. On the other hand, the set of bootstrap parameters $\mathbf{p}_0$ used in the on-line settings could not be found unsafe after about 7 CPU hours and $k = 946$.[1] One may ask how significant it is in practice that the off-line setting is unsafe. As we can see in Table 1, 30% of the players performed shots such that the control system proved to be unsafe in the off-line setting.

A similar problem occurs in the on-line setting. Looking at Table 1, we can see that coordination emits unsafe target positions in eight cases. Interestingly, seven cases cases occur with players #8 and #9, indicating that, no matter how effective the model becomes, still there is chance to emit unsafe predictions. In the on-line safe setting, 879 shots were detected, and none of them triggered an unsafe target prediction. In order to evaluate effectiveness, for each player we consider the parameters $\mathbf{p}$ obtained at the end of the corresponding session, so that we can compute (ρ_{ee}, θ_{ee}) with equation (1). To measure the effectiveness of the parameters $\mathbf{p}$, we extract input coordinates and reference target positions from the off-line training set, and we compute the root mean square error (RMSE) between the reference target positions and the results of equation (1). This gives us an idea about how much the control system is effective and how much it improves across the sequence of plays. In Figure 2 we show the RMSE values in for ρ_{ee} (left) and θ_{ee} (right). As we can see, both on-line settings are able to improve their effectiveness while playing. The basic on-line setting can actually outperform the safe on-line setting. However, we know it is unsafe. On the other hand, in the safe online setting, adaptation is slower, and the final target RMSE is higher. However, the

[1] Experiments with HYSAT are performed on a family of identical Linux workstations comprised of 10 Intel Core 2 Duo 2.13 GHz PCs with 4GB of RAM.

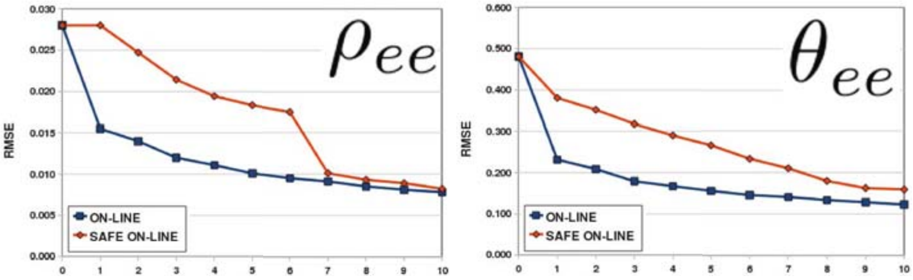


Fig. 2. RMSE on the prediction of ρ_{ee} and θ_{ee}. On the x axis the sequence of subjects, in the same order in which they played the games.

differences between the final RMSE values is not high – 5% of the on-line RMSE for ρ_{ee} and 23% of the on-line RMSE for θ_{ee} – whereas the resulting control system is safe.

5 Discussion and Related Works

Our results show that verifying the safety of a multi agent control system with learning components is doable, even under the tight constraints imposed by real world applications. While ensuring formal safety of the control system is just one of the tasks involved in robot's functional safety – see, e.g. [13] – if a robot is to be deployed in a context where it can harm humans, it is expected that its architecture and implementation reach the highest safety integrity levels (SILs). In most cases, achieving such levels entails the use of structured design methods – such as hybrid automata – and the provision of formal guarantees of safety for the implementations – such as Model Checking or other verification methods. Therefore, we see our approach as one of the key elements towards reaching industry-standard safety for robots operating outside structured environments.

The issue of copying with formal safety in robot control systems has been posed already. The ICRA workshop "Formal Methods in Robotics and Automation" – see, e.g., [14] for the latest event – focuses precisely on this theme. However, to the best of our knowledge, this is the first time in which the issue is investigated in the context of a fast-paced task like the air hockey game, without using simulation results only, and including adaptive components. For other works in robotics dealing with the issue of safety, but not with the problem of formal verification, see, e.g., [15].

Some work has been done in the AI community towards providing a formal framework for checking the safety of adaptive agents. Relevant contributions include a series of paper by Gordon, see e.g. [16]. In these contributions, adaptation is explored as a challenge for safety, but in all of them the formal model of the system does not include continuous state variables, which makes them unsuitable for the kind of modeling required by physical robots. A rather different approach in exploring the same challenge is taken in [17] where the problem of learning how to combine different control modes in a safe way is studied. While this approach is suitable for physical robots, it requires knowledge of the systems' models to be applied. In our case, it is exactly the systems' model which is sought by learning, which makes the approach in [17] inadequate.

References

1. Kramer, J., Scheutz, M.: Development environments for autonomous mobile robots: A survey. Autonomous Robots 22(2), 101–132 (2007)
2. Bagnell, J.A., Schaal, S.: Special issue on Machine Learning in Robotics (Editorial). The International Journal of Robotics Research 27(2), 155–156 (2008)
3. Clarke, E.M., Grumberg, O., Peled, D.A.: Model checking. Springer, Heidelberg (1999)
4. Kern, C., Greenstreet, M.R.: Formal verification in hardware design: a survey. ACM Transactions on Design Automation of Electronic Systems (TODAES) 4(2), 123–193 (1999)
5. Visser, W., Havelund, K., Brat, G., Park, S.J., Lerda, F.: Model checking programs. Automated Software Engineering 10(2), 203–232 (2003)
6. Plaku, E., Kavraki, L.E., Vardi, M.Y.: Hybrid systems: From verification to falsification. In: Damm, W., Hermanns, H. (eds.) CAV 2007. LNCS, vol. 4590, pp. 463–476. Springer, Heidelberg (2007)
7. Bentivegna, D.C., Atkeson, C.G., Cheng, G.: Learning tasks from observation and practice. Robotics and Autonomous Systems 47(2-3), 163–169 (2004)
8. Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.H.: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. LNCS, pp. 209–229. Springer, Heidelberg (1993)
9. Franzle, M., Herde, C., Teige, T., Ratschan, S., Schubert, T.: Efficient solving of large nonlinear arithmetic constraint systems with complex boolean structure. Journal on Satisfiability, Boolean Modeling and Computation 1, 209–236 (2007)
10. Metta, G., Fitzpatrick, P., Natale, L.: YARP: yet another robot platform. International Journal on Advanced Robotics Systems 3(1), 43–48 (2006)
11. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: An update. SIGKDD Explorations 11, 10–18 (2009)
12. Shevade, S.K., Keerthi, S.S., Bhattacharyya, C., Murthy, K.R.K.: Improvements to the SMO algorithm for SVM regression. IEEE Transactions on Neural Networks 11(5), 1188–1193 (2000)
13. Smith, D.J., Simpson, K.G.L.: Functional safety: a straightforward guide to applying IEC 61508 and related standards. Butterworth-Heinemann (2004)
14. Pappas, G., Kress-Gazit, H. (eds.): ICRA Workshop on Formal Methods in Robotics and Automation (2009)
15. Cervera, E., Garcia-Aracil, N., Martinez, E., Nomdedeu, L., del Pobil, A.P.: Safety for a robot arm moving amidst humans by using panoramic vision. In: IEEE International Conference on Robotics and Automation, ICRA 2008, pp. 2183–2188 (2008)
16. Gordon, D.F.: Asimovian adaptive agents. Journal of Artificial Intelligence Research 13, 95–153 (2000)
17. Perkins, T.J., Barto, A.G.: Lyapunov design for safe reinforcement learning. The Journal of Machine Learning Research 3, 803–832 (2003)

Intelligent Training in Control Centres Based on an Ambient Intelligence Paradigm

Luiz Faria[1], António Silva[1], Carlos Ramos[1], Zita Vale[1], and Albino Marques[2]

[1] GECAD – Knowledge Engineering and Decision Support Group / Institute of Engineering – Polytechnic of Porto, Portugal
{lef,ass,csr,zav}@isep.ipp.pt
[2] REN – Transmission Network, Portugal
albino.marques@ren.pt

Abstract. This article describes a new approach in the Intelligent Training of Operators in Power Systems Control Centres, considering the new reality of Renewable Sources, Distributed Generation, and Electricity Markets, under the emerging paradigms of Cyber-Physical Systems and Ambient Intelligence. We propose Intelligent Tutoring Systems as the approach to deal with the intelligent training of operators in these new circumstances.

Keywords: Ambient intelligence, Artificial intelligence, Collaborative work, Cyber-physical systems, Intelligent tutoring systems, Power system restoration, SCADA systems.

1 Introduction

Traditionally, Power Systems were conceived to provide electrical energy under security conditions. Guaranteeing sustainable development is a huge challenge for Power Systems. This requires a significant increasing in Distributed Generation, mainly based on renewable sources. However, this leads to a system that is much more complex to control, since we have many more Power Generation plants, and the generation is more unpredictable than before, due to the difficulty in forecasting the energy production in some renewable sources (e.g. wind and photovoltaic).

The introduction of Electricity Markets, show us the fragility of Power Systems infrastructures. Several severe incidents, including blackouts, occurred (e.g. the 14th August 2003 Blackout in USA, and the 4th October 2006 quasi-blackout affecting 9 European countries). However, the problem is more human than technical, since some of these incidents were caused or had increased consequences due to operators' mistakes, namely at the level of Power Systems Control Centers (CC)[1]. There is a trend to the degradation of this situation with the increasing in the complexity of Power Systems due to the augmenting number of Renewable source installations in the near future.

Thus, training Power Systems Control Centres' Operators for this new reality is a critical goal. For this purpose we are developing CITOPSY (Cyber-Ambient Intelligent Training of Operators in Power Systems CC) project. The project considers the following emerging and advanced paradigms:

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 143–153, 2010.

– Cyber-Physical Systems (CPS), since Power Systems networks, strongly based on SCADA systems, that are expected to evolve for Intelligent SCADA, are typically pointed as a reference example of CPS;

– Ambient Intelligence (AmI) [2], since we claim that CC are a very good example of environment where AmI makes sense;

– and Intelligent Tutoring System (ITS), an advanced technology that is now achieving the maturity to be used in real-world applications, and that is specially adequate for training, namely when combined with simulators. Here we have some experience in the development of ITS for Power Systems incident analysis and diagnosis, power restoration, according to an integrated view [3].

2 Background

In this section we will describe the following aspects: Cyber-Physical Systems (CPS); Ambient Intelligence (AmI); and Intelligent Tutoring Systems (ITS). These paradigms and technologies will be seen in accordance with the Power Systems Control Centres perspective and under the new Electricity Markets and Renewable Sources reality.

CPS are computing systems interacting with physical processes. CPS are typically designed as networks of interacting elements instead of as standalone devices [4]. CPS use computations and communication deeply embedded in and interacting with physical processes to add new capabilities to physical systems [5]. CPS must be dependable, secure, safe, efficient, and operate in real-time. They must also be scalable, cost-effective and adaptive (http://varma.ece.cmu.edu/cps/CFP.htm). The integration of computational and physical processes exhibits a complex behaviour that cannot be analyzed by the computational or physical sciences alone. These systems also transcend traditional computer-controlled systems because of their scale, dependence on man-machine interaction and their rich communication infrastructure that is enabled by the Internet (http://www.qhdctc.com/wcps2008/center.htm).

CPS range from miniscule (pace makers) to large-scale (the national power-grid, blackout-free electricity generation and distribution, optimization of energy consumption) [5].

Power Networks' critical physical infrastructure depends crucially on SCADA (Supervisory Control and Data Acquisition) and DCS (Digital Control Systems) for sensing, monitoring, gathering, and control of distributed physical infrastructures. Power Systems CC are the place where all the information of SCADA and DCS arrive, and CC Operators' must handle the huge amount of data and information arriving from these systems, namely when we are in the presence of critical incidents. Now Cyber-Physical Intelligence is emerging as an important sub-area. The concept of Intelligent SCADA, with decentralized, flexible, and intelligent behavior, being dynamically adaptive to the context of the Power System is appearing [6].

Ambient Intelligence (AmI) can be seen as a digital environment that proactively, but sensibly, supports people in their daily lives [7]. The European Commission's IST Advisory Group - ISTAG believes that it is necessary to take a holistic view of Ambient Intelligence (AmI), considering not just the technology, but the whole of the innovation supply-chain from science to end-user, and also the various features of the academic, industrial and administrative environment that facilitate or hinder realisation of the AmI vision [8].

Most of the work developed today in AmI deals with environments like home, offices, hospitals, cars, transportation systems, museums, and cultural heritage [9]. In [2] authors point to CC like one of the tentative environments where the Ambient Intelligence will be used in the future.

ITS are able to instruct and train students and professionals without the intervention of human beings [10]. ITS introduce a set of ideas like the domain knowledge representation, allowing the possibility to reason and explain automatically on domain problems. Developments were made in trainees' models, instructional and pedagogical planning, and user interface. Some ITS ideas were incorporated in other computer-aided instruction paradigms, like e-learning and distributed learning. However there is a clear difference since these learning concepts are more centered in the interaction with the instructor. Today, ITS can be produced by authoring tools [11], and specific evaluation and assessment methods can be used, appearing sometimes in the form of Cognitive Tutors [12].

ITS were applied in many domains, from medicine to computer programming. There is a set of innovative applications of ITS for training, even in critical and complex domains. Some of these applications provide on-the-job training and can be combined with domain simulators [13].

Electricity Markets bas been a challenge for Power Systems & Economics scientific communities [14]. They introduced a new degree of complexity in Power System Networks control. On the other hand, for environmental reasons most of the new generation plants are based on Renewable Sources of Energy, for which production forecasting is more difficult to perform. This new Distributed Generation scenario points out for an urgent need to create a new generation of tools for training CC operators, combining technologies like ITS, AmI, and CPS.

3 An Intelligent Tutoring System for the Diagnosis and Restoration of Power Systems

The main goal of this work is to supply CC operators with an intelligent environment, fully integrated with the SCADA system, and to provide the CC room with training abilities. In order to accomplish this goal, we will proceed with the integration of an ITS for the diagnosis and restoration of Power Systems in a CC room.

3.1 Tutoring Environment Architecture

Fig. 1 shows this tutoring environment architecture, composed of three complex systems: a CPS module responsible for the acquisition and treatment of the network's physical data; an AmI system (Interaction Manager) that allows the trainee's immersion in the control room environment; and a tutoring system responsible for the pedagogical process. The tutoring system involves two main areas: one devoted to the training of fault diagnosis skills and another dedicated to the training of power system restoration techniques.

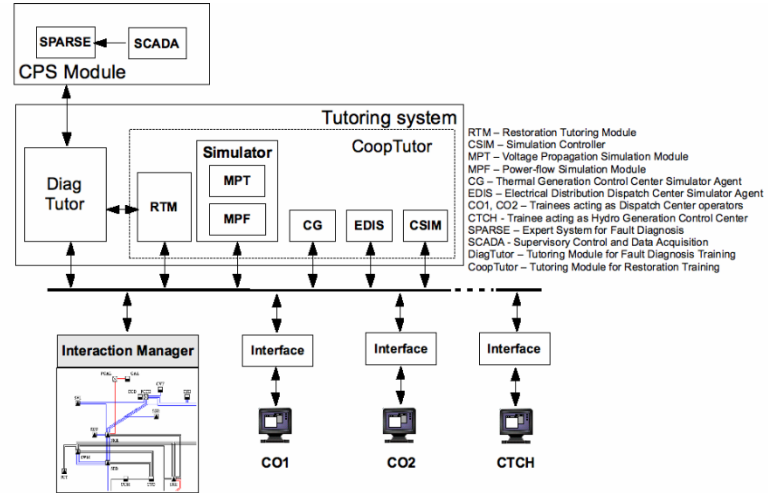

SPARSE
SCADA
CPS Module
Tutoring system
CoopTutor
Diag Tutor
RTM
Simulator
MPT
MPF
CG
EDIS
CSIM
Interaction Manager
Interface
Interface
Interface
CO1
CO2
CTCH
RTM – Restoration Tutoring Module
CSIM – Simulation Controller
MPT – Voltage Propagation Simulation Module
MPF – Power-flow Simulation Module
CG – Thermal Generation Control Center Simulator Agent
EDIS – Electrical Distribution Dispatch Center Simulator Agent
CO1, CO2 – Trainees acting as Dispatch Center operators
CTCH - Trainee acting as Hydro Generation Control Center
SPARSE – Expert System for Fault Diagnosis
SCADA - Supervisory Control and Data Acquisition
DiagTutor – Tutoring Module for Fault Diagnosis Training
CoopTutor – Tutoring Module for Restoration Training

Fig. 1. Tutoring Environment Architecture

The selection of the adequate established restoration procedure strongly depends on the correct identification of the Power System operation state. Therefore, the identification of the incidents or set of incidents occurring in the transmission network is of utmost relevance in order to establish the current Power System operation state. Thus the proposed training framework divides operator's training in to distinct stages. The first one, as described in session 3.2, is intended to give operators with competence needed to get incident diagnosis. After that, operators are able to use CoopTutor to train their skills to manage the restoration procedures (session 3.3).

3.2 Tutoring Module for Fault Diagnosis Training

During the analysis of lists of alarm messages, CC operators must have in mind the group of messages that describes each type of fault. The same group of messages can show up in the reports of different types of faults. So CC operators have to analyze the arrival of additional information, whose presence or absence determines the final diagnosis.

Operators have to deal with uncertain, incomplete and inconsistent information, due to data loss or errors occurred in the data gathering system.

The interaction between the trainee and the tutor is performed through prediction tables where the operator selects a set of premises and the corresponding conclusion. The premises represent events (SCADA messages), temporal constraints between events or previous conclusions.

DiagTutor does not require the operator's reasoning to follow a predefined set of steps, as in other implementations of the model tracing technique [15]. In order to evaluate this reasoning, the tutor will compare the prediction tables' content with the specific situation model. This model is obtained by matching the domain model with the inference undertaken by SPARSE expert system [1]. This process is used to: identify the errors revealing operator's misconceptions; provide assistance on each problem solving action, if needed; monitor the trainee knowledge evolution; and provide learning opportunities for the trainee to reach mastery. In the area of ITS's this goals has been achieved through the use of cognitive tutors [12].

The identified errors are used as opportunities to correct the faults in the operator's reasoning. The operator's entries in prediction tables cause immediate responses from the tutor. In case of error, the operator can ask for help which is supplied as hints. Hinting is a tactic that encourages active thinking structured within guidelines dictated by the tutor. The first hints are generic, becoming more detailed if the help re-quests are repeated.

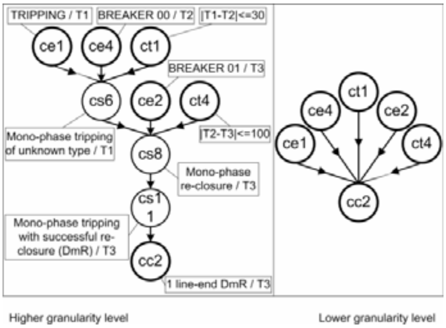


Fig. 2. Higher and lower granularity levels of the situation specific model

The situation specific model generated by the tutoring system for the problem presented is shown in the left frame of Fig. 2. It presents high granularity since it includes all the elementary steps used to get the problem solution. The tutor uses this model to detect errors in the operator reasoning by comparing the situation specific model with the set of steps used by the operator. This model's granularity level is adequate to a novice trainee but not to an expert operator. The right frame of Fig. 2 represents a model used by an expert operator, including only concepts representing events, temporal constraints between events and the final conclusion. Any reasoning model between the higher and lower granularity level models is admissible since it does not include any violation to the domain model. These two levels are used as boundaries of a continuous cognitive space.

Indeed, the process used to evaluate the trainee's reasoning is based on the application of pattern-matching algorithms. Similar approaches with the same propose are used in other ITS's, such as in the TAO ITS [16], an ITS designed to provide tactical action officer students at US navy with practice-based and individualized instruction.

3.3 Tutoring Module for Restoration Training

The management of a power system involves several distinct entities, responsible for different parts of the network. The power system restoration needs a close coordination between generation, transmission and distribution personnel and their actions should be based on a careful planning and guided by adequate strategies [17].

In the specific case of the Portuguese transmission network, four main entities can be identified: the National Dispatch Center (CC), responsible for the energy management and for the thermal generation; the Operational Centre (CO), controlling the transmission network; the Hydroelectric Control Centers (CTCH), responsible for the remote control of hydroelectric power plants and the Distribution Dispatch (EDIS), controlling the distribution network.

The power restoration process is conducted by these entities in such a way that the parts of the grid they are responsible for will be slowly led to their normal state, by performing the actions specified in detailed operating procedures and fulfilling the requirements defined in previously established protocols. This process requires frequent negotiation between entities, agreement on common goals to be achieved, and synchronization of the separate action plans on well-defined moments.

Several agents personify the four entities that are present in the power system restoration process. This multi-agent approach was chosen because it is the most natural way of translating the real-life roles and the split of domain knowledge and performed functions that can be witnessed in the actual power system. Several entities responsible for separate parts of the whole task must interact in a cooperative way towards the fulfillment of the same global purpose. Agents' technology has been considered well suited to domains where the data is split by distinct entities physically or logically and which must interact with one another to pursue a common goal [18].

In this section we describe how we developed a training environment able to deal adequately with the training of the procedures, plans and strategies of the power system restoration, using what may be called lightweight, limited scope simulation techniques. This environment's purpose is to make available to the trainees all the knowledge accumulated during years of network operation, translated into detailed power system restoration plans and strategies, in an expedite and flexible way. The embedded knowledge about procedures, plans and strategies should be easily revisable, any time that new field tests, post-incident analysis or simulations supply new data.

These agents can be seen as virtual entities that possess knowledge about the domain. As real operators, they have tasks assigned to them, goals to be achieved and beliefs about the network status and others agents' activity. They work asynchronously, performing their duties simultaneously and synchronizing their activities only when this need arises. Therefore, the system needs a facilitator (simulator in Fig. 1) that supervises the process, ensuring that the simulation is coherent and convincing.

In our system, the trainee can choose to play any of the available roles, namely the CO and the CC ones, leaving to the tutor the responsibility of simulating the other participants.

The ITS architecture was planned in order that future upgrades of the involved entities or the inclusion of new agents to be simple.

The representation method used to model the trainee's knowledge about the domain knowledge is a variation of the Constraint-Based Modeling (CBM) technique [19]. This student model representation technique is based on the assumption that diagnostic information is not extracted from the sequence of student's actions but rather from the situation, also described as problem state, that the student arrived at. Hence, the student model should not represent the student's actions but the effects of these actions. Because the space of false knowledge is much greater than the one for the correct one, it was suggested the use of an abstraction mechanism based on constraints. In this representation, a state constraint is an ordered pair (Cr,Cs) where Cr stands for relevance condition, and Cs for satisfaction condition. Cr identifies the class of problem states in which this condition is relevant and Cs identifies the class of relevant states that satisfy Cs. Under these assumptions, domain knowledge can be represented as a set of state constraints. Any correct solution for a problem cannot violate any of the constraints. A violation indicates incomplete or incorrect knowledge and constitutes the basic piece of information that allows the Student Model to be built on.

This CBM technique does not require an expert module and is computationally undemanding because it reduces student modeling processing to a basic pattern matching mechanism [20]. One example of a state constraint, as used in our system, can be found below:

If there is a request to CTCH to restore the lines under its responsibility

Then the lines that connect to the hydroelectric power plants must already have been restored

Otherwise an error has occurred

Each violation to a state constraint like the one above enables the tutor to intervene both immediately or at a later stage, depending on the seriousness of the error or the pedagogical approach that was chosen.

This technique gives the tutor the flexibility needed to address trainees with a wide range of experience and knowledge, tailoring, in a much finer way, the degree and type of support given, and, at the same time, spared us the exhaustive monitoring and interpretation of student's errors during an extended period, which would be required by alternative methods.

Nevertheless, it was found the need for a meta-knowledge layer in order to adapt the CBM method to an essentially procedural, time-dependent domain like the power system restoration field. In fact, the validity of certain constraints may be limited to only parts of the restoration process. On the other hand, the violation of a constraint can, in certain cases, render irrelevant the future verification of other constraints. Finally, equally valid constraints in a certain state of the process can have different relative importance from the didactic point of view. This fact suggests the convenience of establishing a constraint hierarchy.

This meta-knowledge layer is composed of rules that control the constraints' application, depending on several issues: the phase of the restoration process in which the trainee is; the constraints previously satisfied; and the set of constraints triggered simultaneously.

These rules establish a dependency network between constraints that can be represented by a graph (Fig. 3). The nodes (1-15) represent constraints. The relationships between constraints expressed by this graph can be of precedence, mutual exclusion or priority.

This tutoring module is able to train individual operators as if they were in a team, surrounded by virtual "operators", but is also capable of dealing with the interaction between several trainees engaged in a cooperative process. It provides specialized agents to fulfill the roles of the missing operators and, at the same time, monitors the cooperative work, stepping in when a serious imbalance is detected. The tutor can be used as a distance learning tool, with several operators being trained at different locations.

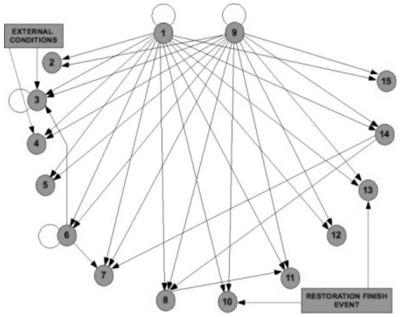


Fig. 3. Constraint Dependency Graph

4 Modelling Power Systems Control Centre Operators According the New Reality

Several acquisition methods and representation techniques have been proposed during the last decades to tackle the problem of how to represent the knowledge describing the student characteristics and preferences. Most of this work has been done to handle academic tutoring studies (e.g. training students on Mathematics, Physics or Programming). On-the-job training is a completely different matter. Trainees are also employees with very well defined tasks and responsibilities like to guarantee the effective operation of Power Systems networks in the case of CC Operators. So Power Systems CC Operators cannot be modeled like traditional trainees. For example, if an operator is being trained on-the-job, and some critical incident occurs he will need to stop training and pay attention to this new incident. This reality is always in the back of the mind of the Operators so training will need to be intelligent enough to catch the operators' model according these conditions.

With the expected evolution for Intelligent SCADA systems, more decentralized, flexible, and context-oriented, CC Operators will need to adopt a different level of skills (e.g. understanding when an incident affect areas strongly dependent of renewable sources). This means to introduce a semantic level above the SCADA system. The operator should work on the top of this new level.

Beyond the information about the trainee knowledge, the user model must trace the trainee misconceptions and knowledge about his actions during problem solving and the corresponding Tutor's feedback. This kind of knowledge will be maintained in the operator's historic training.

Another issue is the identification of a set of general factors concerning the trainee, such as acquisition and retention. Prior research indicates that examining general factors, like those referred above, can be useful to operator modeling. We also want to find how these general factors can be predictive of overall learning and contribute to improve the Tutor responses to the particular idiosyncrasies of the operator.

Finally we need to keep in mind the collaborative nature of Operators work, namely when critical incidents are occurring. In our previous work considering Incident Analysis and Diagnosis we considered the Operator alone and we treated this problem as being different from the Power and Service Restoration. Now, due to the increased complexity of Power Systems derived from the need to operate in a Competitive Electricity Market with Distributed Generation, Incident Analysis and Diagnosis will need to involve more people. A better combination with Power Generation companies, Transmission companies, and Distribution companies is much more necessary than in the past.

This means that we cannot model an operator alone. We will need to model a community of operators, working in the same or different companies, cooperating to maintain the Power System working effectively.

So, for this task it is necessary the development of a completely new model of collaborative communities of trainees. Notice that this is completely different from previous works in other areas, with cooperative students, since they are supposed to work in an academic environment and they are training on the same skills.

5 Conclusions

This paper described how Power Systems Control Center operators can be trained in two main tasks: Incident Analysis and Diagnosis; and Service Restoration. We gave some insights about the use of Cyber-Physical Systems and Ambient Intelligence concepts for this purpose, namely in which concerns the new reality of Electricity Markets, Renewable Sources and Distributed Generation. We explained how things change in tasks like the acquisition of domain knowledge on Incident Analysis, Diagnosis, and Power and Service Restoration, and in the modeling of Power Systems CC Operators according the new reality.

ITS is the approach we are using, combining several Artificial Intelligence techniques, namely: Multi-Agent Systems, Neural Networks, Constraint-based Modeling, Intelligent Planning, Knowledge Representation, Expert Systems, User Modeling, and Intelligent User Interfaces.

Concerning the operators' training, the most interesting features are:

1. The connection with SPARSE, a legacy Expert System used for Intelligent Alarm Processing [1].
2. The use of prediction tables and different granularity levels for fault diagnosis training.
3. The use of the model tracing technique to capture the operator's reasoning.
4. The development of two tools to help the adaptation of the curriculum to the opera-tor - one that generates training scenarios from real cases and another that assists in creating new scenarios.
5. The automatic assignment of the difficulty level to the problems.
6. The identification of the operators' knowledge acquisition factors.
7. The automatic selection of the next problem to be presented, using Neural Net-works.
8. The use of Multi-Agent Systems paradigm to model the interaction of several operators during system restoration.

9. The use of the Constraint-based Modeling technique in restoration training.
10. The availability of an Intelligent User Interface in the interaction with the operator.

References

1. Vale, Z., Machado e Moura, A., Fernanda Fernandes, M., Marques, A., Rosado, C., Ramos, C.: SPARSE: An Intelligent Alarm Processor and Operator Assistant for Portuguese Substations Control Centers. IEEE Expert - Intelligent Systems and Their Application 12(3), 86–93 (1997)
2. Ramos, C., Augusto, J.C., Shapiro, D.: Ambient Intelligence: the next step for AI. IEEE Intelligent Systems magazine 23(2), 15–18 (2008)
3. Faria, L., Silva, A., Vale, Z., Marques, A.: Training Control Centres' Operators in Incident Diagnosis and Power Restoration Using Intelligent Tutoring Systems. IEEE Transactions on Learning Technologies (2009)
4. Lee, E.: Cyber Physical Systems: Design Challenges, University of California, Berkeley Technical Report No. UCB/EECS-2008-8
5. CPS Steering Group, Cyber-Physical Systems Executive Summary (2008), `http://varma.ece.cmu.edu/Summit/CPS-Executive-Summary.pdf`
6. Vale, Z., Morais, H., Silva, M., Ramos, C.: Towards a future SCADA. In: IEEE Power and Energy Society General Meeting General Meeting, Calgary, Alberta, Canada (2009)
7. Augusto, J.C., Cullagh, P.M.: Ambient Intelligence: Concepts and Applications. International Journal on Computer Science and Information Systems 4(1), 1–28 (2007)
8. ISTAG, Strategic Orientations & Priorities for IST in FP6, European Commission Report (2002)
9. Ramos, C.: Ambient Intelligence Environments. In: Rabuñal, J., Dorado, J., Sierra, A. (eds.) Encyclopedia of Artificial Intelligence, Information Science Reference, pp. 92–98 (2009) ISBN 978-1-59904-849-9
10. Wenger, E.: Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge. Morgan Kaufmann Publishers Inc., San Francisco (1987)
11. Heffernan, N.T., Turner, T.E., Lourenco, A.L.N., Macasek, M.A., Nuzzo-Jones, G., Koedinger, K.R.: The ASSISTment Builder: Towards an Analysis of Cost Effectiveness of ITS Creation. In: Proceedings of the 19th International Florida Artificial Intelligence, Research Society Conference, pp. 515–520 (2006)
12. Aleven, V., Koedinger, K.R.: An effective meta-cognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. Cognitive Science 26(2), 147–179 (2002)
13. Ramos, C., Frasson, C., Ramachandran, S.: Introduction to the Special Issue on Real World Applications of Intelligent Tutoring Systems. IEEE Transactions on Learning Technologies 2(2), 62–63 (2009) doi:10.1109/TLT.2009.16
14. Praça, I., Ramos, C., Vale, Z., Cordeiro, M.: MASCEM: A Multi-Agent Systems that simulates Competitive Electricity Markets. IEEE Intelligent Systems 18(6), 54–60 (2003)
15. Anderson, J., Corbett, A., Koedinger, K., Pelletier, R.: Cognitive Tutors: Lessons Learned. The Journal of the Learning Sciences 4(2), 167–207 (1995)
16. Stottler, R., Panichas, S.: A New Generation of Tactical Action Officer Intelligent Tutoring System (ITS). In: Proc. Industry/Interservice, Training, Simulation and Education Conference, I/ITSEC (2006)

17. Sforna, M., Bertanza, V.: Restoration Testing and Training in Italian ISO. IEEE Transactions on Power Systems 17(4) (November 2002)
18. Jennings, N., Wooldridge, M.: Applying agent technology. Applied Artificial Intelligence: An International Journal 9(4), 351–361 (1995)
19. Ohlsson, S.: Constraint-Based Student Modeling. In: Greer, McCalla (eds.) Student Modeling: the Key to Individualized Knowledge-based Instruction, pp. 167–189. Springer, Heidelberg (1993)
20. Mitrovic, T., Koedinger, K., Martin, B.: A comparative analysis of Cognitive Tutoring and Constraint-based Modeling. In: em Brusilovsky, P., Corbett, A., Rovis, F. (eds.) 9th International Conference on User Modelling UM 2003. Springer, Heidelberg (2003)

Injecting On-Board Autonomy in a Multi-Agent System for Space Service Providing

Amedeo Cesta[1], Jorge Ocon[2], Riccardo Rasconi[1],
and Ana María Sánchez Montero[2]

[1] ISTC-CNR, Italian National Research Council, Rome, Italy
{name.surname}@istc.cnr.it
[2] GMV, Advanced Space Systems and Technologies, Tres Cantos, Spain
jocon@gmv.com, amsmontero@gmv.com

Abstract. The paper describes recent work for experimenting on-board autonomy capabilities within a larger multi-agent system that provides product delivery for the GMES Earth Observation System. We will show how robust state of the art technology for constraint-based scheduling and execution can be customized for endowing agents with the autonomy capability.

1 Introduction

This paper describes some of the results obtained within the Distributed Agents For Autonomy (DAFA) study supported by the European Space Agency (ESA). The project aimed at demonstrating the advantages of using distributed agents in a space system. During the DAFA study design, a number of case studies were evaluated as candidates for implementation, for example complex missions like Exomars, multi-satellite missions like Swarm, formation flying missions like Darwin, as well as GMES (Global Monitoring for Environment and Security) scenario. Finally, a software demonstrator for DAFA was implemented on the basis of a GMES-like scenario, where Agent-Based Programming was used to inject autonomy for a specific space application. The initial design of the demonstrator (see [6]) was recently integrated with on-board re-planning capability for an Earth observation scenario; a complete description of the latest version of the DAFA demonstrator can be found in [7].

There are many inter-related aspects that have to be taken into account when exploring the concrete advantages of having autonomy on board of spacecraft. Autonomy is often necessary, as in many cases it is not possible to send to the spacecraft the maneuvering instructions with the necessary frequency for various reasons: environmental uncertainty, propagation delays, required response times, etc. Another relevant advantage of autonomy is the reduction of mission operation costs; this is achieved by executing on board many of the operations that are typically performed on the ground. Costs are reduced because ground crews are relieved from the time consuming task of formalizing and synthesizing the detailed sequences of commands to send to the spacecraft as the “intelligent” spacecraft accepts high level goals [5], autonomously reasoning upon them. Costs are also reduced because of the reduction in communications.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 154–163, 2010.

Moreover, autonomy definitely increases the overall space mission quality, by increasing its reliability, flexibility and responsiveness to science opportunities. Reliability is increased because of the spacecraft's capability to robustly respond to possible failures. In the simpler cases, spacecraft can enact predefined emergency plans when needed, while in the more complex cases spacecraft can enable alternative plans synthesized on occasion. Mission flexibility is a byproduct of the previous features: re-planning skills directly translate into the capability of "adapting" to unforeseen situations, which comprise the ability to detect and take advantage of science opportunities, seen as "positive" contingencies. Lastly, the capability to make decisions directly on board generally allows to enable timely reactive behaviors, thus reducing the science product timeliness, which results in a significant increase of the return of mission investment. Autonomous earth event detection and state-of-the-art plan execution management, as well as continuous re-planning capabilities are exhibited for example in the Earth Observing One (EO-1) mission [3], where the fully on-board spacecraft features are exploited with a significant advantage in terms of mission costs. This paper discusses the advantages of using on-board autonomy with respect to the GMES-like scenario, and compares the performances against a nominal case.

2 A Multi-Agent System for Space Services

Within the DAFA study on-board autonomy has been ranked in three categories according to the employed decision-making capabilities of the spacecraft. In particular, *Type I Spacecraft* are those that manage on-board pre-defined plans that cannot be modified, and the users work by subscription to the corresponding spacecraft; *Type II Spacecraft* manage plans that can be modified to adapt to near real time (NRT) requirements, and rescheduling is possible in *emergency mode* situations only; finally, *Type III Spacecraft* have the capability to detect scientific events over specific areas, where the pre-programmed scientific event detections can trigger a direct modification of the spacecraft's on-board plan (autonomous re-planning).

Autonomy can be generally achieved by means of different technologies, and multi-agent systems (MAS) offer one particularly valid alternative. They represent a software engineering paradigm where systems are built-up out of individual problem-solving agents pursuing high-level goals; moreover, due to the MAS-based homogeneous and powerful design environments available nowadays, the development process of MASs is facilitated, as is the deployment of simulation systems. The main objective is to prove that autonomous MASs are able to meet the necessary requirements in order to be possibly used in real-world operational systems.

The GMES Mission Scenario. GMES (Global Monitoring for Environment and Security - http://www.gmes.info/) is a major European Earth Observation Programme initiative for the establishment of All-European Earth observation capabilities. The GMES services are grounded on the coordination of a number of Earth Observation (EO) satellites and on an effective use of different ground segment facilities. In general we can say that our choice of GMES as a test case has been driven by the extreme relevance of the services that this scenario offers; such services can be classified in three major categories:

- **Mapping**, including topography or road maps but also land-use and harvest, forestry monitoring, mineral and water resources that do contribute to short and long-term management of territories and natural resources. This service generally requires exhaustive coverage of the Earth surface, as well as the archiving and periodic updating of data;
- **Support** to civil protection institutions responsible for the security of people and property for emergency management in case of natural hazards. This service focuses on the provision of the latest possible data before intervening;
- **Forecasting**, to be applied for marine zones, air quality or crop fields. This service systematically provides data on extended areas permitting the prediction of short, medium or long-term events, including their modelling and evolution.

The DAFA Multi-Agent Service Provider. Within the DAFA project we have developed a quite realistic demonstrator of a GMES service, by devising a system which is completely agent-programmed. The main contribution we intend to provide within the DAFA distributed architecture is the capability to autonomously schedule observations on behalf of EO Satellite fleets.

This issue entails large search spaces, complex constraint checking, such as power, thermal data, and limited time over the target, and complex bottlenecks related to planning, non-operational times, downlink windows and limited capacities.

The selected GMES scenario deploys a set of Type I, Type II, and Type III spacecraft that cooperate for the fulfillment of EO operations. Within our DAFA GMES demonstrator, two different basic use cases are considered:

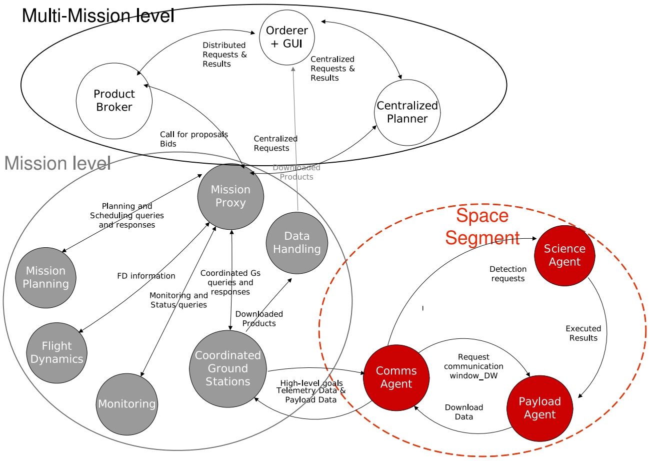


Fig. 1. The agents in the DAFA software

- **Provision of data products.** Different spacecraft compete to provide the best possible data product, taking into account the restrictions that apply for each spacecraft (e.g. next fly-over, current planning, status of the spacecraft and instruments, deadline for delivery, etc.);
- **Detection of a scientific events.** The capabilities of Type III spacecraft is used to detect a given condition. This is a collaborative scenario in which the different spacecraft of Type III work in conjunction to improve the detection of a scientific event. In this scenario agents of different missions collaborate in order to provide early detection.

Figure 1 presents a high level view of the agents that exist in our demonstrator. Note that the agents in white are agents that are Earth-based and that are mission-independent, meanwhile the agents in light gray are agents that are Earth-based and are specific for each mission. In addition, agents in dark grey belong to the space segment. Both agents in light gray and agents in dark grey (with the exception of the *coordinated ground station* agent, which is a single agent for the whole demonstrator) are instantiated for each mission (i.e., a proxy agent is launched for the ENVISAT mission, another proxy agent is instantiated for the ALOS mission, etc.). Note that at the mission and space segment level, we instantiate the corresponding agents for a mission (except the coordinated ground station agent), which means that for a standard simulation with n different missions we have a total of:

- 3 multi-mission agents (orderer, Product broker and centralized planner);
- 5 mission agents per mission (mission planning, mission proxy, flight dynamics, monitoring, and data handling), which yields $5n$ different agents;
- 1 Coordinated ground stations agent;
- 3 agents for the space segment, which yields $3n$ agents.

Overall, we have a total of $4+8n$ agents for a simulation where n missions are deployed, each new mission therefore involving 8 more agents.

Among all the types of agents, the most important agent for the purpose of this paper is the **Science Agent** (SA). In fact, the SA retains all the necessary autonomy to perform the detection of scientific events and, more importantly, on-line planning, scheduling and execution of goal-based plans. The detailed description of the SA features will be the object of Section 3.

3 On-Board Autonomy through Automated P&S Technology

As stated in Section 1, Type I autonomy only entails the capability to run a pre-specified set of actions without intervention from ground; in this case, the analogy with the well-known binomial program-process is very strong: the executing sequences of actions cannot be modified while running. This limitation is overcome with autonomy of Type II, which entails the possibility to change the line of action on the fly with pre-specified plans, depending on the occurring emergency. The Type III spacecraft represent the most advanced concept in terms of on-board autonomy; they have the capability to autonomously detect some pre-defined events, upon which to reason and extract new goals. In fact, they implement a goal-based behavior, where the activity sequence to be executed is synthesized on-board on the base of the objective that has to be reached.

The agentification structure of the Type III spacecraft as developed in our DAFA demonstrator, as well as the relationships among the involved agents are shown in Figure 1. In particular, the agents involved in the space segment are the following:

- The **On-board science analysis agent:** this agent is in charge of observing the Earth for those particular areas of interest using the on-board instruments, detecting the presence of interesting events (e.g fires, oil spills). It also manages the on-board planning and the scheduling of activities to be executed by the spacecraft. This

agent also guarantees the proper execution of commands at the spacecraft level, based on the on-board planning mechanisms and the ground commands that are being sent;
- The **Communication agent:** this agent acts as an interface with the ground system agent, it optimizes the bandwidth and the sending/receiving of data to the ground stations;
- The **Payload agent:** this agent manages and stores the payload data. Uses the X-band communication window available with the Ground stations to download payload data whenever there is an opportunity to do so.

3.1 Spacecraft with On-Board Planning Capabilities

Figure 2 depicts the structure that defines the goal-based behavior of the Type III SA that has been implemented in the DAFA architecture. As previously stated, one of the most appealing aspects of autonomy is the capability to synthesize plans based on the notion of high level objectives; our SA is designed to accept such high level goals, prioritize them and use such information to extract from a *plan library* the most suitable line of action that fulfills the selected goal.

All relevant information about the plan currently under execution is stored in the Current Plan Database (CPD); the data stored in the CDB concern the execution status of the activities (terminated, under execution, to be executed), and, more importantly, all the temporal information related to the constraints that might be imposed among the activities. Such information is actually maintained by means of a Simple Temporal Network (STN) [4], which underlies the current plan. The STN maintains the information about the distances between any pair of time points of the network, which gets properly updated each time a new temporal constraint is inserted. Through the STN it is therefore possible to create and maintain a complex structure of temporal relationship that generally involve all the plan activities; such relationships are continuously enforced in the plan, and this feature is useful to control how unexpected events occurring on a single activity (i.e., delays, duration changes, etc.) propagate to the rest of the plan.

The SA is also able to control its execution by dispatching the activities currently present in the executing plan in due time, by simulating the passing of time in the CPD, and triggering the tasks whose start time meets the current time of execution. It should be noticed that simulating the passing of time entails a continuous modification (i.e., firing of the temporal propagation algorithms) of the STN underlying the plan; in general, temporal constraints should be propagated for executed time

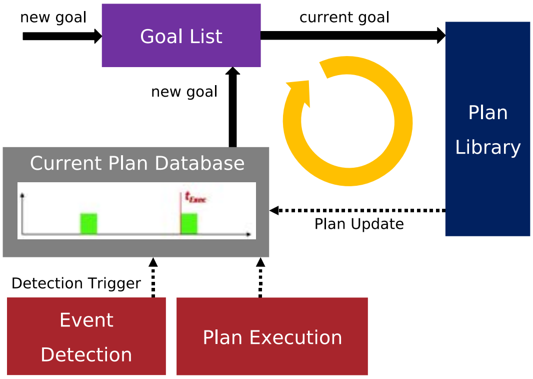


Fig. 2. Our Type III Science Agent architecture

Algorithm 1. Executing and dynamically updating the Plan through a goal-oriented approach

```
Input: GoalList, Plan, PlanLibrary
Output: Executed Plan
while (TRUE) do
    // Receiving new goals
    if (New Goal has been received) then
        GoalList ← UpdateGoalList (Goal, GoalList)
    // Re-planning step
    if (GoalList NOT empty) then
        Plan ← UpdatePlan (Goal, PlanLibrary, Plan)
    advanceExecTime ()
    // Activity Execution step
    if (isExecuted (Activity)) then
        if (Activity == Detection) ∧ (Event is detected) then
            GoalList ← UpdateGoalList (Goal, GoalList)
        else if (Activity == ScienceProduction) then
            informPayload ()
```

points, in order to fixate their temporal flexibility to the current instant of execution (i.e., terminated activities can no longer undergo temporal modifications). Hence the importance of efficient propagation algorithms.

Moreover, the autonomous *Event Detection* capability of the SA allows to simulate the detection of particular events as a result of the execution of a detection task. This ability is of primary importance, as it allows to generate new goals to be inserted in the Goal List at execution time, therefore closing the *sense-plan-act* loop with the real world, and ultimately enabling a continuous goal-based replanning behavior.

The Algorithm 1 presents the scheme of the execution steps performed in the SA, that were described above. The SA enables two *behaviors* in two concurrent threads, the *Listener* behavior which accepts all the incoming goals from Ground and actually performs the replanning by updating the current plan, and the *Executor* behavior, which is in charge of dispatching the scheduled tasks in due time. In the algorithm, both behaviors are shown as a single execution thread, for the sake of simplicity.

At each execution cycle the Listener behavior receives the incoming goals and updates the Goal List; then, a goal is selected and the current plan is updated by choosing the most appropriate sub-plan from the Plan Library and by scheduling it on the current global plan. Scheduling entails the removal of the possible resource conflicts between concurrent tasks, while enforcing the feasibility of all the necessary previously imposed temporal constraints.

As the execution time advances, the Executor behavior checks whether some activity has completed its task; in case a *detection* task has been completed and the related detection is positive (i.e., a fire has been observed), then a new goal is added to the Goal List for future re-planning; in case a generic “science production” task has been completed, the `informPayload()` function is launched, in order to inform the Payload Agent (see Figure 1) that there is a science product ready to be dumped to Ground.

3.2 The JOSCAR Timeline-Based Package

The central idea in enforcing autonomy in satellite operations is the exploitation of on-board planning and scheduling capabilities. More specifically, the current DAFA SA can autonomously decide to insert/remove tasks to/from the on-board executing plan; it has the capabilities to decide feasible insertion times by checking out the onset of possible resource conflicts that may be raised during operations because of the oversubscription of on-board resources. Due to the very mature development & testing stage of the technology used to provide the Planning & Scheduling services, the tailoring phase to adapt our P&S technology to the specific requirements of the DAFA project required a relatively straightforward process. Such features have been implemented by exploiting the services offered by the JOSCAR (Java Object-oriented SCheduling ARchitecture) library, purposely devised for DAFA to provide all the necessary APIs to create, manage and reason upon plans and/or schedules, intended as sequences of temporal events.

In the DAFA SA, plans and/or schedules are reasoned upon as a special type of Constraint Satisfaction Problem (CSP); the scheduling problem we deal with is well-known in the Operations Research (OR) literature, and derives from a project management environment in which activities represent steps that must be performed to achieve project completion (see [2]). These activities are subject to partial order constraints that reflect dependencies on project progression. Such problem represents a specific formulation of the basic scheduling issues which underlie a number of real-world applications and is considered particularly difficult, due to the presence of temporal separation constraints (in particular maximum time lags) between project activities.

The JOSCAR library is implemented so as to capture all the modelling features that are strictly connected with the scheduling of resource-consuming activities in a *Timeline*. A Timeline can be informally described as an ordered sequence of temporal events. The JOSCAR library has been designed to exploit the efficacy of the Timeline-based Planning & Scheduling centered around state-of-the-art Constraint Reasoning techniques. By exploiting this constraint reasoning technology, it is possible to represent all the significant aspects of a planning and scheduling problem in the Timeline (i.e., both the temporal and resource aspects). The capability to efficiently manage a Timeline object therefore corresponds to being in control of the most important modelling and reasoning aspects of the Planning & Scheduling problem. The basic class the JOSCAR library provides is the *TimelineManager* class; by instantiating a *TimelineManager* object it is possible to represent the activities in the plan, control their precise timing data as well as their mutual temporal relationships, check the schedule's resource feasibility through profile-based resource conflict detection routines, as well as requesting initial scheduling solutions for further refinement by means of a built-in scheduling algorithm.

3.3 Using JOSCAR for Onboard Replanning

In this Section, we will briefly describe some of the JOSCAR methods that have been used to implement the DAFA SA's execution and re-planning features described in the algorithm 1. The *UpdatePlan()* function is implemented through the use of the `createActivity()`, `addActivity()` and `retractActivity()` primitives, which allow the dynamic management of a network of tasks; through these methods it is possible to respectively create, add and retract activity from the executing timeline on the fly. Moreover, the dynamic management of the constraint network is guaranteed by the use of the following JOSCAR primitives: `createReleaseConstraint()`, `createPrecedenceConstraint()`, `createDeadlineConstraint()`, `addTemporalConstraints()` and `retractTemporalConstraints()`. As their names suggest, these methods allow for the easy creation and management of any type of temporal constraint that might be present in a plan. Each of the previous primitives entails the execution of state-of-the-art temporal propagation algorithms, whose efficiency is therefore of primary importance. The conflict detection capability is guaranteed by the `isConflictFree()` and the `showResourceProfiles()` primitives, which return the information about the possibly conflicting activities in the executing plan. The built-in scheduling capability is provided by means of the `solve()` primitive, which can be used to attempt a resource conflict resolution, in order to guarantee the continuous resource feasibility of the current solution, while the time feasibility is inherently maintained by the above mentioned temporal propagation algorithms.

3.4 A Practical Case: Running the Demonstrator

The following is a description of the results obtained when running the demonstrator, simulating a particular GMES scenario. In particular, the simulation includes a constellation of nine spacecraft, three of which are of type III (i.e., endowed with additional autonomy). The configuration of each spacecraft is different, and matches the characteristics of real spacecraft that are currently being used for Earth observation (instrument,

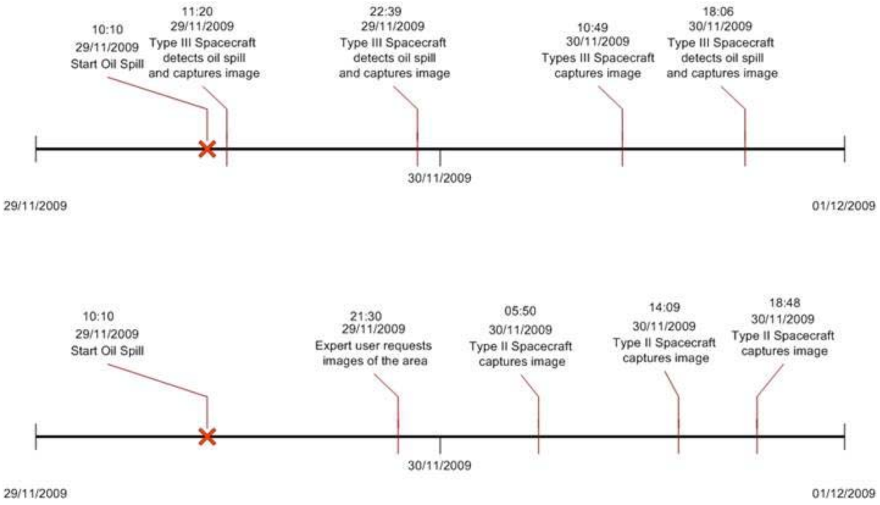


Fig. 3. Timeliness when using several spacecraft with on-board re-planning (upper timeline) and without on-board re-planning (lower timeline) capabilities

instrument parameters, orbits, restrictions of use, communication bandwith, as well as visibility windows). The simulation is centered upon detecting oil spills, and illustrates the advantages of exploiting autonomous on-board re-planning capabilities in a multi-agent system. To this aim, two approaches to this emergency scenario have been reproduced: a first approach, where onboard autonomy is disregarded and the spacecraft are not capable of autonomously detecting the event, and the products are requested from ground (when an oil spill is detected, a team of experts will request the spacecraft that overfly the area for image products to analyze the situation and solve the emergency according to the obtained images); a second approach where autonomy is exploited for early event detections; in this approach, due to the high probability of oil spills, the type III spacecraft will be requested to check for an oil spill on a certain area. Upon event detection on behalf of any of the spacecraft, an image of the event will be automatically taken and downloaded to the first available ground station; moreover, the spacecraft will reschedule its plan so as to keep capturing images of the area of interest until the emergency is resolved.

Figure 3 shows two timelines comparing spacecraft performances, respectively with and without on board autonomy. The first time line (shown on Figure 3) shows the results when several spacecraft of type III are requested to observe an area for a possible disaster. In this particular scenario, an oil spill occurs at 10:10 am close to the Spanish coast. At 11:20h a type III spacecraft detects the oil spill, takes an image of the area and reschedules its plan so that a further image will be taken at the next over flight of the same area (on the following day, at 10:49h). Similarly, at 22:39, another type III spacecraft overflies the area detecting the oil spill, taking an image and rescheduling a new product for the next over-flight. At 18:06 of the next day, an additional Type III spacecraft detects the same oil spill and takes the corresponding image. As a result, since the onset of the event at 10:10 of the first day, four different images of the area have been taken at different times (within 18:06 of the next day) without the need of human intervention.

The second timeline in Figure 3 shows the results when using type II spacecraft only. Following the same pattern of the previous execution, the oil spill starts at 10:10 am. As type II spacecraft capabilities do not allow for an immediate notification, an extra time inevitably elapses before the authorities acknowledge the event and an emergency plan is activated. Such plan entails the involvement of ground operator to re-program the interested spacecraft to obtain the corresponding images of the area with the proper characteristics. Eventually, the first image available will be the one taken at 21:30, followed by other images at 05:50, 14:09 and 18:43 taken by different spacecraft, i.e., the first captured image of the area is taken more than 17 hours later than in the previous case. In large emergencies like oil spills and fires, fast responsiveness is a key factor: therefore, the possibility to obtain immediate evidence of the events represents an invaluable asset for the user. Conversely, the second timeline describes a situation where it is possible to rely on a large number of spacecraft to provide more images, but much later and at a much higher costs. Although this is a particular scenario, different executions for other similar scenarios have consolidated these results: in all cases that a Type III spacecraft overflies the area in the next 12 hours after the event occurs, the results are much better than when using non-autonomous spacecraft.

4 Discussion and Conclusions

In this work we have described a software system based on multi-agent technology that was developed within the Distributed Agent For Autonomy (DAFA) study as a demonstrator for the Global Monitoring for Environment and Security (GMES), an ambitious initiative jointly promoted by the European Space Agency and the European Commission. In particular, the major objective of the DAFA demonstrator was to show the benefits of both agent-oriented programming and on-board autonomy in space missions, by deploying multi-agent based spacecraft capable of autonomous decision-making skills. We have described the overall system, focusing our attention on its multi-agent inner structure combined with the state-of-the-art Planning & Scheduling technology that was employed to guarantee such autonomous capabilities.

The main result of this work is therefore twofold: on one hand, we have shown that P&S technology is robust enough to be employed to solve real-world problems by properly using the off-the-shelf P&S components currently available from the related research areas (see [1] for another example of real-world deployment of AI-based P&S technology which relies on ”of the shelf” support technology); on the other hand, we have shown that multi-agent systems are mature enough to be integrated with truly advanced problem solving capabilities. Experiments carried out with the DAFA demonstrator show how the ability to perform on-board planning decisions by autonomously deciding the current line of action depending on the detected events and/or opportunities, can represent a significant advantage in terms of mission cost reduction, increased quality of science products, and response timeliness.

Acknowledgments. Authors are partially supported by ESA under project DAFA (AO/1-5389/07/NL/HE). Thanks to Quirien Wijnands and Joachim Fuchs for their continuous supervision on our work.

References

1. Cesta, A., Cortellessa, G., Fratini, S., Oddi, A.: MrSPOCK – Steps in Developing an End-to-End Space Application. Computational Intelligence (in press, 2010)
2. Cesta, A., Oddi, A., Smith, S.F.: A Constraint-based Method for Project Scheduling with Time Windows. Journal of Heuristics 8(1), 109–136 (2002)
3. Chien, S., Sherwood, R., Tran, D., Cichy, B., Rabideau, G., Castano, R., Davies, A., Mandl, D., Frye, S., Trout, B., Shulman, S., Boyer, D.: Using Autonomy Flight Software to Improve Science Return on Earth Observing One. Journal of Aerospace Computing, Information, and Communication 2(4), 196–216 (2005)
4. Dechter, R., Meiri, I., Pearl, J.: Temporal Constraint Networks. Artificial Intelligence 49, 61–95 (1991)
5. Muscettola, N., Nayak, P., Pell, B., Williams, B.C.: Remote Agent: To Boldly Go Where No AI System Has Gone Before. Artificial Intelligence 103(1-2), 5–48 (1998)
6. Ocon, J., Rivero, E., Strippoli, L., Molina, M.A.: Agents for Space Operations. In: Proceedings of the SpaceOps Conference, AIAA (2008)
7. Ocon, J.: DAFA - Distributed Agents For Autonomy - Final Report. Technical report, GMV Aerospace and Defence S.A. (October 2009)

Solving Portfolio Optimization Problem Based on Extension Principle

Shiang-Tai Liu

Graduate School of Business and Management, Vanung University
Chung-Li, Tao-Yuan 320, Taiwan, ROC
stliu@vnu.edu.tw

Abstract. Conventional portfolio optimization models have an assumption that the future condition of stock market can be accurately predicted by historical data. However, no matter how accurate the past data is, this premise will not exist in the financial market due to the high volatility of market environment. This paper discusses the fuzzy portfolio optimization problem where the asset returns are represented by fuzzy data. A mean-absolute deviation risk function model and Zadeh's extension principle are utilized for the solution method of portfolio optimization problem with fuzzy returns. Since the parameters are fuzzy numbers, the gain of return is a fuzzy number as well. A pair of two-level mathematical programs is formulated to calculate the upper bound and lower bound of the return of the portfolio optimization problem. Based on the duality theorem and by applying the variable transformation technique, the pair of two-level mathematical programs is transformed into a pair of ordinary one-level linear programs so they can be manipulated. It is found that the calculated results conform to an essential idea in finance and economics that the greater the amount of risk that an investor is willing to take on, the greater the potential return. An example illustrates the whole idea on fuzzy portfolio optimization problem.

Keywords: Portfolio optimization; Extension principle; Absolute deviation function; Fuzzy set.

1 Introduction

Portfolio theory and related topics are among the most investigated fields of research in the economic and financial literatures. The well-known Markowitz's mean-variance approach [9,10] requires to minimize the risk of the selected asset portfolio, while guaranteeing a pre-established return rate and the total use of the available capital.The seminal works of Markowitz maintain strong assumption on probability distributions and Von Neumann-Morgenstern utility functions. Moreover, as the dimensionality of the portfolio optimization problem increases, it becomes more difficult to solving a quadratic programming problem with a dense covariance matrix. Several researches were proposed for alleviating the computational difficulty by using various approximation schemes [15-17]. The use of the index model enables one to reduce the amount of computation by introducing the notion of "factors" influencing the stock prices [12,14].

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 164–174, 2010.

Yet, these efforts are largely discounted because of the popularity of equilibrium models such as capital asset pricing model (CAPM) which are computationally less demanding [5]. Genetic algorithm (GA) and support vector machines (SVM) have also been adopted for solving portfolio optimization problems [4,11]. Despite these and later enhancements, the Markowitz model still offers the most general work.

While the limits of a quadratic approximation of the utility function are acknowledged, the development of operational procedures for constructing portfolios has been mainly hampered by computational problems. To improve Markowitz's model both computationally and theoretically, Konno and Yamazaki [5] proposed a portfolio optimization model using mean-absolute deviation risk function instead of Markowitz's standard deviation risk function. Their model can tackle with the difficulties associated with the classical Markowitz's model while maintaining its advantages over equilibrium models. In particular, their model can be formulated as a linear program, so that a large-scale of portfolio optimization problem may be easily solved.

Conventional portfolio optimization models have an assumption that the future condition of stock market can be accurately predicted by historical data. However, no matter how accurate the past data is, this premise will not exist in the financial market due to the high volatility of market environment. To deal quantitatively with imprecise information in making decisions, Zadeh [19] introduced the notion of fuzziness. Fuzzy set theory has been extensively employed in linear and nonlinear optimization problems [6-8,13]. Intuitively, when the parameters are fuzzy in the portfolio problem, the gain of return of the portfolio will be fuzzy as well. In this case, the associated portfolio optimization problem is a fuzzy portfolio optimization problem. Ammar and Khalifa [1] introduced the formulation of fuzzy portfolio optimization problem based on Markowitz's mean-variance model. Fei [2] studied the optimal consumption and portfolio choice in a Merton-style model with anticipation when there is a difference between ambiguity and risk. Gupta et al. [3] utilized fuzzy set theory into a semi-absolute deviation portfolio selection model for investors. Vercher [18] employed semi-infinite programming technique to solve the portfolio selection problem with fuzzy returns.

In this paper, we utilize the concepts of mean-absolute deviation function [5] and Zadeh's extension principle [19,20] to develop a solution method for the fuzzy portfolio optimization problem whose future return rates and risk are expressed by fuzzy numbers. We construct a pair of two-level mathematical programming models to derive the upper bound and lower bound of the return from the portfolio at a specific α level. Based on the duality theorem and by applying the variable transformation technique, the pair of two-level mathematical programs is transformed into a pair of ordinary one-level linear programs so they can be manipulated. From different values of α, the membership function of the fuzzy return from the portfolio is constructed. This result should provide the decision maker with more information for making decisions.

2 The Problem

Assume we have n assets for possible investment and are interested in determining the portion of available total fund M_0 that should be invested in each of the assets during the investment periods. Let the decision variables be denoted x_j, j=1,…, n, which represent the dollar amount of fund invested in asset j. We then have the constraints $\sum_{j=1}^{n} x_j = M_0$ and $x_j \geq 0$. Let R_j be a random variable representing the rate of return (per period) of the asset j. The expected return (per period) of the investment is given by

$$r(x_1,...,x_n) = E\left[\sum_{j=1}^{n} R_j x_j\right] = \sum_{j=1}^{n} E\left[R_j\right] x_j \tag{1}$$

Generally, one would diversify the investment portfolio so that funds are invested in several different assets to avoid the investment risk. Markowitz [9,10] employed the standard deviation of the return as the measure of risk.

$$\sigma(x_1,...,x_n) = \sqrt{E\left[\left\{\sum_{j=1}^{n} R_j x_j - E\left[\sum_{j=1}^{n} R_j x_j\right]\right\}^2\right]} \tag{2}$$

A rational investor may be interested in obtaining a certain average return from the portfolio at a minimum risk. Markowitz [9,10] formulated the portfolio problem as a quadratic programming problem:

$$V = \text{Min} \sum_{i=1}^{n} \sum_{j=1}^{n} \sigma_{ij} x_i x_j \tag{3}$$

$$\text{s.t.} \ \sum_{j=1}^{n} x_j = M_0, \ \sum_{j=1}^{n} r_j x_j \geq R_0, \ 0 \leq x_j \leq U_j, \ j=1,.., n.$$

where R_0 is the return in dollars and U_j is the upper bound of the investment in asset j.

For reducing computational burden and transaction/management cost and cut-off effect, Konno and Yamazaki [5] introduced the absolute deviation function instead of Markowitz's standard deviation function of the (per period) earning out of the portfolio.

$$w(x) = E\left[\left|\sum_{j=1}^{n} R_j x_j - E\left[\sum_{j=1}^{n} R_j x_j\right]\right|\right] \tag{4}$$

They proved that the measures of standard deviation and absolute deviation functions are essentially the same if (R_1,…, R_n) are multivariate normally distributed and proposed an alternative portfolio problem

$$V = \text{Min} \ E\left[\left|\sum_{j=1}^{n} R_j x_j - E\left[\sum_{j=1}^{n} R_j x_j\right]\right|\right] \tag{5}$$

$$\text{s.t.} \ \sum_{j=1}^{n} x_j = M_0, \ \sum_{j=1}^{n} E\left[R_j\right] x_j \geq R_0, \ 0 \leq x_j \leq U_j, \ j=1,.., n.$$

Suppose we have historical data on each asset for the past T years, which give the price fluctuations and dividend payments. We can then estimate the return on investment from each asset from past data. Let r_{jt} denote the realization of random variable

R_j during period *t*, i.e., the total return per dollar invested in asset *j* during year *t*. Clearly, the values of r_{jt} are not constants and can fluctuate widely from year to year. In addition, r_{jt} may be positive, negative, or zero. Hence, in order to assess the investment potential of asset *j* per dollar invested, denoted by r_j as

$$r_j = E\left[R_j\right] = \frac{1}{T}\sum_{t=1}^{T} r_{jt} \tag{6}$$

Then $w(x)$ can be approximated as

$$E\left[\left|\sum_{j=1}^{n} R_j x_j - E\left[\sum_{j=1}^{n} R_j x_j\right]\right|\right] = \frac{1}{T}\sum_{t=1}^{T}\left|\sum_{j=1}^{n}(r_{jt} - r_j)x_j\right| \tag{7}$$

That is, Model (5) can be reformulated as follow [5]:

$$V = \text{Min } \sum_{t=1}^{T} u_t \Big/ T \tag{8}$$

$$\begin{aligned} \text{s.t. } & u_t + \sum_{j=1}^{n}(r_{jt} - r_j)x_j \ge 0, \quad t = 1,\dots,T, \\ & u_t - \sum_{j=1}^{n}(r_{jt} - r_j)x_j \ge 0, \quad t = 1,\dots,T, \\ & \sum_{j=1}^{n} x_j = M_0,\ \sum_{j=1}^{n} r_j x_j \ge R_0,\ 0 \le x_j \le U_j,\ j=1,..,n. \end{aligned}$$

Presume that we are able to roughly project the return on investment for each asset over the next several yeas. Since the return on investment in the future is uncertain, it is an imprecise number. We suppose that the r_{jt} and r_j are approximately known and can be represented by the convex fuzzy sets $\tilde{R}_{jt}$ and $\tilde{R}_j$, respectively. Let $\mu_{\tilde{R}_{jt}}$ and $\mu_{\tilde{R}_j}$ denote their membership functions, respectively. We have,

$$\tilde{R}_{jt} = \left\{r_{jt}, \mu_{\tilde{R}_{jt}}(r_{jt}) \mid r_{jt} \in S(\tilde{R}_{jt})\}\right\} \tag{9a}$$

$$\tilde{R}_j = \left\{r_j, \mu_{\tilde{R}_j}(r_j) \mid r_j \in S(\tilde{R}_j)\}\right\}, \tag{9b}$$

where $S(\tilde{R}_{jt})$ and $S(\tilde{R}_j)$ are the supports of $\tilde{R}_{jt}$ and $\tilde{R}_j$, respectively. From (6), we have

$$\tilde{R}_j = \sum_{t=1}^{T} \frac{\tilde{R}_{jt}}{T} \tag{10}$$

When the returns on investment $\tilde{R}_{jt}$ are fuzzy numbers, the parameters $\tilde{R}_j$ and objective value will also be fuzzy number as well. Under such circumstances, the portfolio optimization is formulated as the following linear programming problem with fuzzy parameters:

$$\tilde{V} = \text{Min} \sum_{t=1}^{T} u_t / T \tag{11}$$

$$\text{s.t. } u_t + \sum_{j=1}^{n} (\tilde{R}_{jt} - \tilde{R}_j) x_j \geq 0, \quad t = 1,...,T,$$

$$u_t - \sum_{j=1}^{n} (\tilde{R}_{jt} - \tilde{R}_j) x_j \geq 0, \quad t = 1,...,T,$$

$$\sum_{j=1}^{n} x_j = M_0 \quad \sum_{j=1}^{n} \tilde{R}_j x_j \geq R_0 \quad 0 \leq x_j \leq U_j, \; j=1,.., n.$$

Without loss of generality, all $\tilde{R}_{jt}$ and $\tilde{R}_j$ in (11) are assumed to be convex fuzzy numbers, as crisp values can be represented by degenerated membership functions which only have one value in their domains. Since $\tilde{V}$ is a fuzzy number rather than a crisp number, we apply Zadeh's extension principle [19,20] to transform the problem into a family of traditional linear programs to solve.

Based on the extension principle, the membership function $\mu_{\tilde{V}}$ can be defined as:

$$\mu_{\tilde{V}}(v) = \underset{\mathbf{r}}{\text{SupMin}} \{\mu_{\tilde{R}_{jt}}(r_{jt}), \mu_{\tilde{R}_j}(r_j), \forall j,t \mid v = V(\mathbf{r})\} \tag{12}$$

where $V(\mathbf{r})$ is the function of the crisp portfolio optimization problem that is defined in (8). In Eq. (12), several membership functions are involved. To derive $\mu_{\tilde{V}}$ in closed form is hardly possible. According to (12), $\mu_{\tilde{V}}$ is the minimum of $\mu_{\tilde{R}_{jt}}(r_{jt})$ and $\mu_{\tilde{R}_j}(r_j)$, $\forall j,t$. We need $\mu_{\tilde{R}_{jt}}(r_{jt}) \geq \alpha$, and $\mu_{\tilde{R}_j}(r_j) \geq \alpha$, and at least one $\mu_{\tilde{R}_{jt}}(r_{jt})$ or $\mu_{\tilde{R}_j}(r_j)$, $\forall j,t$, equal to α such that $v = V(\mathbf{r})$ to satisfy $\mu_{\tilde{V}}(v)$=α. To find the membership function $\mu_{\tilde{V}}$, it suffices to find the right shape function and left shape function of $\mu_{\tilde{V}}$, which is equivalent to finding the upper bound of the objective value V_α^U and lower bound of the objective V_α^L at a specific α level. Since V_α^U is the maximum of $V(\mathbf{r})$ and V_α^L is the minimum of $V(\mathbf{r})$, they can be expressed as:

$$V_\alpha^U = \text{Max}\left\{V(\mathbf{r}) \mid (R_{jt})_\alpha^L \leq r_{jt} \leq (R_{jt})_\alpha^U, \; \sum_{t=1}^{T}(R_{jt})_\alpha^L / T \leq r_j \leq \sum_{t=1}^{T}(R_{jt})_\alpha^U / T, \; j = 1,...,n, \; t = 1,...,T\right\} \tag{13}$$

$$V_\alpha^L = \text{Min}\left\{V(\mathbf{r}) \mid (R_{jt})_\alpha^L \leq r_{jt} \leq (R_{jt})_\alpha^U, \; \sum_{t=1}^{T}(R_{jt})_\alpha^L / T \leq r_j \leq \sum_{t=1}^{T}(R_{jt})_\alpha^U / T, \; j = 1,...,n, \; t = 1,...,T\right\} \tag{14}$$

Clearly, different values of r_{jt} and r_j produce different objective value (risk). To find the fuzzy objective value, it suffices to find the upper bound and lower bound of the objective values of (11).

From (13), the values of r_{jt} and r_j that attain the largest value for V_α^U at possibility level α can be determined from the following two-level mathematical programs:

$$V_\alpha^U = \underset{\substack{(R_{jt})_\alpha^L \le r_{jt} \le (R_{jt})_\alpha^U \\ \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} \le r_j \le \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T} \\ \forall j,t}}{\text{Max}} \left\{ \begin{array}{l} \underset{\mathbf{x}}{\text{Min}} \ \sum_{t=1}^{T} u_t / T \\ \text{s.t.} \ u_t + \sum_{j=1}^{n} (r_{jt} - r_j) x_j \ge 0, \quad t = 1,\ldots,T, \\ \qquad u_t - \sum_{j=1}^{n} (r_{jt} - r_j) x_j \ge 0, \quad t = 1,\ldots,T, \\ \qquad \sum_{j=1}^{n} x_j = M_0, \quad \sum_{j=1}^{n} r_j x_j \ge R_0, \ \ 0 \le x_j \le U_j, \ \ j=1,\ldots,n. \end{array} \right. \tag{15}$$

The objective value V_α^U is the upper bound of the objective value for (11). From the derived optimal solution, the associated return value from the portfolio can be calculated.

By the same token, to find the values of r_{jt} and r_j that produce the smallest objective value, a two-level mathematical program is formulated by replacing the outer-level program of (15) from “Max” to “Min”:

$$V_\alpha^L = \underset{\substack{(R_{jt})_\alpha^L \le r_{jt} \le (R_{jt})_\alpha^U \\ \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} \le r_j \le \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T} \\ \forall j,t}}{\text{Min}} \left\{ \begin{array}{l} \underset{\mathbf{x}}{\text{Min}} \ \sum_{t=1}^{T} u_t / T \\ \text{s.t.} \ u_t + \sum_{j=1}^{n} (r_{jt} - r_j) x_j \ge 0, \quad t = 1,\ldots,T, \\ \qquad u_t - \sum_{j=1}^{n} (r_{jt} - r_j) x_j \ge 0, \quad t = 1,\ldots,T, \\ \qquad \sum_{j=1}^{n} x_j = M_0, \quad \sum_{j=1}^{n} r_j x_j \ge R_0, \ \ 0 \le x_j \le U_j, \ \ j=1,\ldots,n. \end{array} \right. \tag{16}$$

where the inner-level program calculates the objective value for each r_{jt} and r_j specified by the outer-level program, while the outer-level program determines the values of r_{jt} and r_j that produces the smallest objective value. The objective value V_α^L is the lower bound of the objective values for (11). Similar to (15), the corresponding return from the portfolio can be calculated from the obtained optimal solution.

3 The Solution Procedure

3.1 Upper Bound V_α^U

From previous discussion, we know that to find the upper bound of the objective value (risk) of (11), it suffices to solve the two-level mathematical program of (15). However, solving (15) is not so straightforward because the outer-level program performs maximization operation, but the inner-level program executes minimization function. The outer-level and inner-level programs have different directions for optimization. The dual problem of the inner-level program of (15) is the following mathematical form:

$$\underset{\mathbf{w},\mathbf{y}}{\text{Max}} \;\; M_0 y_{2T+1} + R_0 y_{2T+2} - \sum_{j=1}^{n} U_j w_j \tag{17}$$

$$\begin{aligned} \text{s.t.} \;\; & \sum_{t=1}^{T} (r_{jt} - r_j) y_t - \sum_{t=1}^{T} (r_{jt} - r_j) y_{T+t} + y_{2T+1} + r_j y_{2T+2} - w_j \le 0, \quad j = 1,\dots,n, \\ & y_t + y_{T+t} \le 1/T, \; t = 1,\dots,T, \\ & y_t, y_{2T+2}, w_j \ge 0, \; t = 1,\dots,2T, \; j = 1,\dots,n, \; y_{2T+1} \text{ unrestriced in sign.} \end{aligned}$$

By duality theorem, if one problem is unbounded, then the other is infeasible. Moreover, if both problems are feasible, then they both have optimal solutions with the same objective value. In other words, Model (15) can be reformulated as

$$V_\alpha^U = \underset{\substack{(R_{jt})_\alpha^L \le r_{jt} \le (R_{jt})_\alpha^U \\ \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} \le r_j \le \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T} \\ \forall j,t}}{\text{Max}} \left\{ \begin{aligned} & \underset{\mathbf{w},\mathbf{y}}{\text{Max}} \;\; M_0 y_{2T+1} + R_0 y_{2T+2} - \sum_{j=1}^{n} U_j w_j \\ & \text{s.t.} \;\; \sum_{t=1}^{T} (r_{jt} - r_j) y_t - \sum_{t=1}^{T} (r_{jt} - r_j) y_{T+t} + y_{2T+1} + r_j y_{2T+2} - w_j \le 0, \\ & \quad j = 1,\dots,n, \\ & \quad y_t + y_{T+t} \le 1/T, \; t = 1,\dots,T, \\ & \quad y_t, y_{2T+2}, w_j \ge 0, \; t = 1,\dots,2T, \; j = 1,\dots,n, \\ & \quad y_{2T+1} \text{ unrestricted.} \end{aligned} \right. \tag{18}$$

Now, both the inner-level program and outer-level program have the same maximization operation, they can be merged into a one-level program with the constraints at the two levels considered at the same time:

$$V_\alpha^U = \underset{\mathbf{w},\mathbf{y}}{\text{Max}} \;\; M_0 y_{2T+1} + R_0 y_{2T+2} - \sum_{j=1}^{n} U_j w_j \tag{19}$$

$$\text{s.t.} \;\; \sum_{t=1}^{T} (r_{jt} - r_j) y_t - \sum_{t=1}^{T} (r_{jt} - r_j) y_{T+t} + y_{2T+1} + r_j y_{2T+2} - w_j \le 0, \; j = 1,\dots,n, \tag{19.1}$$

$$y_t + y_{T+t} \le 1/T, \; t = 1,\dots,T, \tag{19.2}$$

$$(R_{jt})_\alpha^L \le r_{jt} \le (R_{jt})_\alpha^U, \; j = 1,\dots,n, \; t = 1,\dots,T, \tag{19.3}$$

$$\sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} \le r_j \le \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T}, \; j = 1,\dots,n, \tag{19.4}$$

$$y_t, y_{2T+2}, w_j \ge 0, \; t = 1,\dots,2T, \; j = 1,\dots,n, \; y_{2T+1} \text{ unrestriced in sign.}$$

The variable transformation technique can be applied to the nonlinear terms $r_{jt} y_t$, $r_{jt} y_{T+t}$, $r_j y_t$, $r_j y_{T+t}$, and $r_j y_{2T+2}$ in (19.1). One can multiply Constraint (19.3) by y_t and y_{T+t}, j=1,2,…, n, $t = 1,\dots,T$, respectively, and substitute $r_{jt} y_t$ by ξ'_{jt} and $r_{jt} y_{T+t}$ by ξ''_{jt}, respectively. Likewise, Constraint (19.4) can be multiplied, respectively, by y_t and y_{T+t}, $t = 1,\dots,T$, and then we substitute $r_t y_t$ by ψ'_{jt} and $r_j y_{T+t}$ by ψ''_{jt}, respectively. By the same token, we can also multiply (19.4) by y_{2T+2} and replace $r_j y_{2T+2}$ with δ_j.

Via the dual formulation and variable transformation, the two-level mathematical program of (19) is transformed into the following one-level linear program:

$$V_\alpha^U = \underset{\boldsymbol{\xi},\boldsymbol{\psi},\boldsymbol{\delta},\mathbf{w},\mathbf{y}}{\text{Max}} \quad M_0 y_{2T+1} + R_0 y_{2T+2} - \sum_{j=1}^{n} U_j w_j \tag{20}$$

$$\begin{aligned}
\text{s.t. } & \sum_{t=1}^{T} \xi'_{jt} - \sum_{y=1}^{T} \xi''_{jt} - \sum_{t=1}^{T} \psi'_{jt} + \sum_{t=1}^{T} \psi''_{jt} + y_{2T+1} + \delta_j - w_j \le 0, \ j = 1,\dots,n, \\
& y_t + y_{T+t} \le 1/T, \ t = 1,\dots,T, \\
& (R_{jt})_\alpha^L y_t \le \xi'_{jt} \le (R_{jt})_\alpha^U y_t, \ j = 1,\dots,n, \ t = 1,\dots,T, \\
& (R_{jt})_\alpha^L y_{T+t} \le \xi''_{jt} \le (R_{jt})_\alpha^U y_{T+t}, \ j = 1,\dots,n, \ t = 1,\dots,T, \\
& \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} y_t \le \psi'_{jt} \le \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T} y_t, \ j = 1,\dots,n, \ t = 1,\dots,T, \\
& \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} y_{T+t} \le \psi''_{jt} \le \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T} y_{T+t}, \ j = 1,\dots,n, \ t = 1,\dots,T, \\
& \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} y_{2T+2} \le \delta_j \le \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T} y_{2T+2}, \ j = 1,\dots,n, \\
& y_t, y_{2T+2}, w_j, \delta_j \ge 0, \ t = 1,\dots,2T, \ \ j = 1,\dots,n, \ y_{2T+1} \text{ unrestriced in sign.}
\end{aligned}$$

By solving (20), we derive the objective value V_α^U, which is the upper bound of the investment risk, and $r_{jt}^* = \xi_{jt}^{'*} / y_t^* = \xi_{jt}^{''*} / y_{T+t}^*$. From the dual prices of the optimal solution, we can find the primal solutions $\overset{*}{x}_j$, j=1,2,…,n. The associated return from the portfolio is calculated as $R_\alpha^U = \sum_{j=1}^{n} r_j^* x_j^* = \sum_{j=1}^{n} \sum_{t=1}^{T} r_{jt}^* x_j^* / T$.

An essential idea in finance and economics is that the greater the amount of risk that an investor is willing to take on, the greater the potential return. Since we obtain the upper bound of the risk V_α^U, the calculated value R_α^U would be the upper bound of the return from the portfolio at a specific α-level. In other words, for risk lovers, they may consider obtaining higher returns by taking higher risk as the objective.

3.2 Lower Bound V_α^L

Since both the inner-level program and outer-level program of (16) have the same minimization operation, they can be combined into a conventional one-level program with the constraints of the two programs considered at the same time.

$$V_\alpha^L = \underset{\mathbf{x}}{\text{Min}} \ \ V = \sum_{t=1}^{T} u_t / T \tag{21}$$

$$\text{s.t. } \ u_t + \sum_{j=1}^{n} r_{jt} x_j - \sum_{j=1}^{n} r_j x_j \ge 0, \quad t = 1,\dots,T, \tag{21.1}$$

$$u_t - \sum_{j=1}^{n} r_{jt} x_j + \sum_{j=1}^{n} r_j x_j \ge 0, \quad t = 1,\dots,T, \tag{21.2}$$

$$\sum_{j=1}^{n} x_j = M_0,\ \sum_{j=1}^{n} r_j x_j \geq R_0,\ 0 \leq x_j \leq U_j,\ j=1,\ldots,n. \tag{21.3}$$

$$(R_{jt})_\alpha^L \leq r_{jt} \leq (R_{jt})_\alpha^U,\ j=1,\ldots,n,\ t=1,\ldots,T, \tag{21.4}$$

$$\sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} \leq r_j \leq \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T},\ j=1,\ldots,n, \tag{21.5}$$

Similar to (19), we can apply the variable transformation technique to the nonlinear terms $r_{jt}x_j$ and $r_j x_j$ in (21.1), (21.2), and (21.4). One can multiply (21.5) and (21.6) by x_j, j=1,2,…, n, and substitute $r_{jt}x_j$ by ρ_{jt} and $r_j x_j$ by η_j, respectively. Consequently, we have the flowing conventional linear program:

$$V_\alpha^L = \underset{\mathbf{x}}{\text{Min}}\ V = \sum_{t=1}^{T} u_t / T \tag{22}$$

$$\begin{aligned}
\text{s.t. } & u_t + \sum_{j=1}^{n} \rho_{jt} - \sum_{j=1}^{n} \eta_j \geq 0, \quad t=1,\ldots,T, \\
& u_t - \sum_{j=1}^{n} \rho_{jt} + \sum_{j=1}^{n} \eta_j \geq 0, \quad t=1,\ldots,T, \\
& \sum_{j=1}^{n} x_j = M_0,\ \sum_{j=1}^{n} \eta_j \geq R_0,\ 0 \leq x_j \leq U_j,\ j=1,\ldots,n. \\
& (R_{jt})_\alpha^L x_j \leq \rho_{jt} \leq (R_{jt})_\alpha^U x_j,\ j=1,\ldots,n,\ t=1,\ldots,T, \\
& \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^L}{T} x_j \leq \eta_j \leq \sum_{t=1}^{T} \frac{(R_{jt})_\alpha^U}{T} x_j,\ j=1,\ldots,n,
\end{aligned}$$

The derived objective value V_α^L is the lower bound of the investment risk, and $r_{jt}^* = \rho_{jt}^* / x_j^*$. Also, we can obtain the corresponding return from the portfolio by calculating $R_\alpha^L = \sum_{j=1}^{n} r_j^* x_j^* = \sum_{j=1}^{n} \sum_{t=1}^{T} r_{jt}^* x_j^* / T = \sum_{j=1}^{n} \sum_{t=1}^{T} \rho_{jt}^* / T$. Similar to the previous discussion in Section 3.1, the value R_α^L is the lower bound of the return from the portfolio. For risk-averse investors, they may consider lower risk as the objective but also obtain lower returns. Together with R_α^U derived from (20), R_α^L and R_α^U constitute the fuzzy interval of the return from the portfolio described in (11).

4 Conclusion

Financial investments are especially important for individual and business financial managers because of the low interest rate. Portfolio optimization has been one of the important fields of research in economics and finance. Since the prospective returns of the assets used for portfolio optimization problem are forecasted values, considerable uncertainty is involved. This paper proposes a solution method for the uncertain portfolio optimization problem whose imprecise parameters are expressed by fuzzy numbers. The idea is based on mean-absolute deviation function [5] and the extension principle [19,20] to transform the fuzzy portfolio optimization problem into a pair of mathematical programs. Based on the duality theorem and by applying the variable transformation technique, the pair of two-level mathematical programs is transformed into a pair of ordinary one-level linear programs so they can be manipulated. By enumerating different α values, the upper and lower bounds of the á-cuts of the fuzzy risk and return values are calculated. Those derived values can thus be used to approximate the membership functions of the risks and return from the portfolio.

Interestingly, the calculated results conform to an essential idea in finance and economics that the greater the amount of risk that an investor is willing to take on, the greater the potential return. Investors who emphasize the gain of the investment return may consider the maximization of returns as the objective by taking higher risk. While risk-averse investors may pursue that minimizing risk is more important than maximizing investment return, and may consider the minimization of risk as the objective.

Acknowledgment

This research was supported by the National Science Council of Republic of China under Contract NSC98-2416-H-238-004-MY2.

References

1. Ammar, E., Khalifa, H.A.: Fuzzy Portfolio Optimization: A Quadratic Programming Approach. Chaos, Solutions & Fractals 18, 1045–1054 (2003)
2. Fei, W.: Optimum Consumption and Portfolio Choice with Ambiguity and Anticipation. Information Sciences 177, 5178–5190 (2007)
3. Gupta, P., Mehlawat, M.K., Saxena, A.: Asset Portfolio Optimization Using Fuzzy Mathematical Programming. Information Sciences 178, 1754–1755 (2008)
4. Ince, H., Trafalis, T.B.: Kernel Methods for Short-Term Portfolio Management. Expert Systems with Applications 30, 535–542 (2006)
5. Konno, H., Yamazaki, H.: Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market. Management Science 37, 519–531 (1991)
6. Liu, S.T.: Geometric Programming with Fuzzy Parameters in Engineering Optimization. International Journal of Approximate Reasoning 46, 17–27 (2007)
7. Liu, S.T.: Quadratic Programming with Fuzzy Parameters: A Membership Function Approach. Chaos, Solutions & Fractals 40, 237–245 (2009)
8. Liu, S.T., Kao, C.: Solving Fuzzy Transportation Problems Based on Extension Principle. European Journal of Operational Research 153, 661–674 (2004)
9. Markowitz, H.: Portfolio Selection: Efficient Diversification of Investment. John Wiley & Sons, New York (1959)
10. Markowitz, H.: Portfolio Selection. Journal of Finance 7, 77–91 (1952)
11. Oh, K.J., Kim, T.Y., Min, S.: Using Genetic Algorithm to Support Portfolio Optimization for Index Fund Management. Expert Systems with Applications 28, 371–379 (2005)
12. Perold, A.: Large Scale Portfolio Selections. Management Science 30, 1143–1160 (1984)
13. Sakawa, M.: Fuzzy Sets and Interactive Multiobjective Optimization. Plenum Press, New York (1993)
14. Sharpe, W.F.: Capital Asset Prices: A Theory of Market Equilibrium under Conditions of Risk. Journal of Finance 19, 425–442 (1964)
15. Sharpe, W.F.: A Linear Programming Algorithm for a Mutual Fund Portfolio Selection. Management Science 13, 499–510 (1967)

16. Sharpe, W.F.: A Linear Programming Approximation for the General Portfolio Selection Problem. Journal of Finance and Quantitative Analysis 6, 1263–1275 (1971)
17. Stone, B.: A Linear Programming Formulation of the General Portfolio Selection Model. Journal of Finance and Quantitative Analysis 8, 621–636 (1973)
18. Vercher, E.: Portfolios with Fuzzy Returns: Selection Strategies based on Semi-Infinite Programming. Journal of Computational & Applied Mathematics 217, 381–393 (2008)
19. Zadeh, L.A.: Fuzzy Sets as a Basis for a theory of Possibility. Fuzzy Sets and Systems 1, 3–28 (1978)
20. Zimmermann, H.J.: Fuzzy Set Theory and Its Applications. Kluwer, Nijhoff (1996)

Knowledge-Based Framework for Workflow Modelling: Application to the Furniture Industry[*]

Juan Carlos Vidal, Manuel Lama, Alberto Bugarín, and Manuel Mucientes

Department of Electronics and Computer Science
University of Santiago de Compostela
E15782 Santiago de Compostela, Spain
{juan.vidal,manuel.lama,alberto.bugarin.diz,manuel.mucientes}@usc.es

Abstract. In this paper, we describe a framework for integrating workflow modelling techniques with a knowledge management approach that enables us to represent the problem-solving knowledge, the coordination of the method execution, and the agents that are involved in the workflow. The case of price estimation workflow in the manufactured furniture industry is described within this framework, that was implemented in a knowledge-based system that is based on a service-oriented architecture that automated the correct execution of the methods needed to solve the estimation task.

1 Introduction

Business process management (BPM) technology [1] allows the explicit representation of the business process logic in a process-oriented view, and is increasingly used as a solution to integrate engineering/manufacturing processes. However, current BPM modelling approaches do not *explicitly* incorporate the problem-solving knowledge in the workflow definition: this knowledge is implicitly used both in its control and organizational structure, but as it is not explicitly represented, it cannot be shared or reused. For dealing with this drawback, a new framework that models workflows at the knowledge-level has been defined [2]: this framework incorporates both the control structure and the participants of a workflow as *two new knowledge components*. The structure of these new components is based on two ontologies: the High-Level Petri Nets ontology [3] and an ontology for process representation and organization modeling.

This new workflow knowledge-based modelling framework has been applied to the price estimation of products manufacturing in custom-furniture industry. This process is strategic since it helps to keep competitive prices, increment productivity, and reduce costs. In the case of many custom-furniture industries this need

[*] This work was supported by the Spanish Ministry of Science and Innovation under grants TIN2008-00040 and TSI2007-65677C02-02. M. Mucientes is supported by the Ramón y Cajal program of the Spanish Ministry of Science and Innovation.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 175–184, 2010.

is even higher due to the large number of processes that are made following non-standardized manual procedures. A solution in this case can only be approached by means of an automation of the price estimation process that involves great volume of information to characterize the furniture, the knowledge management provided by the company experts related to the manufacturing process, and the coordination between many resources (human and software) that execute activities with different perspectives of the process and in different times.

In this paper, we present a workflow that describes how the price estimation task has been solved in the domain of custom-furniture industry. For it, we have used a framework [2] that integrates workflow-modelling techniques with a knowledge management approach. This new framework enables us to represent *(i)* the problem-solving knowledge through a set of methods that have been selected from the CommonKADS library [4], and *(ii)* the coordination of the method execution, and *(iii)* the agents that participate in the workflow. This workflow has been implemented in a knowledge-based system called SEEPIM that is based on a service-oriented architecture that automated the correct execution of the methods needed to solve the main task.

The paper is structured as follows: in section 2 a brief description of the framework for representing workflows is presented, in section 3 the workflow that models the price estimation task is described, in section 4 the main benefits achieved with the applciation of the new workflow framework are discussed in detail. Section 5 presents the most relevant conclusions of the paper.

2 Knowledge-Based Workflow Framework

The workflow for price estimation of custom-designed furniture presented in this paper has been developed within the knowledge-based system SEEPIM, which aims to model workflows as reusable knowledge components. SEEPIM implements a conceptual framework [2] which facilitates the modelling of workflows as problem-solving methods (PSMs) [5]. Three types of knowledge components are usually referred in the bibliography of PSMs:

- *Tasks* describe a problem to solve but not its solution. They describe the required input/output parameters, the pre- and postconditions, and the task objectives. This description does not include information about how it may be solved.
- *Methods* detail the reasoning process to achieve a task, that is, the way to solve a problem. Methods are described by functional properties and can be split into *non-composite* and *composite* methods depending on whether the method can be *decomposed* into subtasks or not.
- *Domain models* describe the knowledge of a domain of discourse, that is, the facts, rules and axioms that are applied to the domain.

These knowledge components are described independently at the knowledge level. Therefore, the same component might be reused in multiple problems, and this reuse is achieved through *adapters* which define a binary relation between

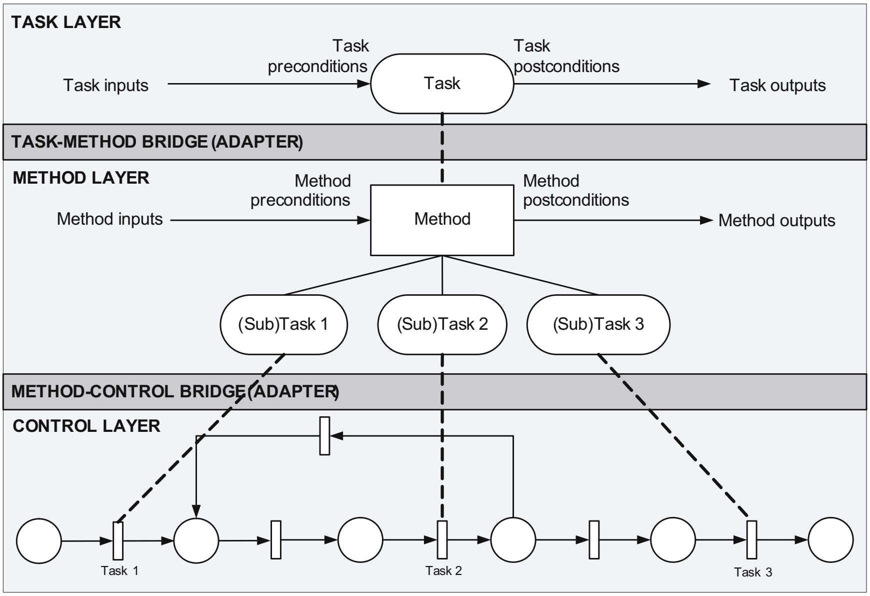


Fig. 1. Task decomposition diagram

two knowledge components. However, it is not possible to model workflows with only these three components, because workflows have two additional features that should be enhanced in the modelling framework: the *process structure* and the *participants*. Therefore two new components are added to the framework:

- *Control component*. Workflow processes are designed with control-flow structures that cannot be modelled with traditional languages of PSMs. Workflows need a language able to deal with parallelization, choice, split, merge or synchronization patterns at least. From the approaches proposed to describe these patterns we selected High-Level Petri Nets (HLPN) [3] because *(i)* they are based on a mathematical formalism and therefore have no ambiguity, *(ii)* they have a graphical representation which facilitates their use and understandability, and *(iii)* they have proven to be one of the most powerful approach to represent the worklfow control [1]. The *control component* is represented through a hierarchical HLPN ontology, and it models the operational description of a composite method; that is, it describes how the execution of non-composite methods is coordinated to solve the task solved by the composite method.
- *Resource component*. Participants play an important role in the workflow specification. Workflow activities are represented in the framework by means of the non-composite methods and will be performed by a (either human or software) agent in the organization. The resource model classifies the organization resources and relates the participants to the activities they may perform.

Within this framework, a workflow consists of a task (problem) to be solved, the method that solves this task, the control of that method, the resources that are involved in its execution, and the domains of discourse of the workflow. For example, Figure 1 depicts the first step of a task decomposition diagram in which a task is solved by a composite method that is split into three (sub)tasks (1, 2 and 3). In this figure, the *task-method adapter* indicates which method will solve the task, and the *method-control adapter* selects the control-flow of this method. The other steps of the diagram indicate how the (sub)tasks of the method are solved in turn by other methods and how these last have associated a control structure.

3 Price Estimation in Custom-Designed Furniture Industry

As an application of the framework described in the previous section we go on to describe here the most relevant aspects of workflow that models the price estimation task in the furniture industry. Other details of context, implementation and validation can be found in [6,7]. The estimation task is a good example for showing the capabilities of the framework since, as it is shown in what follows, it involves a high number of subtasks where heterogeneous sources of different types of knowledge are involved.

Figure 2 shows the first level of decomposition of the solution proposed for the price estimation task. Firstly, using a conceptual model, the method that solves the estimation gets the product description to be manufactured. Based on this description the processing times are estimated using a set of fuzzy rules (TSK type) that had been previously learned with an evolutionary algorithm. This information will be used for planning the workload of the company. Then a *repeat-while* loop is executed. In this loop different technical designs of the product are proposed until the most suitable is found. In each loop iteration the result of the design is evaluated, taking into account the client needs, the technical features of the product, the manufacturing processes necessary to perform them, and the business productivity criteria. For example, at this stage it is decided whether we need to simplify parts of the furniture or look for semi-finished components that can be bought directly without having to manufacture them. The method followed to solve the product design task is called *propose and revise* [4], and is typically used to model constraint satisfaction problems.

3.1 Estimate Processing Times

The resolution of the task for the estimation of processing times is not trivial and depends on the conceptual design and the type of planning to be carried out. Sometimes, it is not possible to obtain a detailed design of the product to be manufactured and therefore to infer the exact operations to be performed is not possible. This situation occurs in the first meetings with the clients for product definition or when they do not have a precise idea of the product they need.

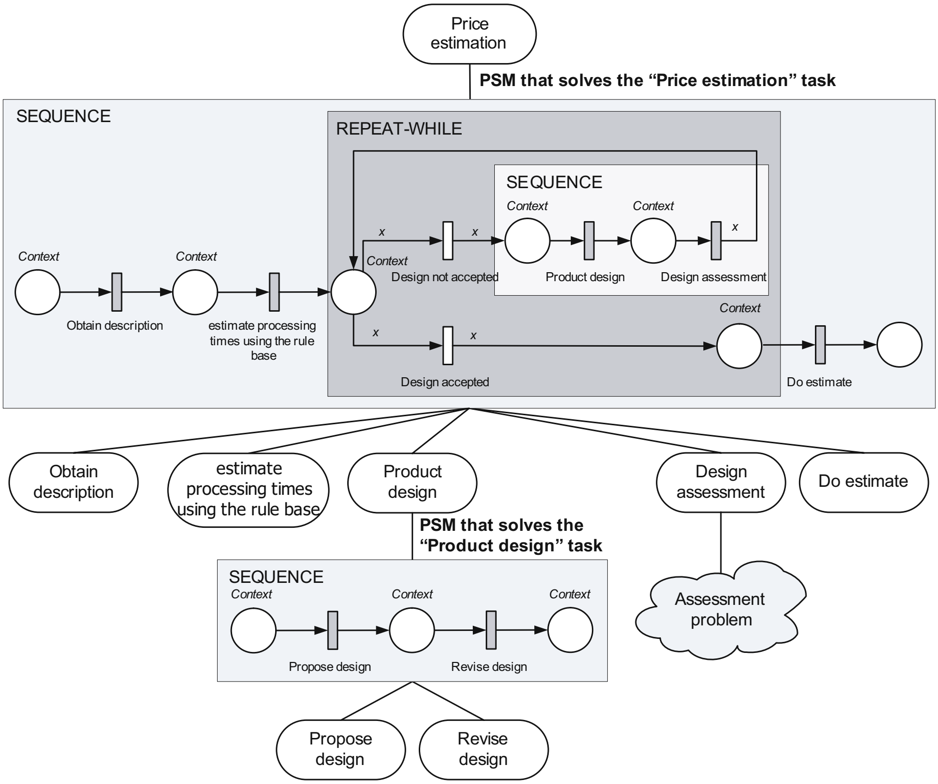


Fig. 2. *Price estimation* task decomposition

However, sometimes the conceptual design can be converted into CAD designs and the operations required to manufacture the product can be extracted in a precise way. In order to address both situations the implementation of the method that solves this task has been performed using a set of TSK fuzzy rules that were obtained using an evolutionary algorithm, which is described in [6].

3.2 Product Design

Figure 3 shows the method that solves the task (*propose design*), in which the technical designs and the final cost of the product are obtained. This task is within the structure *repeat-while* of main workflow and it will be executed while the criteria for completion are not verified.

The first step to solve this task is to create the product design in a CAD format with the aim of *(i)* obtaining the dimensions and materials of the product components, and *(ii)* determining which manufacturing processes will be applied to each of these pieces. To make this design the designers will use standard components and materials, and develop a modular design trying to reduce the cost of piece manufacturing while guaranteeing the robustness of the design.

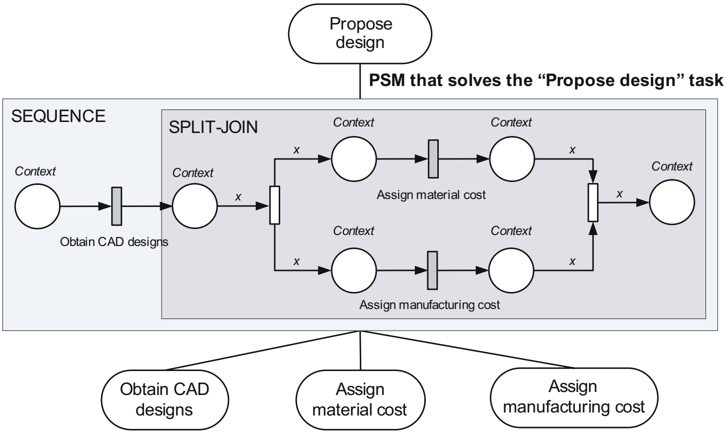


Fig. 3. *Propose design* task decomposition

For example, the product cost depends on the type of joint used to assemble the furniture: some unions require the assembly of the furniture previous to finishing, thus increasing the cost of manufacturing, packing and shipping.

The second step consists of determining the final cost of the product as the sum of the materials cost and the pieces manufacturing cost. These two tasks are performed in parallel and in the following sections we describe how they are solved.

3.3 Assign Material Cost

The task *assign material cost* is responsible for assigning prices to the materials used in the estimation taking into account *(i)* the price and quality of each of the materials used in product design, and *(ii)* the reliability of the suppliers to guarantee that those materials are delivered at the agreed times. The proposed solution to solve this task is shown in Figure 4 and it adapts the general class of CommonKADS library methods known as *propose and revise* to the characteristics of both the domain and the task to be solved. Thus:

- The task *propose material cost* is responsible for selecting the most appropriate price for each material of the product. This cost depends on the direct cost of the material, the cost of the additives (varnish, glues, etc.), and the cost of the wasted material, and it is based on a set of rules/criteria extracted from the company experts in purchases.
- The task *revise material cost* is responsible for verifying the correctness of the proposed price for each material and for modifying the price when necessary. This revision is based on information provided by suppliers about their sales prices, and it checks among other things that the offered price is updated and remains in force at the time of manufacture.

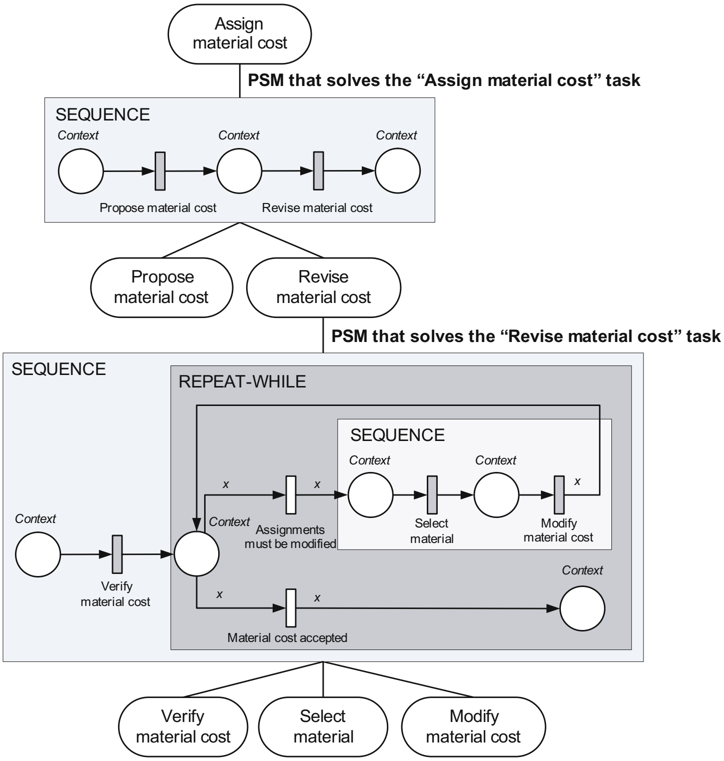


Fig. 4. *Assign material cost* task decomposition

3.4 Assign Manufacturing Cost

The method that solves this task belongs to the general class of methods known as *propose and review*, which is applied to the generation of workplans. Thus, the method that solves the task *propose workplan* executes a *repeat-while* loop until the workplan has been accepted. The loop body consists of a sequence of three tasks. The first task, *Set resource agenda*, is responsible for setting the work schedule of the manufacturing plant. This task takes the work schedule established for each resource and allows to apply changes to allocate/deallocate the resource, assign overtime or include a new manufacturing turn.

The second task, *Set workplan*, is responsible for generating a plan to allocate a work to the resources involved in the product development. This task is necessary to decide whether the plant could support the new workload with the deadline required by the customer, whether a part of the manufacturing is subcontracted, etc. The method that solves this task consists of a sequence of tasks:

- In the first task orders to manufacture are selected. The resource planning is a computationally expensive task and, therefore, the greater the number of orders that are planned, the greater the calculations to be performed and

possibly the worse the results because the search space is of combinatorial size.
- In the second task orders are ranked according to their selected delivery date, its priority, and the client. These criteria rank the orders to perform and will influence the planning. For example, if the product is key to the company, it may be included with a higher priority.
- The third task obtains the workplan, assigning the work to the resources according to a predefined timetable. The resolution of this kind of problem is computationally complex and requires the use of a *scheduling* method whose implementation has been made following an evolutionary approach, which is described in [7].
- In the fourth task the manufacturing director selects the better plan from the set of solutions proposed by the system.

Finally, the task *Revise workplan* analyses the workplan to look for problems: division of labour, overloads, logistic problems, availability of raw materials, etc. From this analysis, manufacturing directors assess whether planning is accepted as is or a new planning is requested.

4 Framework Evaluation

The developed KBS contributed with important improvements in the price-estimation task of the furniture company where the system was introduced. Table 1 contains the average error of time estimations for some machine centers at the start of the project. Some of these estimations have an error greater than 40% which is totally unacceptable in any industry. It must be emphasized that this was not the average error of price estimates: a company with such error ratio would not survive in a market as competitive as furniture industry. The error in price estimates was less than 10% as over and under estimations compensate each other. As result of the automation process we obtained a reduction of the mean error to less than 5% for each machine of the production plant. However, a purely numerical analysis does not reveal the actual extent of the improvement. In this sense, the procedure to achieve the solution and the framework that has been established are as important as the workflow model described here because they led the organization to:

- Restructuring the business process (task, method and control models of the framework).
- Restructuring the organization itself (resource model of the framework).
- A detailed analysis of the business model and of the expert knowledge necessary to perform the product design task (knowledge model of the framework).
- The standarization of materials and the use of methodologies like *Design for Manufacturing and Assembly* for standarizing the processes of the production plant (also in the knowledge model of the framework).

Table 1. Mean error per machine center of the processing time

Machine center	Mean error (%)
Wood preparation	-5,0
Horizontal sawing machines	-8,0
Vertical sawing machines	-22,0
Planer	+11,4
Edge banding machines	-14,5
Sanding machines	-5,5
Edge sanding machines	-33,5
Boring machines	-18,2
Tenoning machines	-40,2
CNC Machine	+7,6
Finger jointing	-26,9
Hand finishing	-15,8
Presses and equipment for gluing	+10,6
Robots for handling, feeding and palleting	+31,0
Packaging equipment for panels	-16,0
Pre-assembly lines for furniture parts	-22,5
...	...

This first phase of the WF modeling already contributed with many benefits. The main one was a major improvement in the processing time estimations: 70% of the error reduction occurred during this phase and the 30% remaining has also a strong dependence on this phase since the variables that influence the processing time of each machine were also identified in this phase. In any case, the automation of this task provided many benefits, some of them intangible from the perspective of the time estimation but of great importance from the organizational point of view:

- *Increased work efficiency.* The work is assigned considering both the agent as the product to design. The work allocation takes into account the role and security permissions of the agent as well as his skills and knowledge to perform the task. In addition, dynamic criteria such as workload and agent work history are also taken into account.
- *Reduction of effort and greater specialization.* The work allocation policy streamlined the effort of agents. In the original procedure tasks were assigned only taking the availability of agents into account. This approach was often not the most appropriate since not all the agents have the most suitable knowledge for a work and consequently the results *(i)* may be of low quality or *(ii)* may take too long.
- *Better use of experts.* The furniture price estimation is a very complex task that usually require the continued involvement of the experts which, due to this effort, neglect their work on the production plant. The acquisition of manufacturing knowledge resulted in a better use of these experts who currently work only in monitoring tasks.

- *Improved scheduling* [7]. In most industries the workload of the factory also affects the calculation of price estimates. It is therefore important to have a tool that facilitates the inclusion of the client orders to estimate in the production schedule and calculate its impact.
- *Continued improvement of processing times* [6]. The processing time estimation was one of the main problems of the original system.In part because each of the experts estimated the time based on their experience but also because the manufacturing of custom made furniture has particular characteristics that make accurate estimations particularly difficult to obtain. To reduce the error of these estimations, the calculation of the processing times was automated by means of a machine learning system.

5 Conclusions

The framework here described allows us to integrate work-flow modelling techniques with a knowledge management approach, enabling us to represent the problem-solving knowledge using methods form the CommonKADS library, the coordination of the methods execution and the agents involved in the workflow. The relevant case of price estimation workflow in the furniture manufacturing industry is described within the framework. Validation of a system implementing this workflow in a real environment show a major improvement in this particular task and also at the organisational level.

References

1. van der Aalst, W.M.P., van Hee, K.: Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge (2004)
2. Vidal, J.C., Lama, M., Bugarín, A.: A Framework for Unifying Problem-Solving Knowledge and Workflow Modelling. In: AAAI 2008 Spring Symposium: AI Meets Business Rules and Process Management, San Francisco, EEUU, pp. 105–110. AAAI Press, Menlo Park (2008)
3. Vidal, J.C., Lama, M., Bugarín, A.: A High-level Petri Net Ontology Compatible with PNML. Petri Net Newsletter 71, 11–23 (2006)
4. Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., de Velde, W.V., Wierlinga, B.: Knowledge Engineering and Management: The CommonKADS Methodology. MIT Press, Cambridge (1999)
5. Benjamins, V.R.: On a Role of Problem Solving Methods in Knowledge Acquisition. In: A Future for Knowledge Acquisition, 8th European Knowledge Acquisition Workshop, EKAW'94, London, UK, pp. 137–157. Springer, Heidelberg (1994)
6. Mucientes, M., Vidal, J.C., Bugarín, A., Lama, M.: Processing Times Estimation by Variable Structure TSK Rules Learned through genetic programming. Soft Computing 13(5), 497–509 (2009)
7. Vidal, J.C., Mucientes, M., Bugarín, A., Lama, M., Balay, R.S.: Hybrid Approach for Machine Scheduling Optimization in Custom Furniture Industry. In: 8th International Conference on Hybrid Intelligent Systems, Barcelona, Spain, pp. 849–854. IEEE Computer Society Press, Los Alamitos (2008)

Decision Trees in Stock Market Analysis: Construction and Validation

Margaret Miró-Julià, Gabriel Fiol-Roig, and Andreu Pere Isern-Deyà

Math and Computer Science Department,
University of the Balearic Islands
07122 Palma de Mallorca, Spain
{margaret.miro,biel.fiol}@uib.es

Abstract. Data Mining techniques and Artificial Intelligence strategies can be used to solve problems in the stock market field. Most people consider the stock market erratic and unpredictable since the movement in the stock exchange depends on capital gains and losses. Nevertheless, patterns that allow the prediction of some movements can be found and studied. In this sense, stock market analysis uses different automatic techniques and strategies that trigger buying and selling orders depending on different decision making algorithms. In this paper different investment strategies that predict future stock exchanges are studied and evaluated. Firstly, data mining approaches are used to evaluate past stock prices and acquire useful knowledge through the calculation of financial indicators. Transformed data are then classified using decision trees obtained through the application of Artificial Intelligence strategies. Finally, the different decision trees are analyzed and evaluated, showing accuracy rates and emphasizing total profit associated to capital gains.

Keywords: Data Mining, Decision Trees, Artificial Intelligence, Stock Market Analysis.

1 The Problem

The problem considered in this paper deals with the application of decision making techniques to stock market analysis [1]. The core of the problem is the generation of automatic buying and selling orders in the erratic stock market. The decision making system can be conceived from a data mining point of view [2]. In this sense, the system's design has three main phases as indicated in Figure 1: the data processing phase, the data mining phase and the evaluation phase. The main elements of the system are the original data base, the Object Attribute Table of transformed or processed data, the patterns or models of classified data and the knowledge extracted from the data.

The data processing phase selects from the raw data base a data set that focuses on a subset of attributes or variables on which knowledge discovery has to be performed. It also removes outliers and redundant information, and uses financial indicators to represent the processed data by means of an Object Attribute Table (OAT). The data mining phase converts the data contained in the OAT into useful patterns, in

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 185–194, 2010.

particular decision trees are found [3]. The evaluation phase proves the consistency of the pattern by means of a testing set. The positively evaluated decision system can then be used in real world situations that will allow for its validation.

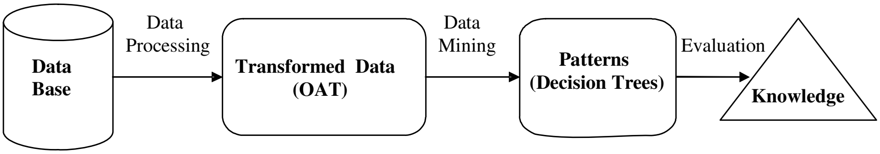


Fig. 1. The Decision Making System

The original data base is formed by financial information available in different web sites. In particular, data from the American stock market have been used. Out of all the variables originally considered the target data is formed by: date (D), opening price (O), closing price (C), daily high (H), daily low (L) and daily volume (V).

In order to obtain a decision making system based on classification techniques a particular company must be selected. This company must satisfy some restrictions: it must not be a technological security, in order to avoid using data from year 2000 economic bubble; it must not be a financial institution, due to the erratic behaviour of the past years; it must have a significant influence within the American stock market; it must be a reference company. These restrictions are appropriate to initiate the process however they might be reconsidered at a later point.

Due to these restrictions and after a careful study of the stock market, Alcoa was selected to complete the project. Alcoa is the world's leading producer and manager of primary aluminum, fabricated aluminum and alumina facilities, and is active in all major aspects of the industry. The targeted data includes daily information from the beginning of 1995 till the end of 2008.

2 The Object Attribute Table

The Data Processing phase transforms the targeted data into useful data written in terms of financial indicators [4]. The financial indicators considered are calculated using graphical methods (where price oscillation is studied) and mathematical methods (where mathematical equations are applied).

2.1 Financial Indicators

The *Simple Moving Average (SMA)* is the unweighted mean of the previous n data points. It is commonly used with time series data to smooth out short-term fluctuations and highlight longer-term trends and is often used for the analysis of financial data such as stock prices (P), returns or trading volumes. There are various popular values for n, like 10 days (short term), 30 days (intermediate term), 70 days (long term) or 200 days depending on the number of days considered.

$$SMA_{n+1}(P) = \frac{P_n + P_{n-1} + \cdots + P_1}{n} \cdot \tag{1}$$

The moving averages are interpreted as support in a rising market or resistance in a falling market. The *SMA* treats all data points equally and can be disproportionally influenced by old data points. The exponential moving average addresses this point by giving extra weight to more recent data points.

The *Exponential Moving Average (EMA)* applies weighting factors which decreases exponentially, giving much more importance to recent observations while not discarding older observations totally. The degree of weighting decrease is expressed as a constant smoothing factor α, which can be expressed in terms of n

$$\alpha = \frac{2}{n+1} \cdot \tag{2}$$

And the exponential moving average is expressed as:

$$EMA_{n+1}(P) = \alpha P_n + (1-\alpha) EMA_n(P) \cdot \tag{3}$$

The moving averages are adequate indicators to determine tendencies. When a short term moving average crosses over a longer term moving average we have a rising tendency. Whereas when a short term moving average crosses under a longer term moving average we have a falling tendency. The moving averages have a drawback since the information provided has a time lag of several days. This can be avoided by using more powerful indicators.

The *Moving Average Convergence Divergence (MACD)* is a combination of two exponential moving averages with different number of data points (days). The *MACD* is calculated by subtracting the y-day *EMA* from the x-day *EMA*. A z-day *EMA* of the *MACD*, called the “signal line”, is plotted on top of the *MACD*, functioning as a trigger for buy and sell signals. Variables x, y, and z are parameters of the *MACD*, usually values x = 12, y = 26 and z = 9 are considered.

The *MACD* is a trend-following momentum indicator that shows the relationship between two moving averages of prices. Its two-folded interpretation is as follows. If the *MACD* line and signal line are considered, when the *MACD* falls below the signal line, it may be time to sell. Conversely, when the *MACD* rises above the signal line, the price of the asset is likely to experience upward momentum, it may be time to buy.

The *stochastic (K)* is an indicator that finds the range between an asset’s high (H) and low price (L) during a given period of time, typically 5 sessions (1 week) or 20 sessions (1 month). The current securities price at closing (C) is then expressed as a percentage of this range with 0% indicating the bottom of the range and 100% indicating the upper limits of the range over the time period covered.

$$K = 100 \frac{C - L}{H - L} \cdot \tag{4}$$

An interesting interpretation for the stochastic can be obtained if the moving average associated with the stochastic *SMA(K)* is considered: the crossing points between *K* and *SMA(K)* trigger buy and sell signals. When *K* crosses over *SMA(K)* it may be time to buy, conversely when *K* crosses under *SMA(K)* selling might be convenient.

The *Bollinger Bands (BB)* consist in a set of three curves drawn in relation to security prices. The middle band M_B is a measure of the intermediate-term trend, usually a moving average (simple or exponential) over n periods. This middle band serves as the base to construct the upper U_B and lower L_B bands. The interval between the upper and lower bands and the middle band is determined by volatility, typically the interval is Q times the standard deviation of the same data that were used to calculate the moving average. The default parameters most commonly used are n = 20 periods and Q = 2 standard deviations.

95% of security prices can be found within the Bollinger Bands, the band represents areas of support and resistance when the market shows no tendencies. When the bands lie close together a period of low volatility in stock price is indicated. When they are far apart a period of high volatility in price is indicated.

Generally Bollinger Bands are used together with other indicators to reinforce its validity. However, on its own they can trigger buy and sell signals. When prices touch the lower band it might be convenient to buy, conversely when prices touch the upper band selling might be convenient.

2.2 Processing of Financial Indicators: Transformed Data

In order to construct the OAT of adequate data, three steps must be followed:

- Processing of the targeted data in order to generate financial indicators.
- Manipulation and discretization of financial indicators into binary attributes.
- Construction of the OAT [5, 6].

The financial indicators are calculated using R, a language for statistical computing and graphics [7]. R is applied mainly as a statistical package but it can be used in other areas of science such as numerical analysis, signal processing, computer graphics and so on because of the number of growing implemented libraries in R to these scientific areas. In particular, the financial indicators are calculated using R's TTR library. R is a popular programming language used by a growing number of data analysts inside corporations and academia. See [8] for details on the use of R in academia and other institutions. R is available as Free Software under the terms of the Free Software Foundation's GNU General Public in source code form. It compiles and runs on a wide variety of UNIX platforms and similar systems (including FreeBSDband Linux, Windows and MacOS).

The indicators considered are the following: *MACD*, *EMA(C)*, *EMA(V)*, *K* and *BB*. They are calculated directly from the 3514 rows that form the daily targeted data. The resulting values are added as new columns in the targeted data table. The added columns are represented by means of eight binary attributes A1, A2, A3, A4, A5, A6, A7 and A8 in the following manner.

- The *MACD* indicator identifies buying (B) or selling (S) orders depending on its crossing with an intermediate term moving average *EMA(z)* called "signal line". If *MACD* crosses under the "signal line" a buying signal (B) is generated. On the other hand, if *MACD* crosses over the "signal line" a selling signal (S) is generated. If there is no crossing no signal is generated. These results are represented using 2 binary attributes, A1 and A2, as follows: when crossing occurs, if *MACD* > *EMA(z)* then A1 = 1; else if *MACD* < *EMA(z)* then A2 = 1. If there is no crossing (no signal is generated) then A1 = 0 and A2 = 0.

- The *EMA(C)* indicator represents the closing price's exponential moving average. When compared to the stock's actual price (*P*) two zones related with the strength-weakness of the asset are generated. These zones can be represented by a binary attribute A3 as follows: if $P > EMA(C)$ (strength) then A3 = 1, else if $P \leq EMA(C)$ (weakness) then A3 = 0.
- The *EMA(V)* indicator represents the volume's exponential moving average. When compared to the actual volume (*V*) two zones related with the trustworthiness of the asset are generated. These zones can be represented by a binary attribute A4 as follows: if $V > EMA(V)$ (trustworthy) then A4 = 1, else if $V \leq EMA(V)$ (non trustworthy) then A4 = 0.
- The *K* indicator also identifies buying (B) or selling (S) orders depending on the crossing of moving averages, in particular of *K* and *SMA(K)*. If *K* crosses under *SMA(K)* a selling signal (S) is generated. On the other hand, if *K* crosses over the *SMA(K)* a buying signal (B) is generated. If there is no crossing no signal is generated. These results are represented using 2 binary attributes, A5 and A6, as follows: when crossing occurs, if $K < SMA(K)$ then A5 = 1; else if $K > SMA(K)$ then A6 = 1. If there is no crossing (no signal is generated) A5 = 0 and A6 = 0.
- The *BB* indicator generates 4 zones which represent areas of support and resistance. When compared with the asset's actual price *P*, these 4 zones can be represented by two binary attributes A7 and A8 as follows: if $P < L_B$ (weak asset), then A7 = 0 and A8 = 0; if $L_B \leq P < M_{B,}$ then A7 = 0 and A8 = 1; if $M_B \leq P < U_B$, then A7 = 1 and A8 = 0; if $U_B \leq P$ (strong asset), then A7 = 1 and A8 = 1.

The resulting table must be filtered. First of all, the original targeted data can be eliminated since they are only used to calculated financial indicators, which constitute the relevant data. Secondly, repeated and irrelevant rows are eliminated. The resulting table, the final Object Attribute Table, is formed by 8 columns of binary attributes, A1, A2, …, A8 and 61 rows as illustrated in Table 1.

Table 1. The Object Attribute Table with 3 classes

A1	A2	A3	A4	A5	A6	A7	A8	class
0	0	0	0	1	0	1	0	S
0	1	0	0	0	0	0	1	B
0	0	1	0	0	1	0	1	N
0	0	0	1	0	1	1	0	N
1	0	1	0	0	0	0	1	S
1	0	1	1	0	0	1	0	S
…	…	…	…	…	…	…	…	…

The next step is to assign the class to each of the remaining rows taking into account the values of the attributes as indicated by the financial expert. Three different classes are used depending on whether buying (B), selling (S) or no action (N) must be taken, as shown in the last column of Table 1.

In order to better evaluate the results of this paper, a second OAT with only two classes, buying (B) and selling (S) has also been considered. In this second OAT the no action class (N) has been eliminated and replaced by B and S classes according to the expert's criteria. Table 2 shows the 2-class OAT.

Table 2. The Object Attribute Table with 2 classes

A1	A2	A3	A4	A5	A6	A7	A8	class
0	0	0	0	1	0	1	0	S
0	1	0	0	0	0	0	1	B
0	0	1	0	0	1	0	1	S
0	0	0	1	0	1	1	0	B
1	0	1	0	0	0	0	1	S
1	0	1	1	0	0	1	0	S
…	…	…	…	…	…	…	…	…

The similarities between Table 1 and Table 2 are evident, since the only changes are the replacement of the N class in Table 1 by a B or S class in Table 2. The results obtained using the 3-class OAT is consistent with a more conservative behavior, whereas the 2-class OAT offers a risky strategy.

3 The Decision Trees

In the data mining phase, the data contained in the OAT is converted into useful patterns. In particular, a decision tree will be obtained. The decision tree is one of the most popular classification algorithms. A decision tree can be viewed as a graphical representation of a reasoning process and represents a sequence of decisions, interrelationships between alternatives and possible outcomes. There are different trees available depending on the evaluation functions considered. In this project, the ID3 algorithm is used due to its simplicity and speed [9]. ID3 is based on information theory to determine the most informative attribute at each step of the process. In this sense, it uses information gain to measure how well a given attribute separates the training examples into the classes. The ID3 algorithm is applied to the OATs represented in both Table 1 and Table 2.

The decision tree for 3 classes, see Figure 2, has a total of 37 nodes, 18 corresponding to the 8 attributes used and 19 leaf nodes or branches. The average branch length is 6. Attributes A2, A1 and A5 appear at the root of the tree suggesting the importance of indicators *MACD* and *K* in the classifying process. This is in accordance with the experts that consider *MACD* and *K* as key indicators in the buying-selling decisions, whereas *EMA(C)*, *EMA(V)* and *BB* are used as reinforcement indicators.

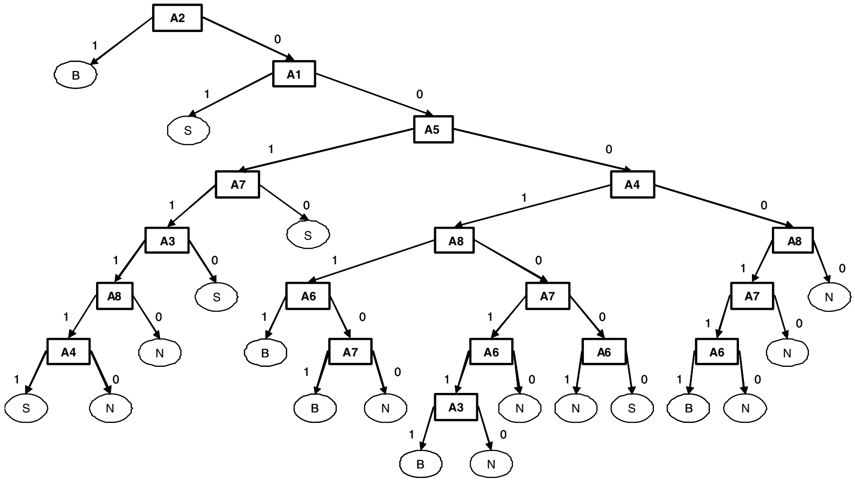


Fig. 2. The Decision Tree for 3 classes

Similarly, the 2-class decision tree, see Figure 3, has a total of 51 nodes, 25 corresponding to the inner nodes and 26 leaf nodes or branches. The average branch length is also 6. Note that when data is classified in 2 classes, the number of borderline data has increased and a larger number of attributes is needed to perform the classification. Attribute A2 appears at the root of the tree suggesting the importance of indicator *MACD* in the classifying process. Attribute A6 also appears at upper levels of the tree indicating the importance of *K* as key indicator in the buying-selling decisions.

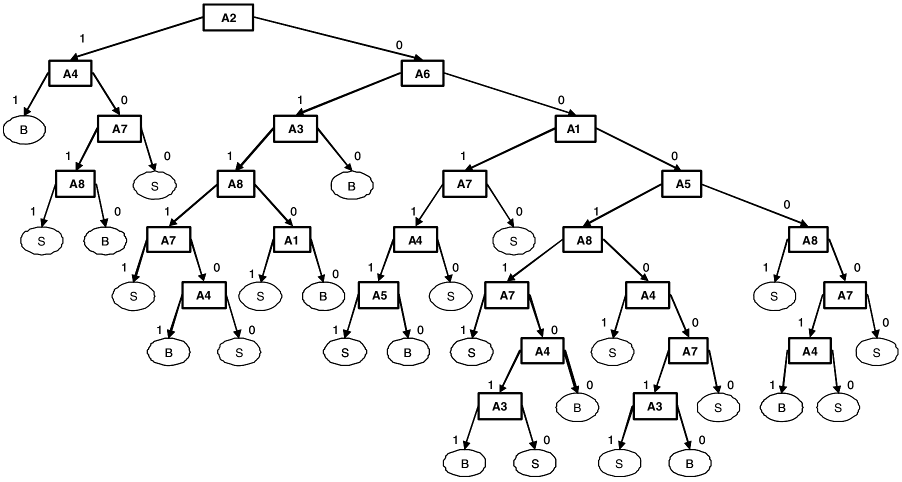


Fig. 3. The Decision Tree for 2 classes

4 Validation of the Decision Trees

The evaluation phase proves the consistency of the ID3 decision tree by means of a testing set. The testing set is formed by the stock market prices of a different asset, of similar characteristics to the one used to train the tree. This raw data are transformed into a testing OAT using the previous indicators and attributes. Now the decision tree is applied to the testing OAT and all instances are classified. The classes obtained with the tree are compared with the classes corresponding to the instances of the testing OAT as suggested by the expert.

The evaluation of the method provides the following results: a) for the 3-class decision tree, Instances = 289; Variables = 8; Correctly classified instances = 113; Incorrectly classified instances = 176; b) for the 2-class tree, Instances = 613; Variables = 8; Correctly classified instances = 300; Incorrectly classified instances = 313. These results are shown on Table 3.

Note that the number of classified instances varied due to the elimination of the N class in the 2-class tree.

Table 3. Evaluacion of the testing set

Parameters	3-class tree	2-class tree
Correct decision	113	300
Incorrect decision	176	313
Correct decision (%)	39.10	48.93
Incorrect decision (%)	60.90	51.06

These results might seem poor. However, stock market analysis is an unpredictable field and even though accuracy is desired, profit associated to capital gain is crucial. Therefore, classification accuracy is not the only parameter that must be evaluated.

The decision tree's behavior respect to profit will be evaluated by an automatic system. This system evaluates the testing set taking into account the following:

- It considers 30-day indicators;
- The system buys only when so indicated by the decision tree;
- The bought stock package is kept until the tree generates a selling order;
- All buying orders generated between the first buying order and the selling order are ignored;
- Once the selling order is generated by the tree, the package will be sold and all selling orders generated before the next buying order will be ignored.

Figure 4 illustrates the process carried out by the automatic system. Point 1 indicates a buying order; Point 2, also a buying order, will be ignored; Point 3 indicates a selling order; Point 4 corresponds to a buying order; Points 5 and 6 (buying orders) will be ignored; Point 7, indicates a selling order and Point 8 (selling order) will be ignored.

The system's performance can be evaluated taking into account the following parameters:

- Total win (%) if correct decision is made;
- Total loss (%) if incorrect decision is made;

- Average win (%) if correct decision is made;
- Average loss (%) if incorrect decision is made;
- Performance (%);
- *Win/Loss* ratio

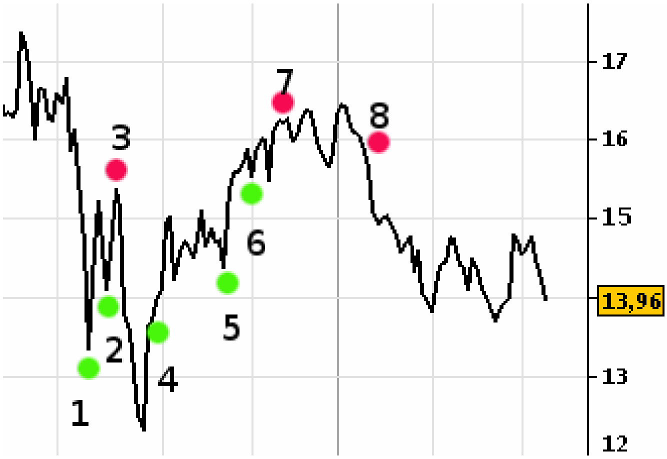


Fig. 4. Automatic System's Behaviour

The method provides the results shown on Table 4, the total yield for the 3-class decision tree is of 118.15%, whereas for the 2-class decision tree the yield is 87.89% which are acceptable results.

Table 4. Performance of the Automatic System

Parameters	3-class tree	2-class tree
Win when correct decision (%)	579.71	610.00
Loss when incorrect decision (%)	-461.56	-522.12
Average win when correct decision (%)	5.13	2.03
Average loss when incorrect decision (%)	-2.62	-1.67
Performance (%)	118.15	87.89
W/L ratio	1.26	1.17

5 Conclusions and Future Work

The stock market is an unpredictable, erratic and changeable domain. Nevertheless, tools that predict the stock market's behaviour exist. This paper considers a method based on financial indicators to generate decision trees that classify buying-selling orders. A decision tree in Artificial Intelligence is equivalent to an automatic investment system in the stock market analysis. The procedure developed here is long and costly, but once the tree is generated, it's automatic and applicable to any financial asset.

The initial raw data are daily stock market values of Alcoa, these data are transformed into an OAT and used as a training set to construct decision trees by means of the ID3 algorithm. Decision trees with 3 and 2 classes have been generated. These decision trees have been tested and evaluated. The results for the 3-class tree show a 39% accuracy percentage and a 118% profit gain, whereas for the 2-class tree the results show a 49% accuracy and a 88% profit gain. These results are considered satisfactory. If both trees are compared, the 3-class tree has a lower percentage accuracy but the profit gain is higher.

This paper offers a limited vision of one of the many solutions available. The following aspects can be considered as future work. Other different financial indicators may offer better results and should be studied, also other decision trees can be obtained using different learning algorithms and should be evaluated. The stock market is a dynamic world, and the decision tree could present time-related disturbances. Therefore, an automatic system using a threshold should be considered in order to avoid strong losing trends.

Acknowledgments. The authors want to thank Damià Vaquer, Gerard Draper and Jaume Jaume for their helpful comments and suggestions.

References

1. Fiol-Roig, G., Miró-Julià, M.: Applying Data Mining Techniques to Stock Market Analysis. Accepted for publication in the 8th International Conference on Practical Applications of Agents and Multi-Agent Systems. Trends and Strategies on Agents and Multiagent Systems (2010)
2. Fayyed, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence, AI Magazine Fall 96, 37–54 (1996)
3. Fiol-Roig, G.: UIB-IK: A Computer System for Decision Trees Induction. In: Raś, Z.W., Skowron, A. (eds.) ISMIS 1999. LNCS, vol. 1609, pp. 601–611. Springer, Heidelberg (1999)
4. Weinstein, S.: Stan's Weinstein's Secrets For Profiting in Bull and Bear Markets. McGraw-Hill, New York (1988)
5. Miró-Julià, M.: Knowledge discovery in databases using multivalued array algebra. In: Moreno-Díaz, R., Pichler, F., Quesada-Arencibia, A. (eds.) EUROCAST 2009. LNCS, vol. 5717, pp. 17–24. Springer, Heidelberg (2009)
6. Fiol-Roig, G.: Learning from Incompletely Specified Object Attribute Tables with Continuous Attributes. Frontiers in Artificial Intelligence and Applications 113, 145–152 (2004)
7. The R project, `http://www.r-project.org/`
8. `http://www.nytimes.com/2009/01/07/technology/business-computing/07program.html`
9. Quinlan, J.R.: Induction of decision trees. Machine Learning 1, 81–106 (1986)

An Intelligent System for Gathering Rates of Local Taxes on the Web

Matteo Cristani and Nicoletta Gabrielli

Department of Computer Science, University of Verona
Strada Le Grazie 15, 37134 Verona
matteo.cristani@univr.it,
nicoletta.gabrielli@univr.it

Abstract. We describe a technology that is used to gather information about the Italian Real Estate Property Tax. The documents are employed for generating records that populate a database in a semi-automatic way, This database is used afterwards for assistance in tax payment.

1 Introduction

In many countries of the Western world municipalities and other local authorities are permitted by the national laws to impose taxes that very often apply in relation to some features of the house where the citizen lives or that she possesses.

In Italy, the municipalities apply a special tax, called ICI (Imposta Comunale sugli Immobili – Municipality Real Estate Tax) that is a Property Tax for every real estate (land on which it is permitted to raise buildings, civil use buildings). Every citizen who possesses a real estate has to pay a percentage, fixed by the law to a maximum of 0.9%, of the value of the Real Estate.

Many exemptions exist, and recently one major exemption has been introduced that imposes the ICI only to those citizens who possesses more than one estate, and relatively to the real estates not used as principal dwelling. The rates, exemptions, reductions and application cases are defined year by year and the municipality council produces a decision every year that is condensed in two main documents: the *tax bylaw* and the *tax code*. The tax bylaw emitted by the local authorities and the tax code are, rather frequently, the major documents in which data about the rates, the application conditions and the rules regarding allowed reductions for citizens with given characteristics, or estates with a given specific destination. Many municipalities in Italy publish on the web these documents. Sometimes the first one is in form of a summary.

1.1 Industrial Motivation

The request of studying this problem has been initiated by the a company that manages Tax assistance for the Italian Trade Unions. Local service provision of tax assistance are named CAAF (Centro Autorizzato di Assistenza

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 195–204, 2010.

Fiscale – Authorised Centre for Tax Assistance) and one of the national trade unions drives a national service called CAAF CISL (CISL = Confederazione Italiana Sociale del Lavoro – Social Italian Confederation of Labour)[1].

The production of these documents is related to the tax deadlines, being the major for ICI in June. As a consequence, the *tax season*, in relation to property taxes, starts in February and ends in May. During the tax season, the local offices of CAAF CISL operated to collect the documents of the surrounding area (the county). This typically requires a few weeks of intense activities, since the documents are typically produced by the local authorities between February and April[2]. The collected documents are interpreted by the employees devoted to this specific activity and put into a database, that is, depending on the local office, either a local copy of a database shared by the offices of a county, or a common database of the offices of one county. Data are then passed onto two different tax management software tools that make use of them for tax payment assistance.

Difficulties are everywhere, but the major issue is that the collection of data is geographically limited, for obvious reasons of effectiveness, and the usage of these data should instead nationwide. In fact, consider a citizen that has one house in the city centre and one in the countryside, in a different county. The responsibility to put the data in the tax payment assistance applications is fully in the hands of the CAAF CISL local offices of the two counties, but the assistance is performed in a single office, with the obvious effects of misalignment of the data in temporal terms, and the possibility of extraordinary work effort required for synchronising the different data. At the moment in which we started the project, no synchronisation took place in the procedural model. In the example above data of the two houses for that particular taxpayer are inserted by the operator.

The above described activities are clearly error prone, and also need a lot of coordination effort that is time-consuming and ineffective in terms of information system. Since the system does not explicitly support the coordination, the errors are spread-out.

1.2 Lack of Technological Solutions

In the phase in which we started defining the solution we investigate the open source and commercial opportunities for supporting the gathering activity and discovered that no actual solution existed at the moment. Indeed, the substitution of the individuals in the performance of the activity sketched above is not easy and requires a lot of functions that need research.

The documents need to be collected, and then interpreted, and finally transformed onto data for a database. The documents are written in natural language

[1] When non confusion arises we refer to the Authorised Centre for Tax Assistance of the Social Italian Confederation of Labour either with the acronym CAAF CISL or with the generic expression Organisation.

[2] As a matter of fact, for several admissible reasons, municipalities can produce these documents also later, and in some exceptional cases, even much later. We encountered municipalities that produced the official documents in October.

(Italian, for the great majority, apart from the documents of the South Tirol area, that are written in German), and often the layout, the legal formula, and even the format of the tax rates are presented in variants. The collection of the documents per se is a complex task, since not all municipalities publish those documents, and the available sources are differently considered by the domain experts.

1.3 Summary of the Paper

In Section 2 we specifically deal with the description of the system requirements, being this analysis the base for the development of the application. It becomes clear in this analysis that the major techniques we need to compose are statistical natural language processing, deep web methods, and interoperability techniques. Subsection 2.1 goes onto the details of this aspects, whilst Subsection 2.2 illustrates the change impact upon the organisation. In Section 3 we provide some experimental results regarding the system performance and the system usage and compare them against the complexity of the problem to be dealt with. In particular Subsection 3.1 summarises these statistical results, and Subsection 3.2 aims at generalising the approach in order to employ it in the practice of some other cases. Section 4 surveys current literature upon the employed techniques, and finally Section 5 takes some conclusions, and sketches further work.

2 Methodological Issues

Building an application for gathering data from the web is not a novel task in the current literature of the applications of Artificial Intelligence to the Engineering domain. This has been, for instance, achieved in [4], and in a few other papers. However, to the best of our knowledge, no other investigation has been yet published that focuses upon taxes. The focus generates a specific mixup of needs, that in turn, drives to a specific usage of techniques that is novel in the literature and can be employed, as we propose in Section 3, to define an architecture proposal that can be employed in many other cases, some of which are exemplified in Section 5. The nature of the problem is explicitly described in Subsection 2.1.

As we detail deeper in the rest of the paper, many aspects of the implementation of the application discussed here, depend upon the knowledge that is added to the system. This knowledge derives from the analysis of the competences of domain experts. We indeed used the knowledge of the operators who serve the Italian trade unions in gathering the information when the system was not in operation. The knowledge added to the system is described in Figure 1.

2.1 From Legal Documents to Database Records

The cycle that is employed by humans for going from legal documents to records in a database is described in Figure 2.

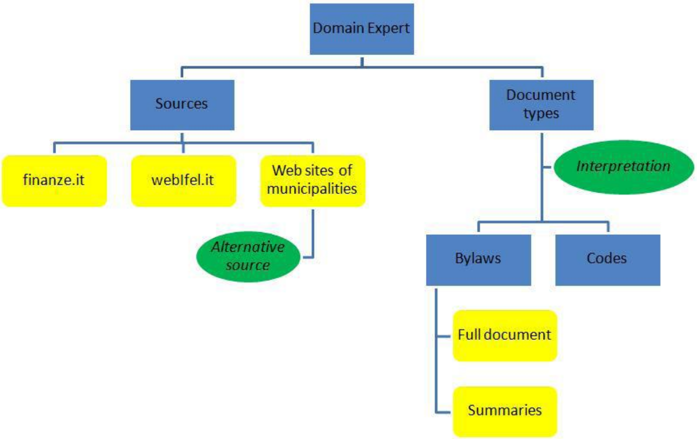


Fig. 1. The domain expert knowledge incorporated in the system

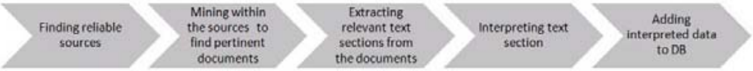

Finding reliable sources
Mining within the sources to find pertinent documents
Extracting relevant text sections from the documents
Interpreting text section
Adding interpreted data to DB

Fig. 2. The cycle used by humans to insert data about real estate property tax rates

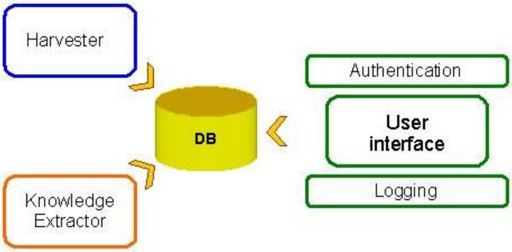


Fig. 3. The general working schema of ICIWEB

This cycle, whose condensation into the application is explained below in this section, gives birth to the application schema which guided the entire implementation, and that is pictorially represented in Figure 3.

Note that the knowledge defined in figure 1 is also the knowledge passed to the new employees who enter the organisation for working in the local tax area.

Finding reliable sources. The major problem of the schema defined above is to find a set of accessible and reliable sources where looking for documents that contain real estate property tax rates. As sketched in the introduction, the law in Italy allows municipalities to impose taxes on real estate property. Each municipality issues two documents per year, the real estate property tax bylaw and the real estate property tax code.

The bylaw contains the institution of the tax and the articulated definition of the rates mainly stated in form of *Regular, Reduced, Principal dwelling.* The same document, moreover, includes the reductions, that can be assigned to citizens under certain conditions.

The tax code, instead, contains the rules to apply rates and reductions. In particular, the determination of the above mentioned conditions, and the rules to evaluate these conditions for the citizens.

The major sources used by the domain experts resulted to be the two sites `finanze.it` and `webifel.it`, both publicly accessible. This two web sites publish regularly the documents mentioned above. However, the level of update and frequency may be substantially variable during the tax season. In case the document would not be available in the moment in which the operator retains it to be necessary, the regular approach is to look for the documents in the web sites of the municipalities. In the near future, when the municipalities will be all publishing timely documents on the web, an approach that look within these sites is likely to be successful.

The selection of the reliable sources is, as the above reasoning confirms, an off-line operation. Once we have defined the right sources the index can be used forever.

Last but not least aspect is the identification of a valid method to parse web sites of the municipalities. The Italian Association of Municipality Administration, a politically neutral organisation whose sole purpose is to provide services to the municipalities and stimulate the central authorities in order of the needs of the single municipalities, has a complete repository of the municipality web sites. We used this to construct the reliable sources list.

Deep web search and relevance analysis. Once we enter in a web site to look for documents treating the theme we use a technology that is able to discriminate between those documents that may contain some information about real estate property taxes and those documents that *undoubtedly* contain exact up-to-date data for these taxes. Referred to each year we have one single bylaw, and one single code. Although it is easy to determine a method for computing in an efficient way pertinent documents, and a method that is very precise in terms of pertinence, an algorithm that is simultaneously precise and efficient is difficult to find. Clearly, if we have to be less precise than desirable, an algorithm has to make only false positive cases, preferably than negative ones. In particular, then, we prefer the situation in which the algorithm selects more than one single document candidate to be the bylaw[3].

The deep web techniques [3] are useful when the data we look for are not indexed explicitly somewhere in the home page or to a page that is directly accessible from the home page. This is strongly enhanced by the dynamic generation of web pages, that makes these kinds of effort very hard.

[3] Often, the web sites `finanze.it` and `webifel.it`, and sometimes also the municipalities, publish bylaw summaries more than full documents. Sometimes this can generate conflicts.

Knowledge extraction techniques. At the end of the cycle the harvesting component works for extracting the data to be used for feeding the data base of tax data. This is summarised in Figure 4.

Data extraction consists of two activities: finding the sections in which the data are settled, and transforming that text part in a string that is processed by a special module that converts it onto data that are compatible with the database formats used in this context. We recurred to an open source architecture, namely Gate, that is used for the first activity and plugged-in for the second one. For a precise reference to the ideas underlying Gate, look at [2].

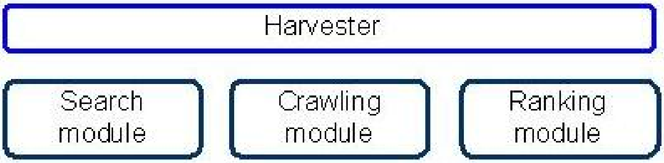


Fig. 4. The working schema of the harvesting component of ICIWEB

2.2 Novelties in the Database Population Procedure

The procedure used by the employees of CAAF CISL for tax payment assistance does get involved in the change management due to the introduction of the new application. In particular when a taxpayer comes to pay in a local office of CAAF CISL in that instant, the entire set of admissible payment values is found. This does not happen anymore. In fact, the system we implemented gathers all the information nationwide and populates the database that only needs to be validated by the operators offline with respect to the insertion of the data of taxpayers. The cycle of gathering we defined in Figure 2 is still valid. Clearly, step 1 is performed once for all, and the other ones are instead cycled until the full set of rates and reductions is found. There are two major issues to clarify, and then we consider the application illustrated in this sense: (a) the interaction between the repository of rates and reductions and the applications for tax payment form filling assistance, and (b) the activities performed by the operators when using the system.

Towards interoperable solutions. As mentioned above, one of the major obstacle in realising the application we describe in this paper, consists in determining a correct correspondence between the application output and the software tools used in assisting taxpayers. The alignment is difficult for two main reasons:

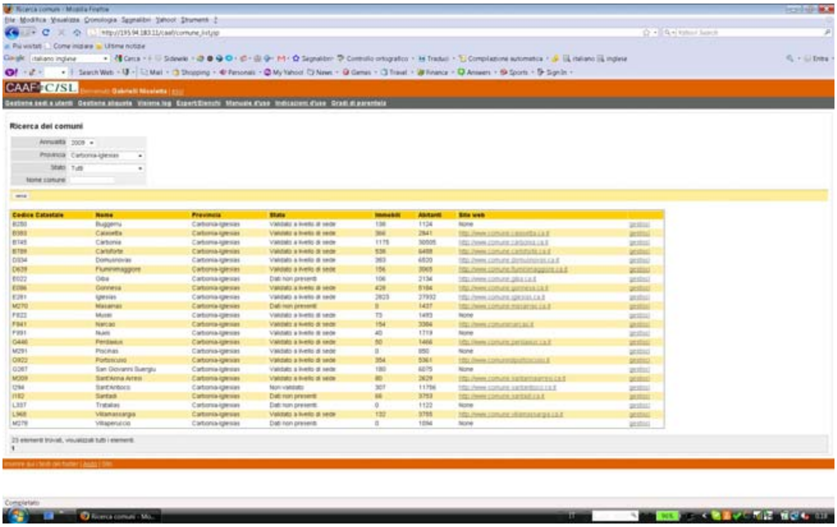
- Different local office are interested in maintaining different applications, mainly because this does not force operators in learning new applications and new procedures for data insertions;
- A new application can be introduced that has not a standard solution for the data.

Looking at the solution we adopted, that is based on the general principle of modulation, it is rather easy to understand that the above defined requisites can be matched in any further extensions to the application. This leads to a flexible architecture, as sketched here.

A decision support system approach. The application which has been implemented is a typical Decision Support system, in which the following cycle takes place:

- The system runs and populates automatically the repository and consequently the database;
- The users with the correct privileges access the system and look at the database table containing the tax rates and reductions. At this stage they
  - Validate the rates and reductions that are complete, making the correct values available nationwide;
  - Modify the records that contain incomplete and incorrect data;
  - Add data that are missing.
- The operators in the national system of CAAF CISL make use of validated data to assist taxpayers.

The screenshot in Figure 5 presents the application in the phase in which an user selects the municipalities in a county and the screenshot in Figure 6 shows an user when looking at the records related to a given municipality.



Ricerca dei comuni

Annualità	2009
Provincia	Carbonia-Iglesias
Stato	Tutti
Nome comune	

Codice Catastale	Nome	Provincia	Stato	Immobili	Abitanti	Sito web	
B250	Buggerru	Carbonia-Iglesias	Validato a livello di sede	138	1124	None	gestisci
B383	Calasetta	Carbonia-Iglesias	Validato a livello di sede	366	2841	http://www.comune.calasetta.ca.it	gestisci
B745	Carbonia	Carbonia-Iglesias	Validato a livello di sede	1175	30505	http://www.comune.carbonia.ca.it	gestisci
B789	Carloforte	Carbonia-Iglesias	Validato a livello di sede	538	6488	http://www.comune.carloforte.ca.it	gestisci
D334	Domusnovas	Carbonia-Iglesias	Validato a livello di sede	363	6620	http://www.comune.domusnovas.ca.it	gestisci
D639	Fluminimaggiore	Carbonia-Iglesias	Validato a livello di sede	156	3065	http://www.comune.fluminimaggiore.ca.it	gestisci
E022	Giba	Carbonia-Iglesias	Dati non presenti	106	2134	http://www.comune.giba.ca.it	gestisci
E086	Gonnesa	Carbonia-Iglesias	Validato a livello di sede	428	5184	http://www.comune.gonnesa.ca.it	gestisci
E281	Iglesias	Carbonia-Iglesias	Validato a livello di sede	2823	27932	http://www.comune.iglesias.ca.it	gestisci
M270	Masainas	Carbonia-Iglesias	Dati non presenti	8	1437	http://www.comune.masainas.ca.it	gestisci
F822	Musei	Carbonia-Iglesias	Validato a livello di sede	73	1493	None	gestisci
F841	Narcao	Carbonia-Iglesias	Validato a livello di sede	154	3384	http://www.comunenarcao.it	gestisci
F991	Nuxis	Carbonia-Iglesias	Validato a livello di sede	40	1719	None	gestisci
G446	Perdaxius	Carbonia-Iglesias	Validato a livello di sede	50	1466	http://www.comune.perdaxius.ca.it	gestisci
M291	Piscinas	Carbonia-Iglesias	Validato a livello di sede	0	850	None	gestisci
G922	Portoscuso	Carbonia-Iglesias	Validato a livello di sede	354	5361	http://www.comunediportoscuso.it	gestisci
G287	San Giovanni Suergiu	Carbonia-Iglesias	Validato a livello di sede	180	6075	None	gestisci
M209	Sant'Anna Arresi	Carbonia-Iglesias	Validato a livello di sede	80	2629	http://www.comune.santannaarresi.ca.it	gestisci
I294	Sant'Antioco	Carbonia-Iglesias	Non validato	307	11756	http://www.comune.santantioco.ca.it	gestisci
I182	Santadi	Carbonia-Iglesias	Dati non presenti	66	3753	http://www.comune.santadi.ca.it	gestisci
L337	Tratalias	Carbonia-Iglesias	Dati non presenti	0	1122	None	gestisci
L968	Villamassargia	Carbonia-Iglesias	Validato a livello di sede	132	3785	http://www.comune.villamassargia.ca.it	gestisci
M278	Villaperuccio	Carbonia-Iglesias	Dati non presenti	0	1094	None	gestisci

23 elementi trovati, visualizzati tutti i elementi.
1

Completato

Fig. 5. A screenshot of the application: municipalities in a county

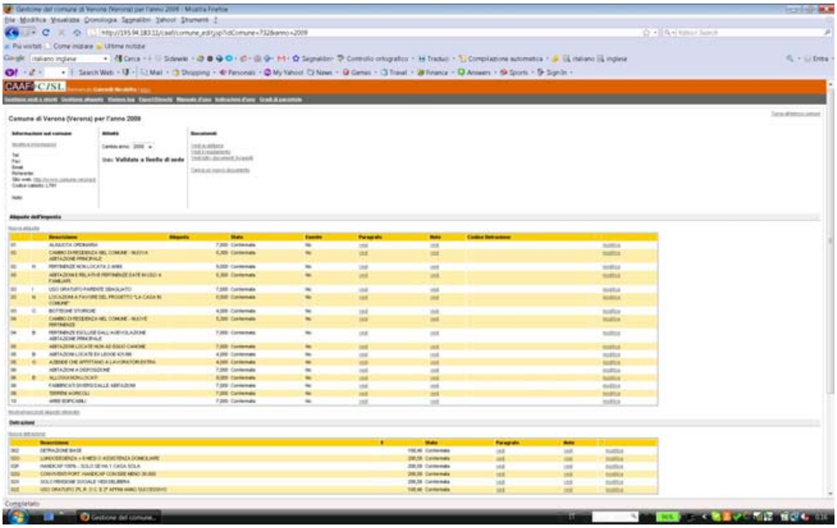


Fig. 6. A screenshot of the application: the system operates on a single municipalities

3 From a Case Study to an Architecture Proposal

The architecture of the system is presented in Figure 3 and has been realised within the project on an experimental basis. Currently the system operates on a limited number of sample local offices, and makes data available nationwide to the local offices in Italy. Data you can look at, and those that are described below are all coming from the analysis of 2009 tax season.

3.1 Evidences

The municipalities in Italy are 8101. The system ran on the web sites of the ministry of Finance (`finanze.it`), on the web site of the ANCI (Association of Italian Municipalities) Trust and on the web sites of the municipalities on the official list of ANCI.

We have performed two distinct analyses, one in June, when the inserted data were covering about the 70% of the total and in November, when the coverage was almost complete.

- The municipalities automatically populated by the system have been, in June, 4398. The municipalities whose data were available were roughly 35% more. Major reasons for this difference are two: (a) unaccessible data, due to use of DRF[4] by the web site, or external link to other sites containing data.
- In November we obtained 5654 correct bylaws, 3636 correct tax codes and a total of 15006 documents retrieved. In total the municipalities for which at least one document was found were 5656.

[4] Data Request Form.

- The total number of inhabitants of Italy is 56675667. The coverage shown above is 44346990, namely 78.25% of the whole population.

The quality of the extraction can be evaluated during the second experimental phase more in detail, since we need to understand how many data need to be corrected, added and integrated to make such an evaluation and this has not yet been completed. However, we can certainly say that the percentage of validation is very high on the municipalities with at least one document, being close to 80% for the sample local offices.

3.2 Generalising the Approach

The approach adopted here to perform gathering and analysis of data can be employed in many other cases, non only for taxes. One rather obvious usage is in collecting information about bylaws for scientific purposes, or for the use of practitioners.

In general, the architecture presented in Figure 3 can be used for information gathering with legal documents.

4 Related Work and Employed Methods

The basic issues related to the usage of the application presented in this paper concern the usage of ontology-driven classification of documents, as, for instance, done in [4,5,6,7,8]. The analysis performed by gate [2] is based on the text retrieval technologies as defined in [1].

5 Conclusions and Further Work

The application described here is an effective solution to the problem of gathering information about local taxes, and has been fruitfully employed to collect real estate property taxes in Italy, with very good results in practice. The technologies employed in the application are up-to-date methods of deep web search and text analysis as proposed in the literature. The employed solution can be extended to further application cases and also used as a case study for proposing, as we did, an architecture that can be used for any similar case.

Acknowledgements

Authors gratefully thank CAAF CISL for financial support, and collaboration throughout the project. In particular we are strongly indebted with the National Director dr. Valeriano Canepari and CAAF CISL analyst and project coordinator Marco Bertoncelli who have been very helpful and available to us. A special thank goes to the people working in the local offices involved in the experimental phase who have been working hard and continue to do so in the second phase of the project. We also would like to thank Stefano Adami for his contributions in the development of the system.

References

1. Baeza-Yates, R., Gonnet, G.H.: A new approach to text searching. Commun. ACM 35(10), 74–82 (1992)
2. Cunningham, H., Humphreys, K., Gaizauskas, R.J., Wilks, Y.: Gate - a general architecture for text engineering. In: ANLP, pp. 29–30 (1997)
3. He, B., Patel, M., Zhang, Z., Chang, K.C.-C.: Accessing the deep web. Commun. ACM 50(5), 94–101 (2007)
4. Klinkert, M., Treur, J., Verwaart, T.: Knowledge-intensive gathering and integration of statistical information on european fisheries. In: Logananthara, R., Palm, G., Ali, M. (eds.) IEA/AIE 2000. LNCS (LNAI), vol. 1821, pp. 230–235. Springer, Heidelberg (2000)
5. Lee, C.-S., Jian, Z.-W., Huang, L.-K.: A fuzzy ontology and its application to news summarization. IEEE Transactions on Systems, Man and Cybernetics Part B 35(5), 859–880 (2005)
6. Lee, C.-S., Jiang, C.-C., Hsieh, T.-C.: A genetic fuzzy agent using ontology model for meeting scheduling system. Information Sciences 176(9), 1131–1155 (2006)
7. Lee, C.-S., Wang, M.-H., Chen, J.-J.: Ontology-based intelligent decision support agent for cmmi project monitoring and control. International Journal of Approximate Reasoning (2007)
8. Lee, C.-S., Wang, M.-H.: Ontology-based computational intelligent multi-agent and its application to cmmi assessment. Applied Intelligence (2007)

Intelligent Systems in Long-Term Forecasting of the Extra-Virgin Olive Oil Price in the Spanish Market

María Dolores Pérez-Godoy[1], Pedro Pérez[1], Antonio Jesús Rivera[1],
María José del Jesús[1], María Pilar Frías[2], and Manuel Parras[3]

[1] Department of Computer Science
[2] Department of Statistics and Operation Research
[3] Department of Marketing University of Jaén, Spain
{lperez,arivera,mjjesus}@ujaen.es, pedro.perez.recuerda@gmail.com,
mpfrias@ujaen.es, mparras@ujaen.es

Abstract. In this paper the problem of estimating forecasts, for the Official Market of future contracts for olive oil in Spain, is addressed. Time series analysis and their applications is an emerging research line in the Intelligent Systems field. Among the reasons for carry out time series analysis and forecasting, the associated increment in the benefits of the implied organizations must be highlighted. In this paper an adaptation of CO^{2}RBFN, evolutionary COoperative-COmpetitive algorithm for Radial Basis Function Networks design, applied to the long-term prediction of the extra-virgin olive oil price is presented. This long-term horizon has been fixed to six months. The results of CO^{2}RBFN have been compared with other data mining methods, typically used in time series forecasting, such as other neural networks models, a support vector machine method and a fuzzy system.

1 Introduction

Spatial and/or temporal data mining [13][22] is a growing application task inside the intelligent systems (data mining methods) field. There are different reasons for this growth, such as: the increase of this king of data which are collected and requiring analysis, their availability in Internet, the commercial advantage of the results obtained, etc.

Time series analysis and forecasting is one of the most important research topic of temporal data mining. A time series is a set of regular time-ordered observations of a quantitative characteristic of an individual phenomenon taken at successive periods or points of time. The problems in which the data are not independent but also have a temporal relationship are called time series forecasting problems. The ultimate aim in these problems is to increase our knowledge of a phenomenon or aspect in order to have a better understanding of it in the future.

Olive oil has become an important business sector in a continuously expanding market. In 2009 (http://www.mfao.es), World produced 2,888,000 of tons of olive

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 205–214, 2010.

oil, Spain is the first olive oil producing and exporting country and Jaén is the most productive province of Spain, with 430,000 tons, the 15% of the total production in the planet.

This is especially important in the Official Market for the negotiation of Future contracts for Olive Oil (MFAO) in Spain: a society whose objective is to discover prices that will balance, supply and demand, in some future time periods. Our aim is to predict these future prices in order to increase the global benefits of the sector.

The time series forecasting problem is usually addressed with data mining methods, such as neural networks [3][23][21][7] or fuzzy rule based systems [2][14][15][27]. Moreover statistic models, such as ARIMA [4], are used in time series analysis but this kind of methods are not suitable in long-term predictions [9].

Radial Basis Function Networks (RBFNs) are an important artificial neural network paradigm [5] with interesting characteristics such as a simple topological structure or universal approximation ability [18]. The overall efficiency of RBFNs has been proved in many areas such as pattern classification [6], function approximation [18] or time series prediction [25]. Typically these networks are design by means of evolutionary algorithms [11].

Authors have developed an evolutionary cooperative-competitive method, CO^{2}RBFN [19], for the design of RBFNs. An adaptation of this algorithm has proven its efficiency in short-term predictions [20]. The objective of this paper is to test the efficiency of the adapted CO^{2}RBFN in the long-term forecasting of the extra-virgin olive oil price in the Spanish market. It must be highlighted that, for first time, an horizon of six months has been fixed for this problem.

The results obtained using CO^{2}RBFN are also compared with other intelligent systems methods, typically used in time series forecasting, such a classical method for developing Fuzzy Systems (Fuzzy-WM) [24], a MultiLayer Perceptron Network trained using a Conjugate Gradient learning algorithm (MLP-ConjGrad) [17], a support vector machine (NU-SVR) [8], and a classical design method for Radial Basis Function Network learning (RBFN-LMS) [26].

This paper is organized as follows: section 2 describes CO^{2}RBFN and its extension to time series forecasting. In section 3 the other forecasting methods are described. The study and results obtained for the forecast methods are detailed in Section 4. In Section 5, conclusions and future works are outlined.

2 CO^{2}RBFN for Time Series Forecasting

CO^{2}RBFN [19], is an hybrid evolutionary cooperative-competitive algorithm for the design of RBFNs. In this algorithm each individual of the population represents, with a real representation, an RBF and the entire population is responsible for the final solution. The individuals cooperate towards a definitive solution, but they must also compete for survival. In this environment, in which the solution depends on the behavior of many components, the fitness of each individual is known as credit assignment. In order to measure the credit assignment of an individual, three factors have been proposed: the RBF contribution to the network output, the error in the basis function radius, and the degree of overlapping among RBFs.

The application of the operators is determined by a Fuzzy Rule-Based System. The inputs of this system are the three parameters used for credit assignment and the outputs are the operators' application probability.

The main steps of CO^{2}RBFN, explained in the following subsections, are shown in the pseudocode in Figure 1.

```
1. Initialize RBFN
2. Train RBFN
3. Evaluate RBFs
4. Apply operators to RBFs
5. Substitute the eliminated RBFs
6. Select the best RBFs
7. If the stop condition is not
     verified go to step 2
```

Fig. 1. Main steps of CO^{2}RBFN

RBFN initialization. To define the initial network a specified number m of neurons (i.e. the size of population) is randomly allocated among the different patterns of the training set. To do so, each RBF centre, $\boldsymbol{c}_i$, is randomly established to a pattern of the training set. The RBF widths, d_i, will be set to half the average distance between the centres. Finally, the RBF weights, w_{ij}, are set to zero.

RBFN training. The Least Mean Square algorithm [26] has been used to calculate the RBF weights.

RBF evaluation.A credit assignment mechanism is required in order to evaluate the role of each RBF ϕ_i in the cooperative-competitive environment. For an RBF, three parameters, a_i ,e_i ,o_i are defined:

- The contribution, a_i, of the RBF ϕ_i, $i = 1 \dots m$, is determined by considering the weight, w_i, and the number of patterns of the training set inside its width, npi_i. An RBF with a low weight and few patterns inside its width will have a low contribution:

$$a_i = \begin{cases} |w_i| & if \quad pi_i > q \\ |w_i| * (npi_i/q) & otherwise \end{cases} \tag{1}$$

  where q is the average of the npi_i values minus the standard deviation of the npi_i values.
- The error measure, e_i, for each RBF ϕ_i, is obtained by calculating the Mean Absolute Percentage Error (MAPE) inside its width:

$$e_i = \frac{\sum_{\forall pi_i} \left| \frac{f(pi_i) - y(pi_i)}{f(pi_i)} \right|}{npi_i} \tag{2}$$

where $f(pi_i)$ is the output of the model for the point pi_i, inside the width of RBF ϕ_i, $y(pi_i)$ is the real output at the same point, and npi_i is the number of points inside the width of RBF ϕ_i.

- The overlapping of the RBF ϕ_i and the other RBFs is quantified by using the parameter o_i. This parameter is computed by taking into account the fitness sharing methodology [10], whose aim is to maintain the diversity in the population. This factor is expressed as:

$$o_i = \sum_{j=1}^{m} o_{ij} \qquad o_{ij} = \begin{cases} (1 - \|\phi_i - \phi_j\|/d_i) & if \|\phi_i - \phi_j\| < d_i \\ 0 & otherwise \end{cases} \tag{3}$$

where o_{ij} measures the overlapping of the RBF ϕ_i y ϕ_j $j = 1 \ldots m$.

Applying operators to RBFs. In CO^{2}RBFN four operators have been defined in order to be applied to the RBFs:

- Operator Remove: eliminates an RBF.
- Operator Random Mutation: modifies the centre and width of an RBF in a percentage below 50% of the old width.
- Operator Biased Mutation: modifies the width and all coordinates of the centre using local information of the RBF environment. In the same way that the LMS algorithm.
- Operator Null: in this case all the parameters of the RBF are maintained.

The operators are applied to the whole population of RBFs. The probability for choosing an operator is determined by means of a Mandani-type fuzzy rule based system [16] which represents expert knowledge about the operator application in order to obtain a simple and accurate RBFN. The inputs of this system are parameters a_i, e_i and o_i used for defining the credit assignment of the RBF ϕ_i. These inputs are considered as linguistic variables va_i, ve_i and vo_i. The outputs, p_{remove}, p_{rm}, p_{bm} and p_{null}, represent the probability of applying Remove, Random Mutation, Biased Mutation and Null operators, respectively.

Table 1 shows the rule base used to relate the described antecedents and consequents. In the table each row represents one rule. For example, the interpretation of the first rule is: If the contribution of an RBF is Low Then the probability of applying the operator Remove is Medium-High, the probability of applying the operator Random Mutation is Medium-High, the probability of

Table 1. Fuzzy rule base representing expert knowledge in the design of RBFNs

Antecedents				Consequents				Antecedents				Consequents			
	v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}		v_a	v_e	v_o	p_{remove}	p_{rm}	p_{bm}	p_{null}
R1	L			M-H	M-H	L	L	R6		H		M-H	M-H	L	L
R2	M			M-L	M-H	M-L	M-L	R7			L	L	M-H	M-H	M-H
R3	H			L	M-H	M-H	M-H	R8			M	M-L	M-H	M-L	M-L
R4		L		L	M-H	M-H	M-H	R9			H	M-H	M-H	L	L
R5		M		M-L	M-H	M-L	M-L								

applying the operator Biased Mutation is Low and the probability of applying the operator null is Low.

Introduction of new RBFs. In this step, the eliminated RBFs are substituted by new RBFs. The new RBF is located in the centre of the area with maximum error or in a randomly chosen pattern with a probability of 0.5 respectively. The width of the new RBF will be set to the average of the RBFs in the population plus half of the minimum distance to the nearest RBF. Its weights are set to zero.

Replacement strategy. The replacement scheme determines which new RBFs (obtained before the mutation) will be included in the new population. To do so, the role of the mutated RBF in the net is compared with the original one to determine the RBF with the best behaviour in order to include it in the population.

3 Other Forecasting Methods

In this section other typically forecasting methods to apply to the long-term forecasting of the extra-virgin olive oil price are presented. These methods are a classical method for developing Fuzzy Systems (Fuzzy-WM)[24], a MultiLayer Perceptron Network trained using a Conjugate Gradient learning algorithm (MLP-ConjGrad)[17], a support vector machine (NU-SVR)[8], and a classical design method for Radial Basis Function Network learning (RBFN-LMS)[26]. Concretely:

- Fuzzy-WM [24]. This fuzzy system based design algorithm generates fuzzy rules from numerical input-output data pairs of the dataset. Domain intervals of the dataset are divided in regions and each region is assigned to a fuzzy membership function. For each input-output data pair one rule is generated. These fuzzy rules will represent a mapping from input to output space.
- MLP-ConjGrad [17]. MLP-ConjGrad uses the conjugate-gradient algorithm to adjust weight values of a multilayer perceptron [12]. Compared to gradient descent, the conjugate gradient algorithm takes a more direct path to the optimal set of weight values. Usually, the conjugate gradient is significantly faster and more robust than the gradient descent.
- RBFN-LMS. It builds an RBFN with a pre-specified number of RBFs. By means of the K-Means clustering algorithm it chooses an equal number of points from the training set to be the centres of the neurons. Then, it establishes a single radius for all the neurons as half the average distance between the set of centres. Finally, weights are analytically computed using the LMS algorithm [26].
- NU-SVR. The SVM (Support Vector Machine) model uses the sequential minimal optimization training algorithm and treats a given problem in terms of solving a quadratic optimization problem. The NU-SVR, called also v-SVM, for regression problems is an extension of the traditional SVM and it aims to build a loss function [8].

4 Experimentation and Results

The dataset used in the experimentation have been obtained from *Poolred*, an initiative of the Foundation for the Promotion and Development of the Olive and Olive Oil located in Jaén, Spain (http://www.oliva.net/poolred/). The time series dataset contains the monthly extra-virgin olive oil price per ton. With these data the addressed task is to forecast the extra-virgin olive oil price after six months. In this study, the data used are the price at origin from the 1^{st} month of the year 2002 to the 12^{th} month of the year 2008 in Spain. The cases in the data set were divided into two subsets: one for training and the other for testing. The data from the 1^{st} month of 2002 to the 12^{th} month of 2008 were used for training. The performance of the different predictions and methods were tested by estimating the data from the 1^{st} month to the 12^{th} month of 2008. Figure 2 shows the time series data and training and test datasets.

As mentioned, experiments carry out predictions with horizons of six months. In this way the patterns for data mining methods are heuristically (recommended by the experts of the sector) composed of $(n-5, n-4, n-3, n-2, n-1, n, n+6)$, when the price to forecast is $n+1$ and must be determined from the past prices $n-5$ to n.

The series has been differentiated to avoid problems related with the stationarity. The predictions have been performed using the differenced data, but errors have been calculated after reconstruct the original series.

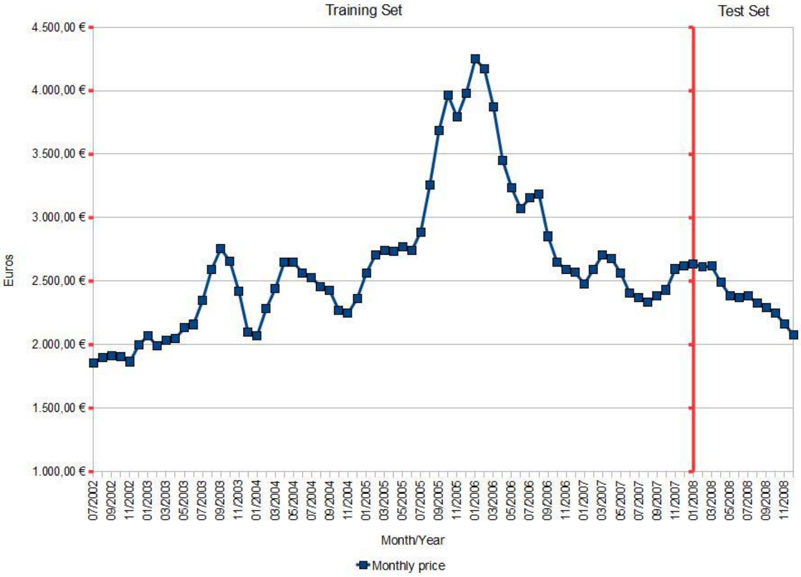


Fig. 2. Monthly extra-virgin olive oil prices in Tons / Euro

The implementation of the data mining methods has obtained from KEEL [1]. The parameters used in these data mining methods are the values recommended in the literature. For CO^{2}RBFN the number of executions is 200 and the number of RBFs or individuals in the population is set to 10. To obtain the results, algorithms have been executed 10 times (repetitions).

To estimate prediction capacity, the error considered is the Mean Absolute Percentage Error (MAPE):

$$MAPE = \frac{\sum_i^n (\mid (f_i - y_i)/f_i \mid)}{n} \tag{4}$$

where f_i is the predicted output of the model and y_i is the desired output.

Table 2 shows the obtained statistics, average and standard deviation for 10 repetitions, for MAPE. NU-SVR and Fuzzy-WM are deterministic methods and obviously the standard deviation for 10 repetitions is 0. Figure 3 shows the prediction achieved in the test set by the best training repetition of the methods.

Table 2. Results obtained by different methods forecasting the price of olive oil

	Statistics for 10 repetitions	
Method	Training MAPE	Test MAPE
Fuzzy-WM	0,08520 ± 0,0	0,21378 ± 0,0
MLP-ConjGrad	0,05270 ± 0,02007	0,19958 ± 0,02566
NU-SVR	0,10484 ± 0,0	0,18131 ± 0,0
RBFN-LMS	0,04343 ± 0,01875	0,17789 ± 0,03469
CO^{2}RBFN	0,10622 ± 0,00362	0,17567 ± 0,01111

If we analyze the results we can draw the following conclusions:

- The method proposed by the authors, CO^{2}RBFN, is the best method in average.
- Moreover, CO^{2}RBFN has the lower standard deviation, obviously for non-deterministic, which demonstrates the robustness of the method.
- With only 10 neurons, CO^{2}RBFN is the method with lowest complexity, along with MLP-ConjGrad.
- The dataset problem has a bias to overtraining and models such as RBFN-LMS (50 neurons) or MLP-ConjGrad have obtained lower training errors but they suffer overtraining in the test dataset.
- Methods do not achieve good predictions for the latest data in the test set, see Figure 3. Perhaps the behavior or the slope of these data is not in the training set.

Finally, it must be highlighted that the accuracy of the results obtained for this long-term prediction has been of interest for the olive-oil sector experts.

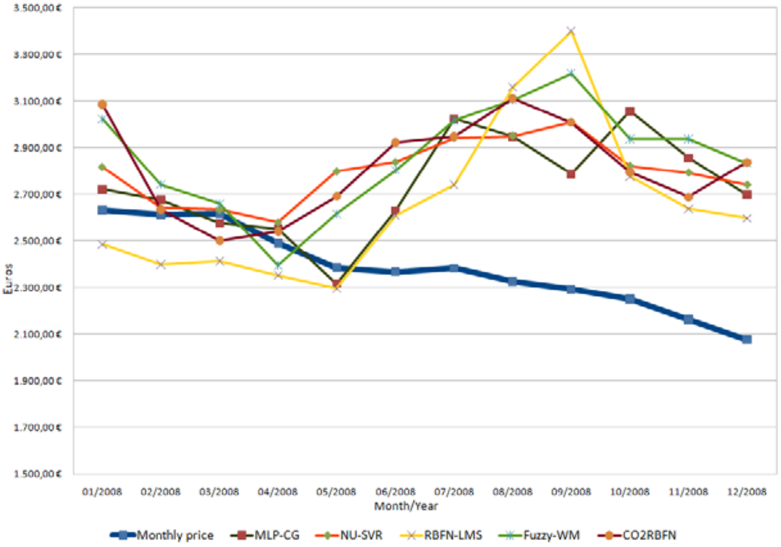


Fig. 3. Forecasting in the test set of the best training repetition

5 Concluding Remarks

This paper presents an application of the intelligent systems to the prediction of the extra-virgin olive oil price. This task is very important for the official Market for the negotiation of Futures contracts for Olive Oil (MFAO) in Spain: a society whose objective is to negotiate an appropriate price for the olive oil at the moment it is to be sold at a fixed time in the future. It must be highlighted that for first time the prediction horizon has been fixed to six months, therefore long-term predictions have been estimated.

With this aim in mind authors have tested an adaptation of their evolutionary cooperative-competitive algorithm (CO^{2}RBFN) to the forecasting of the extra-virgin olive oil price. As important key point of our proposal it is must be highlighted the identification of the role (credit assignment) of each basis function in the whole network. It is defined by three factors are defined and used: the RBF contribution to the network's output, a_i; the error in the basis function radius, e_i; and the degree of overlapping among RBFs, o_i. Another important key is that the application of the evolutive operators is determined by a fuzzy rule-based system which represents expert knowledge of the RBFN design. The inputs of this system are the three parameters used for credit assignment.

Typically data mining methods have been applied for comparisons such as methods for designing MLP networks, fuzzy systems, support vector machines of RBFNs.

From the results it can be conclude that CO^{2}RBFN is the best method in average for this problem. Moreover, the lowest standard deviation of CO^{2}RBFN demonstrates the robustness of the method. Finally, CO^{2}RBFN developed the model with the lowest complexity, with only 10 neurons.

As future lines, pre-processing for feature selection and exogenous features like meteorology or econometric data can be taken into account in order to increase the performance of the forecast.

Acknowledgments. Supported by the Spanish Ministry of Science and Technology under the Projects TIN2008-06681-C06-02, the Andalusian Research Plan TIC-3928 and the Project of the University of Jaén UJA-08-16-30.

References

1. Alcalá-Fdez, J., Sánchez, L., García, S., Del Jesus, M.J., Ventura, S., Garrell, J.M., Otero, J., Romero, C., Bacardit, J., Rivas, V., Fernández, J.C., Herrera, F.: KEEL: A Software Tool to Assess Evolutionary Algorithms for Data Mining Problems. Soft Computing 13(3), 307–318 (2009)
2. Azadeh, A., Saberi, M., Ghaderi, S.F., Gitiforouz, A., Ebrahimipour, V.: Improved estimation of electricity demand function by integration of fuzzy system and data mining approach. Energy Conversion and Management 48(8), 2165–2177 (2008)
3. Aznarte, J.L., Nieto, D., Benítez, J.M., Alba, F., de Linares, C.: Forecasting airborne pollen concentration time series with neural and neuro-fuzzy models. Expert Systems with Applications 32, 1218–1225 (2007)
4. Box, G., Jenkins, G.: Time series analysis: forecasting and control, revised edn. Holden Day, San Francisco (1976)
5. Broomhead, D., Lowe, D.: Multivariable functional interpolation and adaptive networks. Complex System 2, 321–355 (1998)
6. Buchtala, O., Klimek, M., Sick, B.: Evolutionary optimization of radial basis function classifiers for data miningapplications. IEEE Transactions on Systems, Man and Cybernetics Part B 35(5), 928–947 (2005)
7. Co, H.C., Boosarawongse, R.: Forecasting Thailand's rice export: Statistical techniques vs. artificial neural networks. Computers and Industrial Engineering 53(4), 610–627 (2007)
8. Fan, R.E., Chen, P.H., Lin, C.J.: Working set selection using the second order information for training SVM. Journal of Machine Learning Research (6), 1889–1918 (2005)
9. Franses, P.H., van Dijk, D.: Non-linear time series models in empirical finance. Cambridge University Press, Cambridge (2000)
10. Goldberg, D., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. In: Grefenstette (ed.) Proc. Second International Conference on Genetic Algorithms, pp. 41–49. Lawrence Erlbaum Associates, Mahwah (1987)
11. Harpham, C., Dawson, C.W., Brown, M.R.: A review of genetic algorithms applied to training radial basis function networks. Neural Computing and Applications 13, 193–201 (2004)
12. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice-Hall, Englewood Cliffs (1998)
13. Hsu, W., Li Lee, M., Wang, J.: Temporal and Spatio-temporal Data Mining. IGI Publishing (2007)
14. Jang, J.R.: ANFIS: Adaptative-Network-based Fuzzy Inference System. IEEE Trans. Systems, Man and Cybernetics 23(3), 665–685 (1993)

15. Khashei, M., Reza Hejazi, S., Bijari, M.: A new hybrid artificial neural networks and fuzzy regression model for time series forecasting. Fuzzy Sets and Systems 159(7), 769–786 (2008)
16. Mandani, E., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. International Journal of Man-Machine 7(1), 1–13 (1975)
17. Moller, F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks 6, 525–533 (1990)
18. Park, J., Sandberg, I.: Universal approximation using radial-basis function networks. Neural Computation 3, 246–257 (1991)
19. Pérez-Godoy, M.D., Rivera, A.J., Berlanga, F.J., del Jesus, M.J.: CO2RBFN: an evolutionary cooperative-competitive RBFN design algorithm for classification problems. Soft Computing (2009) doi: 10.1007/s00500-009-0488-z
20. Pérez-Godoy, M.D., Pérez, P., Rivera, A.J., del Jesus, M.J., Frías, M.P., Parras, M.: CO2RBFN for short-term forecasting of the extra-virgin olive oil price in the Spanish market. International Journal of Hybrid Intelligetn Systems 7(1), 75–87 (2010)
21. Pino, R., Parreno, J., Gomez, A., Priore, P.: Forecasting next-day price of electricity in the Spanish energy market using artificial neural networks. Engineering Applications of Artificial Intelligence 21(1), 53–62 (2008)
22. Roddick, J.F., Spiliopoulou, M.: A bibliography of temporal, spatial and spatio-temporal data mining research. ACM SIGKDD Explorations Newsletter 1(1), 34–38 (1999)
23. Ture, M., Kurt, I.: Comparison of four different time series methods to forecast hepatitis A virus infection. Expert Systems with Applications 31(1), 41–46 (2006)
24. Wang, L.X., Mendel, J.M.: Generating Fuzzy Rules by Learning from Examples. IEEE Transactions on Systems, Man and Cybernetics 22(6), 1414–1427 (1992)
25. Whitehead, B., Choate, T.: Cooperative-competitive genetic evolution of Radial Basis Function centers and widths for time series prediction. IEEE Trans. on Neural Networks 7(4), 869–880 (1996)
26. Widrow, B., Lehr, M.A.: 30 Years of adaptive neural networks: perceptron, madaline and backpropagation. Proceedings of the IEEE 78(9), 1415–1442 (1990)
27. Yu, T., Wilkinson, D.: A co-evolutionary fuzzy system for reservoir well logs interpretation. Evolutionary computation in practice, 199–218 (2008)

Processing Common Sense Knowledge to Develop Contextualized Computer Applications

Marcos Alexandre Rose Silva, Ana Luiza Dias, and Junia Coutinho Anacleto

Federal University of São Carlos. Washigton Luis KM 235, São Carlos, São Paulo, Brazil
{marcos_silva,ana_dias,junia}@dc.ufscar.br

Abstract. This paper describes how to use and to collect the common sense knowledge and how to process such information to create semantic network and to available it for applications. In this context, it is also presented an educational application which uses this knowledge to create a contextualized stories, because this game allows educators being co-authors on creating the story context taking into account their goals, pedagogical approach and the student's cultural reality. Then, teachers can adopt values, metaphors, cause and consequence relations or even a common vocabulary before and during the narrative game and, consequently, enabling students to feel identified in that story context being considered and get interested and engaged in collaborating with the teacher and other students to develop the story. In order to observe the use of this game in an educational environment and collect the target group opinion, a study case was performed at school with teachers and children, and is described in this paper.

Keywords: Common Sense, Storyteller, Narrative Game, Context, Education, Educational game, Culture.

1 Introduction

The culture has a significant role in social life, because it is contained in every gesture, attitude and thinking, becoming a key element in how the everyday is configured and modified. Thus, culture should be viewed as something fundamental, which determines people's life, feature, among others [8]. Therefore, to consider the experience, language, beliefs, values, finally people's common sense in the activities promoted by them, whether for entertainment or education is to consider their culture in order to allow them to better identify and more interest in performing a particular activity, because it is considering the people's reality to context this activity. In this context, common sense is a set of facts known by most people living in a particular culture, "covering a great part of everyday human experience, knowledge of spatial, physical, social and psychological aspects. In short, common sense is the knowledge shared by most people in a particular culture [1].

1.1 Processing Common Sense Knowledge

OMCS-Br project, developed by the Advanced Interaction Laboratory (LIA) at UFSCar in collaboration with Media Lab from MIT, has been collected common sense

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 215–224, 2010.

of a general public through a web site which can be accessed by anyone through http://www.sensocomum.ufscar.br. After entering, the person can register and have access to various activities and themes available in this site. One of the themes available is about Children's Universe, which allows people to talk about situations, objects and characters existing in the Children's Universe, such as Folklore, Fairy Tale, among others. Most of the activities and themes are templates as shown in Figure 1. Template **atrair meninos é uma característica do(a) personagem** Iara, (in English, **attract boys is a characteristic of the character** Iara).

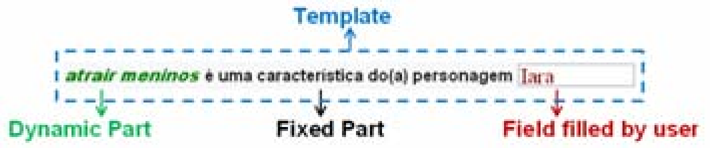


Fig. 1. Example of Children's Universe template

Templates are simple grammatical structures. They have fix and dynamic parts. Dynamic parts (green part) change when they are presented to users. They are filling out with data from other users' contribution already registered on the site. Therefore this base uses the stored knowledge to collect new one. Templates also have a field (red part) to be filled by users considering their everyday experiences and knowledge. In short that represents for them a common sense fact. Words typed by users are stored. These words are in natural language. Because of this, it is necessary to process them to computer to be able to use them. Generator Module of ConceptNet is responsible to process these data to generate a semantic network called ConceptNet [8].

1.2 Generator Module of ConceptNet

Generator Module is divided in three stages: Extraction, Normalisation and Relaxation. These modules are following described in a nutshell.

Extraction: This stage is responsible to define template relation, for example, the template relation **attract boys is a characteristic of the character** Iara (Figure 2) is PropertyOf. Because Iara has property of attracting boys. This template is stored (**PropertyOf 'Iara' 'attract boys'**).

There are others relations, such as: IsA, MotivationOf, UsedFor, etc. Minsky [8], researcher of Intelligence Artificial Area, has started studying these relations. He believed that computers could store all the data through binary relations. In short storing data modeled as a semantic network. His theory have showing useful for the culture sensitive software development [1].

Normalisation: In this stage some concepts (words) of the relation are normalized because nouns and adjectives of the sentence need are in singular and verbs in infinitive form. This process is executes in two steps: firstly Curupira [7], natural language parser, identify the syntactic structure, for example, Iara/name attract/verb boys/noun; Second, these structures are sent to Normalizer [1], it has as objective to define the singular and infinite words form, for instance, Iara/name to attract/verb boy/noun. Iara is in its singular form, attract is normalized to "to attract" and boys to boy. Normalisation is necessary to increase connectivity between concepts on ConceptNet. Avoiding that same concept can be stored in many forms, such as: attract, attracts, to attract, attracted. Then, if anybody wants to know what concepts are connected to "to attract", he/she does not need to type all forms.

Relaxation: This stage is responsible to define frequency (f) and inference (i) of the relations. Frequency is the number of times that users typed the same words on template, for example, if three users typed the same words as Figure 1 shows, frequency will be 3. In short (PropertyOf 'Iara' 'to attract boy' 'f=3; i=0').

Inference is the number of relations created automatically through inferences rules. For example, if user types: **Iara is a** funny girl. It will be created this relation: **(IsA 'Iara' 'funny girl')**. One rule describes that if have noun near adjective 'funny girl', is created a relation PropertyOf, because a property of girl (noun) can be funny (adjective), for instance, **(PropertyOf 'girl' 'funny' 'f=0; i=1')**. Figure 2 shows an image with simple example about how the concepts are stored. All concepts are connected with Minsky´s relations.

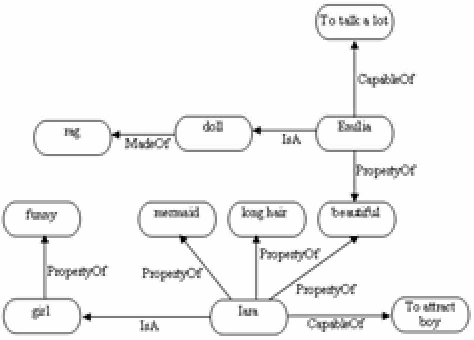


Fig. 2. Simple example of ConceptNet

After semantic network created any software can access it through API - *Application Programming Interface* [1]. It allows any application accesses ConceptNet through interface XML/RPC. This paper presents an application, called Contexteller, which uses ConceptNet to be contextualized.

Contexteller is a narrative game inspired in Role-Playing Game – RPG [3]. In this type of game there are participants and the master, who usually is the most experienced player and his task is to present the story to the group, with characters, their characteristics, scenarios and situations that require choices by other participants, who are the players. These players are not just spectators; they contribute actively in the story, through their characters that choose paths and take own decisions, and most of the time not foreseen by the master contributing to the spontaneous and unexpected development of the story.

In the context work the master is the teacher who introduces the story and intervenes collaboratively with the players who are the students. This game is meant to support a teacher to interact with students which have different social and cultural backgrounds.

The quality of relationships established at school, especially at the elementary school, can affect their learning and development. Because of that, the relationship among students and between students and teachers is very important. According to Benford [2], collaboration is also other important skill for young children to learn. These skills should be learned during childhood. In Brazil and in other emergent countries teaching those skills at school is still a challenge. Another skill that is important in education is to know how to live and to communicate with different people because each person has his own culture, values and socio-cultural reality [3].

On the other hand, activities to promote work in group can rarely occur spontaneously [4] so that teachers and students need to have activities and tools to support this new way of studying. There are many narrative games that teachers can use in the classroom [3], but most of them has common characteristic, as a previously defined context. They have a fixed set of characters, scenarios and themes to the storytelling. Because of this, if teachers want to use these games, they need to adapt their classes to the games rules. They also do not consider student's culture, knowledge. Some researchers Freire [6], Papert [9], Vygotsky [12] have described that when children identify the relation between what they are learning and their reality, they feel themselves more interest. In short, they can identify that the semantic of the words is significant to their life, because it is close to their reality.

Because of this, the narrative game proposed in this paper, Contexteller is a storyteller environment contextualized by common sense knowledge and it intends to support teachers on telling stories collaboratively with students according their pedagogical objective. Contexteller also intends to help children to notice familiarity with the characters, their characteristics and plot of the story. Therefore, this gives the teacher computer support through contextualized information so that he/she can create and tell stories. This support is provided by common sense knowledge – ConceptNet - that represents cultural aspects of the students´ community.

2 Contexteller

Figure 3 shows the interface available for players. This interface allows the players to see their card (I), their dice (II), and the text area (III), which allows the master to read all the messages sent to students and master during the composition of the collaborative story. In area (IV), the card, with another color and size, represents the master of the game, and area (V) shows to other characters´ card [11].

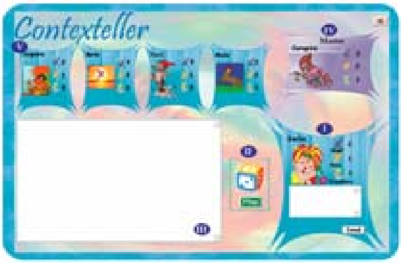


Fig. 3. Contexteller interface for players

2.1 Steps to Create a Story

At the Contexteller there are five steps to support teachers to create it.

First, some information about the teacher (master) is stored, such as: name, state, city, among others. Throughout this data it is possible to identify which teacher has created the game.

Second, it is defined which students are going to participate, their names, states and school education are some important information. The teacher can identify which students played and how each student told the story. Therefore, the teacher can observe children's behaviour, evolution and growth, for example, comparing children's attitude to the first and last stories. After the registers, Contexteller gets the students' state and filters the ConceptNet considering the knowledge collected from the desired profile in order to contextualize the game content for the target group. Because of this, in the next steps the teacher just sees contextualized information related to student's profile.

Step three, the teacher needs to choose one between two options: Creating a new game or choosing an existing game. If the teacher chooses the second option he/she can use another game that he/she or another teacher created.

Step four In this stage the teacher needs to define six characters: one represents her/him and the others represent students. Number 6 is usually used in RPG of cards [5] and according to Díaz-Aguado [4] six is the ideal number to work collaboratively. 5 players also facilitate the teacher to monitor the whole story that is being told by the players (students). If the number of players has been greater than 5, the teacher can face difficulties in reading all the messages, in interacting appropriately during the story and in observing the development and behaviour of each character.

When the teacher clicks on one of the cards, it was presented to her/him an interface to define a character, as in Figure 4. On this interface, the teacher can define a number for each element: magic and force. He/she can also choose an image to represent the character.

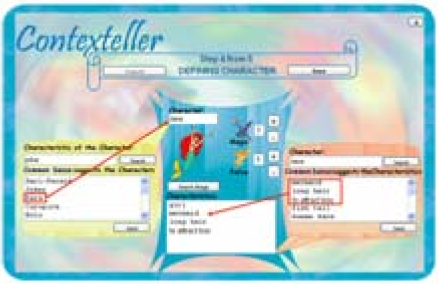


Fig. 4. Defining each character

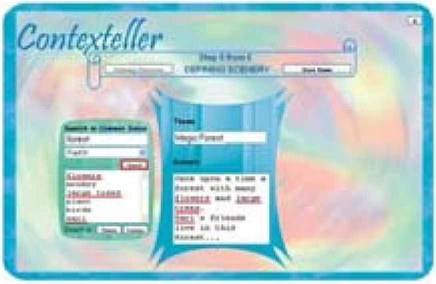


Fig. 5. Defining Scenery

There are two common sense cards to support teachers to define the characters´ names and characteristics.

In the first card (I), the teacher types a characteristic and searches the common sense knowledge base to obtain the characters´ names. For instance, if he/she wants to tell a story about a joke character and know which characters are related to this characteristic, considering the student's knowledge and culture, he/she can type this word (joke) on the card. Through the common sense knowledge base the following characters can be seen: Caipora, Chico Bento, Cuca, Saci-Pererê, Iara, Curupira (from the Brazilian folklore), Robin Hood, Elves, among others.

In the second card (II), it is possible to obtain the characters´ characteristics when the characters´ names are written on the card. For example, some characteristics coming up from Iara´s character are: mermaid, long hair, to attract boy, beautiful, fish tail, woman face, etc. The teacher can join this information, which students know about, with the story to define characters and theirs characteristics.

Figure 5 illustrates the **fifth and last step** in which teacher needs to define a subject and a title for the story. In this step the teacher uses common sense through a card from which he/she can get specific information of a word typed on the card, such as: IsA, PropertyOf, CapableOf, DefinedAs, DesireOf, MotivationOf, PartOf and others Minsky´s relations. For example, if the teacher is going to tell a story about forest and wants to know what students think about what exists in a forest, he/she can type "forest" on the card and select "PartOf" option, and then get some data, such as: characters, characteristics, animals and other things that students believe there are in a forest, i.e., concepts connected to "forest" through "PartOf" relation.

After these steps, the game is created. To start the game, students need to identify themselves to see the title and description of the story created for them, and the teacher's name. After, they can change the image that represents their characters. This feature allows the student to express himself not only through the story but also through the image. He can choose an image that makes his character sad, joyful, angry and so on.

Common sense knowledge also supports the teacher during story because he/she can see some data that represent students´ language and expression and can use them to describe something. For example, the teacher wrote ".. It's so dangerous. Caipora could you help to extinguish the fire?". There are many characters that can extinguish fire but children, in a specify region, know that Caipora has this characteristic. If the teacher obtains this information through common sense, he/she can tell the story considering the students´ reality and describe the actions in a familiar way.

On the other hand, it is important to explain that the objective of the common sense is to help the teacher to find out what students know about a story or even about events, causes and consequences. Therefore, Contexteller does not teach common sense to the teacher but gives him/her a cultural feedback and help him/her to find out what the students´ knowledge about the stories, facts, actions and among others. In order to observe the use of this game in an educational environment and also collect the views of students and teachers about the game, a case study was realized, described in next section.

3 Case Study

The Case Study was performed in a school that is a partner of educational projects in the Research Laboratory where the Contexteller Game was developed, in the state of Sao Paulo – Brazil with about 50 thousand people. This case study has divided the Case Study in three steps, each one with a distinct goal, which will be described as to following: First step: to explain the process and purpose of the Case Study for the school coordinator; Second step: to show and use the Contexteller Game with the teacher so that she could know the Game; Third step: to use the Contexteller Game in educational environment with teachers and students interacting.

First step: The proposal of the first step was presented to the coordinator of the school whole steps of the case study, the Contexteller and explained ways of collecting data. Presenting Contexteller to Coordinator: An instance of the narrative Contexteller Game was created in order to explain to the coordinator all user interfaces which the teacher and the students would have access during their interaction with the Game. The coordinator reaction seemed positive during all the presentation, because she kept her attention on the presented questions and gave some tips related to other ways to use Contexteller, as:
"While I´m listening the presentation, many possibilities comes to my mind. Did you imagine that this Game can be used in many ways? An idea that came to my mind is that I can use the Game during the History classes. For example, I can choose some Brazilian History characters and define that each student will be one of them and through these character the students can learn not only the History, but they will experience it too. " "There are lot of possibilities with this Game. The possibility to retrieve common sense suggestions is very important."

Second step: A teacher was chosen by the coordinator to participate to case study, but other teachers showed interest in participating of this step. In total, there were three teachers. In this step, a Contexteller Game instance was also created in order to show all the user interfaces that the teacher and students would have access. During the presentation, the teachers pointed out two positive aspects: the common sense and the co-authoring possibility. Teachers said that common sense stimulates creativity; expands the possibilities and the repertory. They also found relevant the possibility to create a game instance according to their needs, i.e., the possibility to be co-authors.

The teachers reported that many times they searched on the Web in order to find learning materials to use in class, but they said that it is too difficult to find a learning material that fits the target audience needs. Teacher 3 reported: "We know what we want, but we can't find it". Teacher 1: "That is right ". Then, Teacher 2 reported one of her experience with learning material edition: "Most of the time I need to print the material founded in the Web, scan it and change the material according to what I want… I do this because I can't change the site, but maybe I don't know how to use it properly, but I think that the site doesn't have this feature at all".

After the presentation, discussion and interaction with the Game, the teachers were invited to create an instance of the Game. It was possible to observe three different ways to use common sense knowledge to create the Game instance. The Teacher 1 utilized the common sense to find some character names and characteristics according to the suspense scenario that she was developing. She said that: "the common sense enriches".

Teacher 2 preferred to see all the existing characters that exist inside the common sense knowledge base, and using these options, choose some characters which she knows and have domain. Through the video analysis, it was possible to observe a comment that Teacher 2 did to the Teacher 1: "I want to use a story that I have certainty that I know… to do the inferences. I don't want to choose characters that I don't know. I want to use one that I have some domain". In this case, it is perceived that the Teacher 2 was afraid to create a story which she could lose the situation's control.

Teacher 3 preferred to define the characters, their characteristics and the scenario without utilize the common sense. After, she reported: "I stuck in the first step and I didn't use the common sense. It's interesting that I wanted to resume it and try the common sense. It's a database that is available and I didn't open the box. Did you understand? I went straight. I think that I did it because I didn't have experience with the tool".

Third Step: Three options were offered to teachers in order to investigate the use of common sense in a second contact with Contexteller: 1. Changing the instance created at the second step; 2. Creating new instance, or 3.Using an instance already created.

Teacher 2 chose the option 2, i.e., to create a new instance of the Game again. Because of this, she came to school 30 minutes before the students. This was the time teachers used to create an instance at the Second step.

It was possible to observe, analyzing videos, that the teacher used common sense cards more in this second contact with Contexteller, as illustrated in Table 1. In this table there are how many times each teacher searched and used the common sense help in first and second contact.

Table 1. Using common sense through Contexteller

Teacher	First contact	Second contact
Teacher 1	Searched: 8 times Used: 6 times	Searched: 11 times Used: 11 times
Teacher 2	Searched: 9 times Used: 6 times	Searched: 12 times Used: 8 times
Teacher 3	Searched: 0 times Used: 0 times	Searched: 5 times Used: 2 times

Using contexteller: Figure 6 shows the five students: Gender – 3 Feminine and 2 Male; Age – 11 and 12 years old, teacher and two researchers who are observing.

The teacher 2 created a scenario and characters with common sense help. The scenario was about a castle and the characters were: Beauty, Witch, Princess, Elf, Vampire and the Teacher. During the story, the teacher has used the common sense twice. First, to know what the students know about castle, and second what there is in the

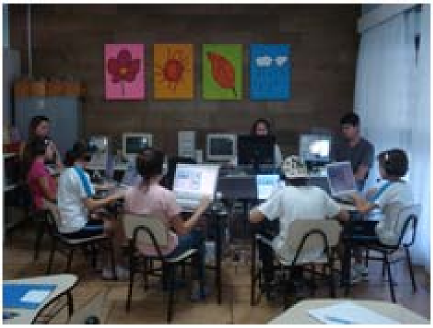


Fig. 6. Teacher and students using Contexteller

castle. After the Game, which took 45 minutes, the students said that Contexteller was very cool and funny. They also enjoyed the colors, the characters and the use of the Game was: easy. All students answered that they liked their characters chosen by the teacher with common sense help. Some of their comments: Student 5: "I loved very much to be an Elf, I imagined that I was him"; Student 1: "My character was Beauty. I liked her very much, she helped a lot to find out the password"; Student 3: "I was a Vampire and I like it. Because of this, I enjoyed to be one".

Most of the students considered very cool the possibility of telling stories with the teacher and friends; they also observed that everybody helped to tell the story. Students said that they did not want to stop the story; on the contrary, they wanted to play with their friends again. When they read the question about what they liked most, they answered: Student 1: "I liked to use my imagination to find out the password", Student 2: "Everything", Student 3: "mystery", Student 4: "I though that everything was real", Student 5: "I liked to use the imagination".

The teacher considered the Game easy and she liked to tell the story together with the students through the Contexteller. She liked a lot to use a scenario that she created by herself.

"There are many common sense suggestions to support before and during the story", she said in the questionnaire. She did not see any disadvantages in using common sense. Her opinion, "common sense is a good way to get to know the students´ culture and language. I believe that exchanging knowledge is important to learn in a collaborative way and it is a source of knowledge for everybody".

She wrote that the Game was suitable for children to express themselves and to work collaboratively. The teacher realized that the students collaborate with each other: "Characters were interconnected. It was important the collaboration." She realized that the students´ motivation during the story was good: "I noticed that the students were curious about what could happen later". Her last comment about the Game: "It was a pleasure to play this Game. I believe that it can be a good tool for teachers to help students to improve themselves in many development and knowledge areas".

4 Conclusions

The results of the case study have shown that common sense knowledge can be used to develop computer applications which access this knowledge to help teachers to create and to tell stories and it can allow the students feel themselves involved and liked the stories. Common sense knowledge makes easier for teacher to define characters with their characteristics, themes and scenarios. For the teacher this knowledge helps to remember these pieces of information and to present what the students like. Students also described that all the characters (chosen with common sense help) were very cool. Common sense also supports teachers to tell contextualized stories with characters, events, places, among others that students know.

Finally, common sense can be a good way to develop contextualized computer applications which consider a cultural context. Because of this, users can identify meanings through cultural symbolism, from common sense knowledge, in content of the application. Then, they can interact better with this application because the content of interface is related to their culture, values and socio-cultural reality. In this case, the students and the teacher were involved with application, i.e., the story.

Acknowledgements

We thank FAPESP, CNPq and CAPES for partial financial support to this research. We also thank all the collaborators of the Open Mind Common Sense in Brazil Project who have been building the common sense knowledge base considered in this research.

References

1. Anacleto, J.C., Lieberman, H., Tsutsumi, M., Neris, V.P.A., Carvalho, A.F.P., Espinosa, J., Zem-Mascarenhas, S.: Can common sense uncover cultural differences in computer applications? In: Bramer, M. (Org.) Artificial intelligence in theory and practice - WCC 2006, vol. 217, pp. 1–10. Springer, Berlin (2006)
2. Benford, S., Bederson, B.B., Akesson, K., Bayon, V., Druin, A., Hansson, P., Hourcade, J.P., Ingram, R., Neale, H., O´Malley, C., Simsarian, K.T., Stanton, D., Sundblad, Y., Taxén, G.: Designing Storytelling Technologies to Encourage Collaboration Between Young Children. In: Conference on Human Factors in Computing Systems, pp. 556–563 (2000)
3. Bittencourt, R.J., Giraffa, L.M.M.: A utilização dos Role-Playing Games Digitais no Processo de Ensino-Aprendizagem. Technical Reports Series, Number 031 (2003)
4. Diaz-Aguado, M.J.D.: Educação Intercultural e Aprendizagem Cooperativa. Ed. Porto, Porto (2003)
5. Fernandes, V.R.: What is RPG? RPG - Dragon Magazine in Brazil 123 (2008)
6. Freire, P.R.: Neves. Pedagogia da autonomia: saberes necessários à prática educativa. Paz e Terra, RJ (1996)
7. Martins, R.T., Hasegawa, R., Nunes, M.G.V.: Curupira: um parser funcional para a língua portuguesa. In: Relatório Técnico, 43 p. NILC, São Carlos (2002)
8. Minsky, M.: The society of mind. Simon & Schuster (1987)
9. Papert, S.: Mindstorms: Children, Computers, and Powerful Ideas. Basic Books, New York (1980)
10. Piaget, J.: Judgement and Reasoning in the Child. Littlefield Adams, Richmond (1999)
11. Silva, M.A.R., Anacleto, J.C.: Promoting Collaboration Through a Culturally Contextualized Narrative Game. In: 11th International Conference on Enterprise Information Systems (ICEIS 2009), Milan, anais ICEIS (2009)
12. Wertsch, J.V.: Vygotsky and the Social Formation of Mind (1988)

Estimating the Difficulty Level of the Challenges Proposed in a Competitive e-Learning Environment

Elena Verdú, Luisa M. Regueras, María Jesús Verdú, and Juan Pablo de Castro

University of Valladolid, Higher Technical School of Telecommunications Engineering,
Campus Miguel Delibes, 47011 Valladolid, Spain
{elever,luireg,marver,jpdecastro}@tel.uva.es

Abstract. The success of new learning systems depends highly on their ability to adapt to the characteristics and needs of each student. QUESTOURnament is a competitive e-learning tool, which is being re-designed in order to turn it into an adaptive e-learning system, managing different contests adapted to the progress of the students. In this adaptation process, the first step is to design a mechanism that objectively estimates the difficulty level of the challenges proposed in this environment. The present paper describes the designed method, which uses a genetic algorithm in order to discover the characteristics of the answers to the questions corresponding to the different difficulty levels. The fitness function, which evaluates the quality of the different potential solutions, as well as other operators of the genetic algorithm are described. Finally, an experiment with a real data set is presented in order to show the performance of this approach.

Keywords: Adaptive e-learning, competitive learning, fitness function, genetic algorithms, intelligent tutoring systems.

1 Introduction

In the last years, more and more European universities are adopting Learning Management Systems (LMS) to support their regular courses. On the one hand, these systems satisfy the demands of the new generation of digital natives as well as the needs of the current society, allowing universities to remain competitive. On the other hand, universities have to converge to the European Higher Education Area, adopting a learning model in which the student is the central actor of the learning process and thus, active learning methods should be applied. LMS, such as Moodle, incorporate a variety of tools that support and facilitate this type of learning. One of these tools is QUESTOURnament, an innovative e-learning system for active and competitive learning. Through the development of contests with several questions or challenges, QUESTOURnament enhances the students' motivation and academic results [1].

However, being the student the central part of the learning process, these tools should be able to adapt to the different characteristics of the students: their knowledge level, preferences, skills, etc., since not all the students process and perceive the information in the same way [2]. Moreover, the success of new learning systems will

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 225–234, 2010.

depend on the capacity for adapting the system to the student's cognitive level through the application of Artificial Intelligent methods and the definition of Intelligent Educational Systems [3].

Adaptive systems build a model of each student and use this model in order to adapt the course to his/her needs. It has been proved that the use of these adaptive systems improves the students' satisfaction and the effectiveness of the learning process [4]. Therefore, the QUESTOURnament system is being redesigned so that it adapts to the students' needs and requirements without losing its competitive nature. It requires grouping similar students who will compete among themselves. This process of adaptation includes different strategies. This paper focuses on how to estimate the difficulty level of the questions proposed in the contests through the use of Genetic Algorithms.

2 The Adaptive QUESTOURnament System

The QUESTOURnament system is a competitive e-learning tool, which allows teachers to organize contests, where students must solve several questions (called challenges within the system) in a time-constrained way. A ranking with the highest scores is permanently shown in order to promote the competition.

The competitive nature of QUESTOURnament provides important advantages since competitive learning is an effective method in order to capture the students' interest, increase the motivation and satisfaction [5] [6] and improve the learning process [1] [7].

However, this motivation is different for each student, according to their results, skills and knowledge level. Thus, it is necessary to develop a new competitive learning environment that adapts its functionalities to the needs and requirements of each student in order to motivate both winners and losers. This adaptation for QUESTOURnament requires the development of several strategies:

- Define how to estimate the objective difficulty level of the learning objects [8] [9], that is, the questions or challenges.
- Establish how to model and classify the degree of knowledge acquired by students during the course or contest [10], when they answer to the questions previously classified into different difficulty levels.
- Establish learning paths (or different contests in QUESTOURnament) adapted to the students' progress and knowledge level. The aim is that students with similar capabilities compete among themselves. In this way, the options of competitiveness within a contest are maintained in order to avoid the loss of motivation produced when a team or student distances oneself from other competitors during a contest.

Thus, the first step to be accomplished is to analyse how the difficulty level of a challenge can be objectively estimated from Moodle and QUESTOURnament logs. For each user interaction, the QUESTOURnament tool stores the time of interaction and information about the action (reading, submission…). The registered interaction data will be combined with grade values to obtain the final set of parameters that will be used as input for the genetic algorithm.

3 Genetic Algorithm to Estimate the Difficulty Level of the Challenges

Genetic algorithms emulate the natural evolution, working with a population of individuals, which evolve generation after generation achieving better fitness. This evolution process corresponds to a search through a space of potential solutions to a problem [11]. The starting point is a set of individuals (also called chromosomes), usually randomly generated, which represent possible solutions to the problem. Some of these individuals, normally those with higher fitness, are selected and combined through a crossover operator to produce children, which take part of the next generation. Even, with a low probability, some children suffer mutation through a mutation operator. Again, those individuals with higher fitness have more possibilities to be selected for a new crossover. Then, the fitness of individuals improves generation after generation. Fitness is computed using an objective evaluation function, commonly called fitness function, which plays the role of the environment and rates each individual according to its fitness to that environment [11]. The genetic algorithm finishes when no improvement is achieved after some generations or following other different criteria such as reaching a maximum number of generations. The output of the genetic algorithm is the best individual of the last generation, which represents the best solution to the problem.

Genetic algorithms have been widely used as search and optimization tools in various problem domains [12] as, for example, timetabling [13], vehicle routing [14], information retrieval [15] or marketing problems [16]. In the last years, genetic algorithms have been also applied to e-learning systems in order to automatically answer students' questions [17], retrieve useful student information [18] and predict student performance [19]. An example closer to the present problem is the genetic algorithm described in [8], which is used to estimate the difficulty level of the exercises proposed in a web-based system. However, the questions in this system can be solved through multiple attempts and receiving different hints, and then, most input parameters to this algorithm are not applicable to the present problem. Moreover, this algorithm needs an initial set of rules to characterize the different difficulty levels that must be defined by the teacher.

In the proposed solution for QUESTOURnament, when teachers add challenges to the system, they only have to indicate their estimated difficulty level: easy, moderate or hard. Starting from this initial classification done by teachers, the system automatically optimizes it by adapting to the real difficulty found by students when they solve the challenges. In this interaction of the students with the system, three parameters are taken into account in order to adapt the difficulty level of a challenge: time in minutes from the last reading of the challenge until the submission of the answer; grade; and the number of accesses to the description of the challenge before submitting the answer. Therefore, the genetic algorithm works with response patterns that have the following structure: <time, grade, accesses>. The objective is to characterize the answers belonging to each one of the three difficulty levels, finding numerical ranges for these three parameters that include the highest number of response patterns of each concrete difficulty. The genetic algorithm is run three times, to characterize the answers of the three difficulty levels. From the obtained numerical ranges, classification rules will be defined for each difficulty level and challenges will be re-classified according to these rules. Fig. 1 shows this adaptation process through the genetic system. The system assumes that teachers are experts in the knowledge domain of the course so that nearly all challenges are correctly classified by them. From this initial subjective classification, and once the answers of the students are submitted and assessed, the genetic system intends to emulate the subsequent labor of teachers, corresponding to the objective analysis of the answers for re-allocating those few challenges that were not solved as initially expected.

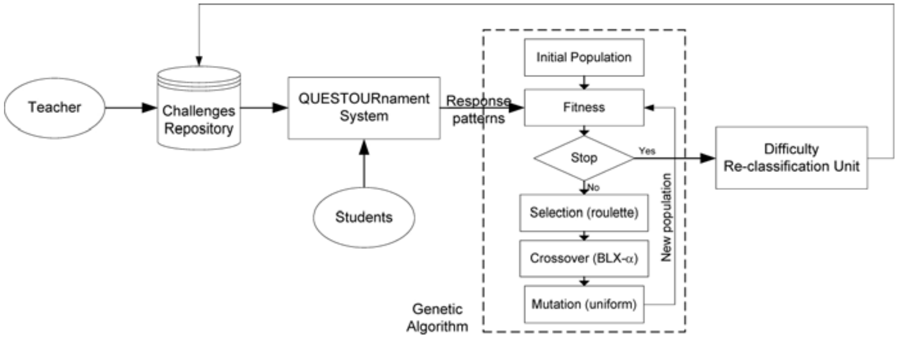


Fig. 1. Genetic System

3.1 Design of the Genetic Algorithm

Genetic algorithms typically work with binary-coded individuals or chromosomes. However, for real parameter optimization problems, real representation generally allows higher precision and performs faster than binary representation [11] [20]. Moreover, real representation facilitates the design and use of different crossover or mutation operators incorporating problem specific knowledge [11] [20]. Therefore, a real representation scheme was chosen for the present algorithm. As above mentioned, the objective is to find time, grade and accesses ranges that simultaneously comprise the highest number of response patterns of a concrete difficulty. Then, each possible solution to the problem is represented with a real-coded chromosome, which has the following structure: [t1, t2, g1, g2, a1, a2], where t1 and t2 correspond to the lower and upper limits of a time range, g1 and g2 correspond to the lower and upper limits of a grade range, and a1 and a2 correspond to the lower and upper limits of an accesses range.

After experimenting with several crossover operators, the BLX-α crossover operator [21] was selected for this first version of the algorithm. Moreover, the algorithm performs a uniform mutation [11] adapted to the problem so that both lower and upper limits of a range undergo the mutation process simultaneously. The BLX-α crossover operator as well as the mutation operator include the necessary checkings so that the generated chromosomes comply with the constraints of the problem: limits of the ranges within the variable domain and upper limit higher than lower limit. On the other hand, the implemented selection method is the roulette wheel [11].

Finally, among all the parameters and operators that have to be selected and tuned when working with genetic algorithms, the choice of the fitness function is crucial for the good performance of the algorithm [15] [22], as the utility of each chromosome is measured by this function.

3.2 The Fitness Function

The fitness function is required by the genetic algorithm in order to evaluate the quality of the individuals of a population, specifying the attributes desired for an optimal solution [23]. The fitness function must characterize the function to be optimized so its concrete definition depends on the specific problem. For example, there are fitness functions appropriate for information retrieval problems [15] [17] [22] or specific for solving the Snake in the Box problem [24]. In some subgroup discovery problems the authors base their fitness function on one or more classical measures found in the literature, which objectively evaluate the inferred rules [16].

As the searched solutions in the present problem will represent classification rules, objective quality measures used to evaluate inferred rules can make up the fitness function. Some of these measures are *coverage* [16] [25] [26], *support* [16] [25] [26] (also called *frequency* [27]), *size* [16] [26], *significance* [16] [26], *unusualness* or *weighted relative accuracy* [16] [26] [27] (also called *leverage* [25]), *accuracy* [27] or *confidence* [16] [25], and *novelty* [27].

The fitness function of the adaptive system is based on the support measure. The support of a rule is defined as the frequency of correctly classified covered examples [26], that is, those that satisfy both the antecedent and the consequent parts of the rule. Some other measures, such as confidence (number of examples satisfying both the antecedent and the consequent parts of a rule divided by those examples satisfying only the antecedent part), were also considered for taking part in the fitness function but, due to the reasons explained below, they were not included. As the fitness function is calculated multiple times, the time necessary for fitness evaluation is a critical aspect [19]. Calculating the support requires working with only patterns corresponding to one of the difficulty levels. Adding other measures (confidence or weighted relative accuracy) to the fitness function involves working with patterns of all difficulty levels and then, computational cost increases much, being this a critical aspect in the system. Moreover, the interest is to find ranges that include the highest number of patterns corresponding to a given difficulty, regardless of whether these ranges include also many patterns of other difficulty levels. Being interested in rules that maximize some specific measures such as support, to find rules that also maximize other measures would confuse the discovered interesting rules [25]. The aim is to characterize the questions corresponding to each difficulty level independently, dealing with the possible overlaps and gaps subsequently, once each level is characterized.

Therefore the fitness function is defined as shown in equation (1).

$$Fitness\ (C_i) = \frac{n_j^i}{n_j} \cdot w_t \cdot w_g \cdot w_a \,. \tag{1}$$

In equation (1) n^i_j is the number of patterns of difficulty j, which belong to the ranges of time, grade and accesses represented by the chromosome C_i, and n_j is the total number of patterns of difficulty j. Then, n^i_j divided by n_j corresponds to the support measure, whose possible values range from 0, when no patterns of a given difficulty are found within the intervals specified by the chromosome C_i, to 1, when all patterns match those intervals. Additionally, three weights w_t, w_g and w_a are used in order to award narrowest ranges and penalize widest ranges of time, grade and accesses respectively. Logically widest ranges will cover almost all the domain and will have high support. The interest is to find medium and narrow ranges which cover most examples. For example, if there are 100 response patterns to characterize the easy questions and all of them have a high grade between 85 and 100, the support of a chromosome specifying a grade range between 20 and 100 will be the maximum as all patterns match that interval (time and accesses ranges are ignored in this example for a clearer explanation). However, this wide range that almost comprises all the possible grade values, does not give much information about the actual grades obtained by students when solving these easy questions. On the other hand, a chromosome specifying a grade range between 80 and 100 will have the same support but it is much more interesting as it is more specific. Therefore, it is necessary to scale the support measure with some weights that award those chromosomes representing more specific ranges. Then, for the first chromosome whose grade range is 4 times wider than the range of the second one, w_g is equal to 0.1. For the second chromosome, w_g is equal to 1.2, thus its fitness value is much higher. The values of the weights for the different widths of the intervals were set experimentally.

3.3 Experiments and Results

Some experiments have been carried out in order to analyze the performance of the designed genetic algorithm. The data to be analyzed have been those registered from a contest made with the QUESTOURnament tool in a Telecommunications Engineering course during the year 2008-2009. The input response patterns correspond to the real answers given by students to 12 challenges. Initially, the teacher classified 4 challenges as easy, 3 as moderate, and 5 as hard. The total number of available answers for the easy, moderate and hard levels was 134, 103 and 169, respectively.

Table 1 shows the results of five executions of the genetic algorithm for each difficulty level using the fitness function of equation (1), the crossover operator BLX-α (α=0.3) with a crossover probability p_c=0.6 and a uniform mutation operator with a mutation probability p_m=0.01. The algorithm works with a population of 200 individuals and the number of generations is 10.

Table 1. Time, Grade and Accesses ranges found by the genetic algorithm for each difficulty level

Easy level				Moderate level				Hard level			
Time	Grade	Acc.	Fitness	Time	Grade	Acc.	Fitness	Time	Grade	Acc.	Fitness
[8,40]	[80,100]	[1,2]	0.657	[2,33]	[30,50]	[1,2]	0.352	[15,42]	[48,67]	[2,3]	0.138
[7,38]	[80,100]	[1,2]	0.657	[8,34]	[35,55]	[1,2]	0.301	[6,55]	[0,16]	[3,4]	0.132
[8,52]	[81,100]	[1,2]	0.399	[0,29]	[37,50]	[1,2]	0.301	[1,47]	[0,9]	[3,4]	0.155
[2,25]	[80,100]	[1,3]	0.419	[7,38]	[33,50]	[1,2]	0.402	[3,55]	[0,10]	[3,4]	0.167
[4,34]	[85,100]	[1,2]	0.502	[6,36]	[31,50]	[1,2]	0.377	[13,60]	[49,68]	[2,3]	0.132

Important differences are found in grade and accesses ranges among the different difficulty levels. The fitness reached in the easy level is much higher than the fitness achieved in the other difficulty levels due to the fact that most easy responses have a similar pattern. However, the values in the response patterns of the moderate or hard level challenges are more dispersed.

The ranges found by the algorithm fit the real data so the algorithm works as expected. However, only in a few cases the chromosomes of biggest fitness are diverse. In the case of easy level challenges, there are similar solutions which are the best; but, for the moderate and hard levels, there are different good solutions. For example, as it can be seen in Table 1 for the hard level, the second solution containing the grade range [0,16] and the accesses range [3,4] is as good as the fifth solution containing the grade range [49,68] and the accesses range [2,3] (both solutions have equal fitness while they are very different). Therefore, the aim is that the algorithm finds and delivers all these diverse good solutions. One method is to run the algorithm several times and get the different solutions it finds [16]. In order to avoid similar or duplicated solutions in subsequent executions, a penalty in the fitness function can be included for those solutions already found in previous executions. Methods that promote and control diversity typically use nonstandard selection, crossover, replacement or mutation strategies [28]. Some specific techniques are: injection of chromosomes into the population [13] [29], marking individuals with hash tags, not allowing duplicated hash tags in the population [30], adaptive control of the crossover and mutation rates [14], use of artificial neural networks to relocate converged solutions to solutions with a uniform distribution [31], niching methods [32], etc. Increasing mutation and crossover probabilities may promote diversity. However, some tests were done with the crossover and mutation probabilities set to 0.8 and 0.1, respectively (the rest of parameters remained the same as in the previous experiment), and a better performance in terms of diversity was not observed. Then other techniques, which promote diversity, have to be analysed and tested with different data sets in order to achieve diverse solutions.

Last, for all the executions of the algorithm shown in Table 1, all challenges would remain classified as the teacher did, except for two challenges initially classified as moderate and hard by the teacher, which would be re-classified. However, the re-classification result varies. The genetic algorithm has found two very different best solutions for the hard level in different runs and then re-classification of the questions differs. This re-classification has been done by taking the response patterns of a challenge and calculating the number of these patterns, which fit the ranges obtained by the algorithm for each of the difficulty levels. The challenge is assigned the difficulty level whose ranges include a higher number of the response patterns of this challenge. By incorporating diversity techniques, a more precise classification will be obtained for all executions of the algorithm.

4 Conclusions

A genetic algorithm has been implemented with the goal of objectively classifying the difficulty level of the challenges proposed in the competitive e-learning tool QUESTOURnament. The algorithm tries to find those time, grade and accesses ranges, which contain most response patterns corresponding to a given difficulty level. First experiments show an adequate performance of the algorithm when most of the response patterns have similar characteristics. However, the algorithm should deliver several best solutions, and not only the best one, when two or more different ranges characterize the response patterns of a given difficulty. Thus, the next step in this implementation is providing the algorithm with a mechanism that promotes diversity in solutions. The aim is that the algorithm delivers several solutions that maximize the fitness function, if they exist.

Besides achieving diversity, next future work includes:

- Post-processing of the output of the genetic algorithm: >From the ranges that the genetic algorithm delivers, several classification rules have to be defined. Overlaps and gaps among the ranges of the different difficulties can exist and, therefore, they must be treated. Fuzzy Logic will be analyzed as a possible component of the solution to this post-processing task.
- Definition of the process for re-classification of the difficulty level of the challenges, according to the previous post-processing solution.
- Tests with other existing crossover operators as well as new crossover and mutation operators designed specifically for the present problem with the aim of optimizing the performance of the genetic algorithm.
- Study of a possible implementation, which combines the genetic algorithm solution with a simple local search method (e.g. hill-climbing) in order to refine the output of the genetic algorithm.

Finally, during the year 2009-2010, new experiments are being done in order to test the algorithm with a higher number of input response patterns and so, to be able to verify the behavior of the defined rules.

References

1. Regueras, L.M., Verdú, E., Muñoz, M.F., Pérez, M.A., de Castro, J.P., Verdú, M.J.: Effects of Competitive E-learning Tools on Higher Education Students: A Case Study. IEEE Trans. Educ. 52(2), 279–285 (2009)
2. Lee, M.-G.: Profiling students' adaptation styles in Web-based learning. Comput. Educ. 36(2), 121–132 (2001)
3. Brusilovsky, P., Peylo, C.: Adaptive and Intelligent Web-based Educational Systems. Int. J. Artif. Intell. Educ. 13, 156–169 (2003)
4. Verdú, E., Regueras, L.M., Verdú, M.J., de Castro, J.P., Pérez, M.A.: An analysis of the Research on Adaptive Learning: The Next Generation of e-Learning. WSEAS Trans. Inform. Sci. Appl. 5(6), 859–868 (2008)
5. Philpot, T.A., Hall, R.H., Hubing, N., Flori, R.E.: Using games to teach statics calculation procedures: Application and assessment. Comput. Appl. Eng. Educ. 13(3), 222–232 (2005)
6. Siddiqui, A., Khan, M., Katar, S.: Supply chain simulator: A scenario-based educational tool to enhance student learning. Comput. Educ. 51(1), 252–261 (2008)
7. Zaphiris, P., Ang, C.S., Law, D.: Individualistic vs. Competitive Game-based E-learning. Adv. Technol. Learn. 4(4), 208–921 (2007)
8. Koutsojannis, C., Beligiannis, G., Hatzilygeroudis, I., Papavlasopoulos, C., Prentzas, J.: Using a hybrid AI approach for exercise difficulty level adaptation. Int. J. Contin. Eng. Educ. Life-Long Learn. 17(4/5), 256–272 (2007)

9. Smoline, D.V.: Some problems of computer-aided testing and “interview-like tests”. Comput. Educ. 51(2), 743–756 (2008)
10. Romero, C., Ventura, S., García, E.: Data mining in course management systems: Moodle case study and tutorial. Comp. Educ. 51(1), 368–384 (2008)
11. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1996)
12. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, United Kingdom (2001)
13. Sultan, A.B.M., Mahmod, R., Sulaiman, M.N., Abu Bakar, M.R.: Maintaining Diversity for Genetic Algorithm: A Case of Timetabling Problem. J. Teknol. 44, 123–130 (2006)
14. Zhu, K.Q.: A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time Windows. In: 15th IEEE International Conference on Tools with Artificial Intelligence, pp. 176–183. IEEE Computer Society, Los Alamitos (2003)
15. López-Pujalte, C., Guerrero-Bote, V.P., Moya-Anegón, F.: Order-based Fitness Functions for Genetic Algorithms Applied to Relevance Feedback. J. Am. Soc. Inform. Sci. Technol. 54(2), 152–160 (2003)
16. del Jesus, M.J., Gonzalez, P., Herrera, F., Mesonero, M.: Evolutionary Fuzzy Rule Induction Process for Subgroup Discovery: A Case Study in Marketing. IEEE Trans. Fuzzy Syst. 15(4), 578–592 (2007)
17. Hwang, G.-J., Yin, P.-Y., Wang, T.-T., Tseng, J.C.R., Hwang, G.-H.: An enhanced genetic approach to optimizing auto-reply accuracy of an e-learning system. Comput. Educ. 51, 337–353 (2008)
18. Romero, C., Ventura, S., de Castro, C., Hall, W., Hong, N.M.: Using Genetic Algorithms for Data Mining in Web based Educational Hypermedia Systems. In: AH 2002 workshop Adaptive Systems for Web-based Education (2002)
19. Minaei-Bidgoli, B., Kashy, D.A., Kortmeyer, G., Punch, W.F.: Predicting student performance: an application of data mining methods with an educational Web-based system. In: 33rd Annual Frontiers in Education Conference, pp. T2A- 13–18. IEEE Press, Los Alamitos (2003)
20. Wright, A.H.: Genetic Algorithms for Real Parameter Optimization. In: Rawlins, G.J.E. (ed.) Foundations of Genetic Algorithms, pp. 205–218. Morgan Kaufmann Publishers, San Mateo (1991)
21. Takahashi, M., Kita, H.: A Crossover Operator Using Independent Component Analysis for Real-Coded Genetic Algorithms. In: 2001 Congress on Evolutionary Computation, vol. 1, pp. 643–649. IEEE press, Los Alamitos (2001)
22. Fan, W., Fox, E.A., Pathak, P., Wu, H.: The Effects of Fitness Functions on Genetic Programming-Based Ranking Discovery for Web Search. J. Am. Soc. Inform. Sci. Technol. 55(7), 628–636 (2004)
23. Riekert, M., Malan, K.M., Engelbrecht, A.P.: Adaptive Genetic Programming for Dynamic Classification Problems. In: IEEE Congress on Evolutionary Computation, pp. 674–681. IEEE Press, Los Alamitos (2009)
24. Diaz-Gomez, P.A., Hougen, D.F.: Pseudo Random Generation of the Initial Population in Genetic Algorithms: an Experiment. In: 2nd Annual Computer Science Research Conference. University of Oklahoma Computer Science Graduate Student Association (2006)
25. Webb, G.I., Zhang, S.: K-Optimal Rule Discovery. Data Min. Knowl. Disc. 10, 39–79 (2005)
26. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup Discovery with CN2-SD. J. Mach. Learn. Res. 5, 153–188 (2004)

27. Lavrač, N., Flach, P., Zupan, B.: Rule Evaluation Measures: A Unifying View. In: Džeroski, S., Flach, P.A. (eds.) ILP 1999. LNCS (LNAI), vol. 1634, pp. 174–185. Springer, Heidelberg (1999)
28. Burke, E.K., Gustafson, S., Kendall, G.: Diversity in Genetic Programming: An Analysis of Measures and Correlation with Fitness. IEEE Trans. Evol. Comput. 8(1), 47–62 (2004)
29. Chang, P.-C., Huang, W.-H., Ting, C.-J., Fan, C.-Y.: Dynamic Diversity Control in Genetic Algorithm for Extended Exploration of Solution Space in Multi-Objective TSP. In: 3rd International Conference on Innovative Computing Information and Control, p. 461. IEEE Computer Society, Washington (2008)
30. Ronald, S.: Preventing Diversity Loss in a Routing Genetic Algorithm with Hash Tagging. Complexity Int. 2 (1995)
31. Kobayashi, K., Hiroyasu, T., Miki, M.: Mechanism of Multi-Objective Genetic Algorithm for Maintaining the Solution Diversity Using Neural Network. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 216–226. Springer, Heidelberg (2007)
32. Mahfoud, S.W.: Niching Methods. In: Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.) Evolutionary Computation 2, Advanced Algorithms and Operators, pp. 87–92. Institute of Physics Publishing, Bristol (2000)

Automatic Assessment of Students' Free-Text Answers with Support Vector Machines

Wen-Juan Hou, Jia-Hao Tsao, Sheng-Yang Li, and Li Chen

Department of Computer Science and Information Engineering
National Taiwan Normal University
No.88, Section 4, Ting-Chou Road, Taipei 116, Taiwan, R.O.C.
{emilyhou,g97470652,g97470470,g97470523}@csie.ntnu.edu.tw

Abstract. For improving the interaction between students and teachers, it is fundamental for teachers to understand students' learning levels. An intelligent computer system should be able to automatically evaluate students' answers when the teacher asks some questions. We first built the assessment corpus. With the corpus, we applied the following procedures to extract the relevant information: (1) apply the part-of-speech tagging such that the syntactic information is extracted, (2) remove the punctuation and decimal numbers because it plays the noise roles, and (3) for grouping the information, apply the stemming and normalization procedure to sentences, (4) extract other features. In this study, we treated the assessment problem as the classifying problem, i.e., classifying students' scores as two classes such as above/below 6 out of 10. We got an average of 65.28% precision rate. The experiments with SVM show exhilarating results and some improving efforts will be further made in the future.

Keywords: Free-text assessment, natural language processing, support vector machine.

1 Introduction

Assessing students' answers is a very time-consuming activity that makes teachers spend much time that they can devote to other duties. Automatic evaluation of students' answers is thus worthy of investigation. Recently, there are some related works for automatically assessing students' free-text answers. In [1], Dessus et al. designed Apex, a module of web-based learning environment which is able to assess student knowledge based on the content of free texts. It is not restricted to accepting the tests of multiple-choice questions, but furthermore based on the free-text essay. The technique they used is Latent Semantic Analysis (LSA) [2, 3]. Perez et al. [4] thought that though the BLEU algorithm only can be applied to the specific problem, it has some advantages of simple to programming and language independence. Consequently, the approach is suitable to integrate other techniques

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 235–243, 2010.

and resources to help scoring. In [5], the Tangow system and the Atenea system are integrated. The Tangow system records the student's action at runtime, including the login time, learning situation, and so on. The Tangow system will give the appropriate test questions, which can be used to guide the students during the rest of the course where Atenea is used to automatically score students' short answers. The Willow system is proposed in [6]. It combines the natural language processing (NLP) and the adaptive hypermedia (AH) techniques to automatically evaluate students' free-text answers given a set of correct ones.

For improving the correctness of the scoring, some researchers used part-of-speech (POS) tags to help with semantic representation. Wiemer-Hastings and Zipitria [7] evaluated a model of language understanding that combines a syntactic processing mechanism with an LSA-based semantic representation. They added POS tag for each word in the representation. Kanejiya et al. [8] presented an approach called Syntactically Enhanced LSA which generalized LSA by considering a word along with its syntactic neighborhood given by the POS tag of its preceding word, as a unit of knowledge representation.

In this paper, we focus on the automatic assessment of students' free-text answers by analyzing the information of sentences using Support Vector Machines (SVMs) [9, 10]. We treated the assessment problem as the classifying problem, i.e., classifying students' scores as two classes such as "good" or "not good" in the view of teachers. The classification of two categories is reasonable because the teacher can quickly understand the most students' learning situation for some topic and then decide the next teaching direction whether she/he should keep ongoing or stay for improving some teaching material. We presented a method that, given a collection of students' answers, extracted related information to produce a set of new features to be the representation of each answer. Then an SVM classifier for the assessment task was applied.

The rest of this paper is organized as follows. Section 2 presents the overview of our system architecture. We describe the proposed method in details, i.e. the components in our system, in Section 3. The experimental data used to evaluate the method and the results achieved by the proposed method are shown and discussed in Section 4. Finally, we express our main conclusions.

2 Architecture Overview

Figure 1 shows the overall architecture of our method for the assessment to the students' answers. At first, we preprocessed each training answer, tagged the part-of-speech of the word, and extracted several features from it, which will be explained in details in Section 3.2. We used a part-of-speech tag set, Penn Treebank. Each of the extracted features was considered a representation of this answer. Finally, a support vector machine (SVM) classifier was obtained.

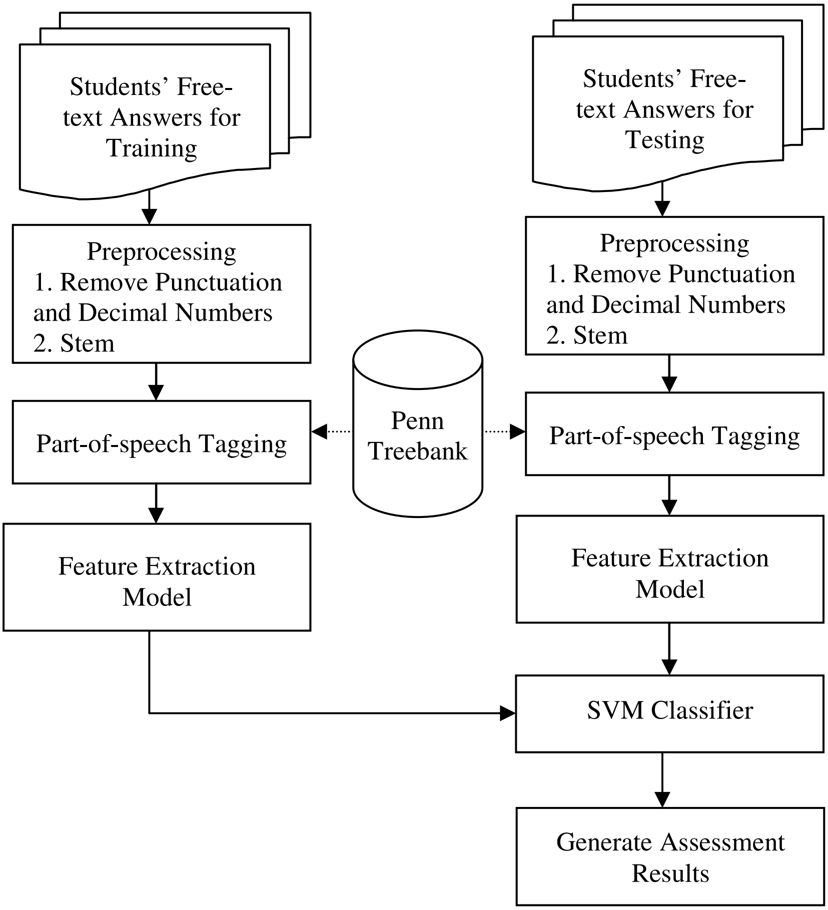

Students' Free-text Answers for Training
Students' Free-text Answers for Testing
Preprocessing
1. Remove Punctuation and Decimal Numbers
2. Stem
Preprocessing
1. Remove Punctuation and Decimal Numbers
2. Stem
Penn Treebank
Part-of-speech Tagging
Part-of-speech Tagging
Feature Extraction Model
Feature Extraction Model
SVM Classifier
Generate Assessment Results

Fig. 1. System architecture for the students' answers assessment

3 Methods

The methods are divided into three parts: (1) data preprocessing, (2) feature extraction and (3) SVM classification. In the preprocessing phase, we want to group word variations by stemming and filter noise by exclusion of punctuation and decimal numbers. In feature generation phase, we extracted POS for each word, term frequency (TF), Inverted Document Frequency (IDF) and entropy as the features to build our feature vector. In SVM classification step, we completed the classification for the answers. The details for each part are explained in the following subsections.

3.1 Data Preprocessing

Exclusion of Punctuation and Decimal Numbers. For gathering the useful information from student's answers, the first processing step is to remove punctuation and decimal numbers because they served as the noise roles.

Stemming. Stemming is a procedure of transforming an inflected form to its root form. For example, "inhibited" and "inhibition" will be mapped into the root form "inhibit" after stemming. Stemming can group the same word semantics and reflect more information around the word variations.

3.2 Feature Extraction

POS tagging. We used the Penn Treebank tag set [11] to assign part-of-speech tags to words. For example, for the title "Genetic characterization in two Chinese women", the tagged result will be "Genetic/JJ characterization/NN in/IN two/CD Chinese/JJ women/NNS" to indicate "Genetic" as an adjective, "characterization" as a singular or mass noun, "in" as a preposition, "two" as a cardinal number, "Chinese" as an adjective and "women" as a plural noun. There are 36 part-of-speech tags in the tag set. Examples of the tags are shown in Table 1. In this way, we extracted the preceding and the next POS tags for each word as the feature of the student's answers.

Table 1. Examples of the Penn Treebank tag set

POS Tag	Description	Example
CC	coordinating conjunction	and
CD	cardinal number	1, third
DT	determiner	The
EX	existential there	*there* is
FW	foreign word	d'hoevre
IN	preposition/subordinating conjunction	in, of, like
JJ	adjective	green
JJR	adjective, comparative	greener
JJS	adjective, superlative	greenest
LS	list marker	1)

Term Frequency (TF). If some word occurs frequently in the students' answers, it means the word may play an important role. Hence, under the bag-of-word representation, the term frequency is considered as a feature. The term count in the given answer is simply the number of times a given term appearing in that answer. We normalized this count to prevent a bias towards longer answers, which may have a higher term count regardless of the actual importance of that term in the answer, to give a measure of the importance of the term t_i within the particular answer a_j. Thus we have the term frequency $tf_{i,j}$, defined as follows.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} . \tag{1}$$

where $n_{i,j}$ is the number of occurrences of the considered term t_i in answer a_j, and the denominator is the sum of number of occurrences of all terms in answer a_j.

tf-idf (Term Frequency-Inverse document frequency). The tf-idf weight is a statistical measure used to evaluate how important a word is to a document in a corpus. In the study, we treated the student's answer as the document mentioned hereafter. The importance increases proportionally to the number of times that a word appears in the document but is offset by the frequency of the word in the corpus. The formula of computing the inverse document frequency is as follows:

$$idf_i = \log \frac{|D|}{|\{d : t_i \in D\}|} . \tag{2}$$

where $|D|$ is the total number of documents in the corpus, and $|\{d\text{: } t_i \in D\}|$ is the number of documents where the term t_i appears (that is $n_{i,j} \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. We will use the value of $1+|\{d\text{: } t_i \in D\}|$ instead.

The formula of the tf-idf weight is as the following:

$$(\text{tf-idf})_{i,j} = tf_{i,j} \times idf_i . \tag{3}$$

Entropy. The entropy is the average uncertainty of a single random variable. The definition of the entropy $H(X)$ is expressed in terms of a discrete set of probabilities $p(x_i)$:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) . \tag{4}$$

In the case of transmitted messages, these probabilities were the probabilities that a particular message was actually transmitted, and the entropy of the message system was a measure of how much information was in the message. In the paper, x_i is mapped to the word i of the students' answers. Thus, $H(x)$ represents the average uncertainty of the word appearance.

In summary, the features we extracted from students' answers are listed in Table 2. In Table 2, "TF" stands for the term frequency, "TF-IDF" is the weight of term frequency-inverse document frequency, "Pre tag" means the part-of-speech of the preceding word, and "Pos tag" is the part-of-speech of the next word.

Table 2. Features of each word in the feature vector

	No.	TF	TF-IDF	Entropy	Pre tag	Pos tag
Word 1	1	0.00042	0.00291	0.78372	NNNP … 1 0 …	NNNP … 0 1 …
Word 2	2	0.00094	0.00273	1.25396	NNNP … 0 1 …	NNNP … 1 0 …
Word 3	3	0.00735	0.00212	1.90172	NNNP … 0 0 …	NNNP … 1 1 …
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

3.3 SVM Classification

In this phase, we used SVM to classify students' answers. In our collected data, the score of each answer is graded from 0 to 10. For a teacher, she/he needs to understand the level of students in the class. Therefore, we give two categories where the answers with 6 and more scores are classified as category +1 and the others are classified as category -1.

The software SVM^{light} [12] was employed to deal with SVM-related operations. Radial basis function was adopted as the kernel function, and 4-fold cross validation was performed to select the model attaining the highest normalized utility.

4 Experiments and Results

4.1 Experimental Data

The experimental data was obtained from the course of automata and formal languages in the university. There were 38 students majoring in the course. In the course, the teacher prepared 9 questions and the corresponding references. After the examination, we got 38 answers for each question. The course teaching-assistant then gave the scores for each question according the corresponding references. The perfect score for each question is 10 points. Finally, there were 38 answers with the corresponding scores ranging from 0 to 10 for each question. Including the corresponding references, there were 351 answers for training and testing in the study. Because we treated the assessing problem as the classification problem, we therefore classified the collection data into two categories. One category included teacher's reference and students' answers with scores above and equal to 6. The other category contained student's answers with scores ranging from 0 to 5.

4.2 Evaluation Metric

The evaluation metric we used in this paper is the precision rate. The formula of precision is given in the following:

$$\text{Precision (P)} = \frac{TP}{TP + FP}. \tag{5}$$

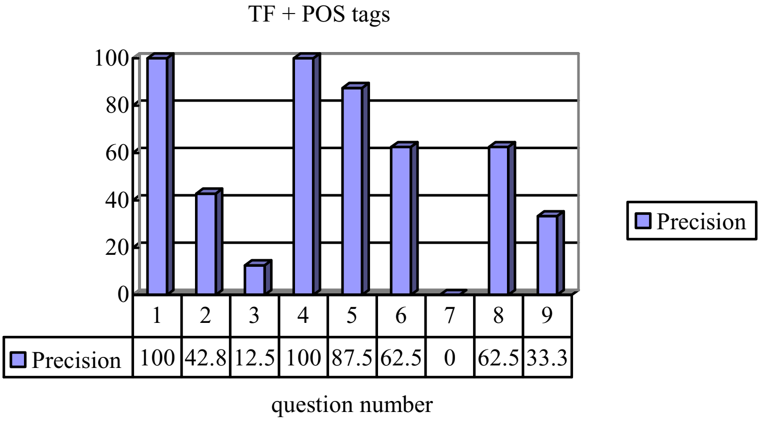
where *TP* is the number of true positives, and *FP* is the number of false positives. In this experiment, true positives are the correct classification proposed by our system. False positives are incorrectly classified as positives. For example, suppose that there is an answer a_i that should be classified as category +1. *TP* means the system classified it as category +1, which is correct. *FP* means the system classified it as category -1, which is incorrect.

4.3 Experimental Results and Discussions

In the study, we experimented with each question in sequence. For preparing training data, the standard answer was selected and then randomly chose 30 students' answers. Therefore, there were 31 answers for training for each question. In our training process through the SVM, the ratio between positive and negative answers was 277/283. The similar procedure was applied to other 8 questions. The resting 8 answers were left for testing. Then we computed the precision rate for each question to measure the performance.

We want to explore the importance of each feature, so the experiments using different features are made in the work. The experimental results are shown in Figures 2 and 3.

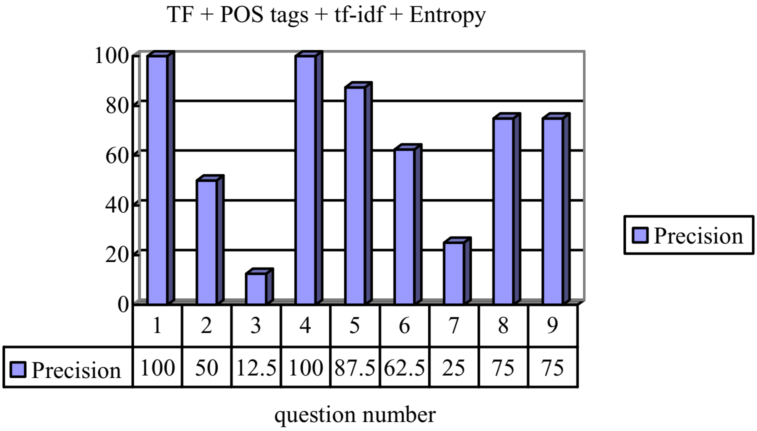
In Figure 2, we used the term frequency and the part-of-speech tag as features and got 55.68% average precision rate.



	1	2	3	4	5	6	7	8	9
□ Precision	100	42.8	12.5	100	87.5	62.5	0	62.5	33.3

Fig. 2. The precision using the term frequency and the part-of-speech tag (including Pre tag and Pos tag) as features

Then, we added the tf-idf and the entropy to the feature vector. Repeating the training and testing steps, we got the experimental result as shown in Figure 3.



	1	2	3	4	5	6	7	8	9
□ Precision	100	50	12.5	100	87.5	62.5	25	75	75

Fig. 3. The precision using the term frequency, the part-of-speech tag (including Pre tag and Pos tag), the tf-idf and the entropy as features

Table 3. Summary of the experimental results

Features	Precision for each question (%)									Ave.
	1	2	3	4	5	6	7	8	9	Prec.
TF+POS tags	100	43	13	100	88	63	0	63	33	55.68
TF+POS tags +TF-IDF +Entropy	100	50	12.5	100	87.5	62.5	25	75	75	65.28

The average precision rate in Figure 3 is 65.28%. The performance increases 9.60% compared to Figure 2. It shows that the tf-idf and the entropy do have positive influence for students' answers.

5 Conclusion

In this paper, we proposed the method to automatically identify students' learning level by analyzing students' free-text answers. We treated the assessing problem as the classification problem and then classified them into two categories in the study. We extracted the term frequency, the part-of-speech tag, the term frequency and inverse document frequency, and the entropy from students' answers. They served as features in the SVM machine learning approach. We experimented with different features and the summary results are listed in Table 3. In Table 3, the first column represents the adapted features in the SVM. Column 2 is the precision for question 1. Column 3 is the precision for question 2, and so on. The last column stands for the average precision for all questions. The average precision rate is 65.28%. The experiments show exhilarating results and some improving efforts will be further made in the future.

Acknowledgments. Research of this paper was partially supported by National Science Council, Taiwan, under the contracts NSC 98-2221-E-003-021 and NSC 98-2631-S-003-002.

References

1. Dessus, P., Lemaire, B., Vernier, A.: Free Text Assessment in a Virtual Campus. In: Proceedings of the 3rd International Conference on Human System Learning, pp. 61–75 (2000)
2. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T., Harshman, R.: Indexing by Latent Semantic Analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)
3. Foltz, T., Kintsch, W., Landauer, T.: The Measurement of Textual Coherence with Latent Semantic Analysis. Discourse Processes, Special Issue: Quantitative Approaches to Semantic Knowledge Representations 25(2-3) (1998)

4. Pérez, D., Alfonseca, E., Rodríguez, P.: Application of the BLEU Method for Evaluating Free-text Answers in an E-learning Environment. In: Proceedings of Language Resources and Evaluation Conference (LREC 2004), Portugal (2004)
5. Alfonseca, E., Carro, R.M., Freire, M., Ortigosa, A., Perez, D., Rodríguez, P.: Educational Adaptive Hypermedia meets Computer Assisted Assessment. In: Proceedings of the Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia, at AH 2004, Eindhoven, pp. 4–12 (2004)
6. Perez-Marin, D., Pascual-Nieto, I., Alfonseca, E., Anguiano, E., Rodriguez, P.: A Study on the Impact of the Use of an Automatic and Adaptive Free-text Assessment System during a University Course. In: Proceedings of the International Conference on Web-based Learning (ICWL), pp. 186–195. Prentice-Hall, Englewood Cliffs (2007)
7. Wiemer-Hastings, P., Zipitria, I.: Rules for Syntax, Vectors for Semantics. In: Proceedings of the 23rd Annual Conference of the Cognitive Science Society, pp. 1140–1145 (2001)
8. Kanejiya, D., Kumary, A., Prasad, S.: Automatic Evaluation of Students' Answers using Syntactically Enhanced LSA. In: Proceedings of the HLT-NAACL Workshop on Building Educational Applications Using Natural Language Processing, pp. 53–60 (2003)
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
10. Hsu, C.W., Chang, C.C., Lin, C.J.: A Practical Guide to Support Vector Classification (2003), http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html
11. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics 19(2), 313–330 (1993)
12. Support vector machine, http://svmlight.joachims.org/

On the Application of Planning and Scheduling Techniques to E-Learning

Antonio Garrido and Eva Onaindia

Universidad Politecnica de Valencia
Camino de Vera s/n, 46071 Valencia, Spain
{agarridot,onaindia}@dsic.upv.es

Abstract. AI planning techniques have been widely used in e-learning settings to create fully adapted courses to the specific needs, background and profile of students, but the practical issues of the real setting in which the course must be given have been traditionally ignored. In this paper, we present an integrated planning and scheduling approach that accommodates the temporal and resource constraints of the environment to make a course applicable in a real scenario. We also provide some experimental results to check the approach scalability.

Keywords: Intelligent systems in education, planning and scheduling.

1 Introduction and Motivation

E-learning most often means an approach to facilitate and enhance learning by means of computers and learning objects (LOs), typically any sort of digital resource on the Web. E-learning attempts to be a student-centered solution by offering a flexible learning process where courses and activities are tailored to the specific needs, background and profile of each student. The application of AI planning techniques has reported important advances to assemble LOs for course generation within e-learning, but this requires a whole meta-data description of structural relationships to reflect the logical sequence of contents [2]. A significant step in this direction is introduced by the IMS standards [8], which focus on LOs definitions and their sequencing adapted to the students' profiles.

Planning techniques have been used in the area of educational software to dynamically construct customized courses [3,6,9,11] as a means to bring the right content to the right person. However, a full-adapted course must cover not only the pedagogical/educational aspects, but also the practical issues of the real setting in which the course is given. While planning is primarily concerned with the achievement of a suitable combination of LOs, the physical aspects of the setting and the course have usually been left aside. For instance, we can have activities which require group interaction and collaboration, others may require to share some particular (and perhaps costly) resources, and students have their own scheduling constraints, as well as the temporal restrictions imposed by the environment. Intuitively, it does not suffice bringing the right content to the right person, but also at the right time and with the right resources.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 244–253, 2010.

In this paper, we examine the way in which planning and scheduling (P&S) techniques can be applied to aggregate LOs to build fully tailored and more practical e-learning courses. First, we explore the construction of learning courses (plans) by selecting the most adequate LOs. Second, we show how to achieve a plan scheduling that accommodates the temporal and resource constraints to make it applicable in a particular real setting. Our integrated approach combines a planner and a constraint satisfaction module which turn out to be very effective for both the elaboration and scheduling of e-learning courses, as well as for the detection of unsolvable problems with respect to the given constraints.

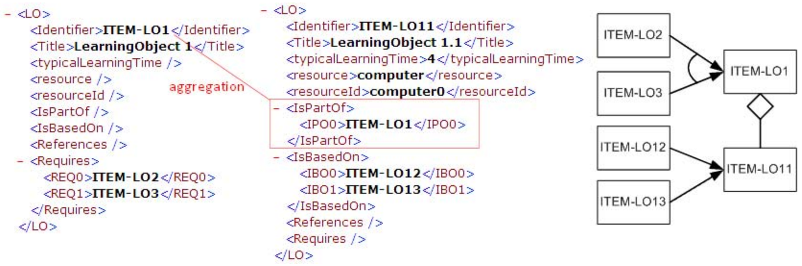
2 Mapping E-Learning to a Planning and Scheduling Model

In general, e-learning relies on instructional courses based on an explicit representation of the structure of the topics in the teaching domain and a library of LOs [12]. Modeling an e-learning environment, together with its related courses, is traditionally based on a thorough content description and meta-data labeling (see Fig. 1), a definition of the learning outcomes and a prerequisite analysis.

Basically, e-learning aims at forming a course as a composition of interrelated LOs which are usually specified in an XML standard format such as LOM [10]. The idea is to create a flow-like structure where some LOs are prerequisites of others. This sequence of LOs represents a learning route and makes it possible to attain the students' learning outcomes. This route keeps a strong resemblance with a plan, as usually modeled in AI planning [6]. On the other hand, AI planning models include collections of actions with prerequisites and outcomes —preconditions and effects, respectively— defined in a standard format known as PDDL [7]. When one action supports another, a causal link is established, meaning that the predecessor action must be finished before starting the successor one. Starting from an initial state, the set of causal links and applicable actions form the flow-like structure the plan needs to be properly executed and, consequently, to achieve the goals. These similarities make it possible to automatically map an e-learning model to a particular case of planning. To do this, we have implemented an authoring tool that models an e-learning problem as a PDDL planning problem by automatically extracting the essential information required for planning from the meta-data annotations of each LO.

Meta-Data for Planning and Scheduling

The standard LOM definition provides a full meta-data labeling specification for LOs, as depicted in Fig. 1. There exist many entries in this specification, but only a few of them are essential for P&S: i) information about the student's learning style; ii) relationships among LOs and their types; iii) typical learning time, i.e. duration of the LO; and iv) required resources, such as some special equipment. The only item that is not self-descriptive is *Relationships*, which comprises hierarchical structures and ordering relations based on content dependencies. The



```
- <LO>
    <Identifier>ITEM-LO1</Identifier>
    <Title>LearningObject 1</Title>
    <typicalLearningTime />
    <resource />
    <resourceId />
    <IsPartOf />
    <IsBasedOn />
    <References />
  - <Requires>
      <REQ0>ITEM-LO2</REQ0>
      <REQ1>ITEM-LO3</REQ1>
    </Requires>
  </LO>
```

```
- <LO>
    <Identifier>ITEM-LO11</Identifier>
    <Title>LearningObject 1.1</Title>
    <typicalLearningTime>4</typicalLearningTime>
    <resource>computer</resource>
    <resourceId>computer0</resourceId>
  - <IsPartOf>
      <IPO0>ITEM-LO1</IPO0>
    </IsPartOf>
  - <IsBasedOn>
      <IBO0>ITEM-LO12</IBO0>
      <IBO1>ITEM-LO13</IBO1>
    </IsBasedOn>
    <References />
    <Requires />
  </LO>
```

Fig. 1. Example of two LOs in XML meta-data specification and graphic formats

hierarchical structures use the *IsPartOf* relationship, which represents a hierarchical aggregation of LOs. For instance, if 'LO2 *IsPartOf* LO1' and 'LO3 *IsPartOf* LO1' both LO2 and LO3 are necessary in a learning route that includes the (higher level) object LO1. We also consider three types of ordering relations, and their inverse relations too, to represent causal dependencies: i) *Requires*, as a conjunctive prerequisite; ii) *IsBasedOn*, as a disjunctive prerequisite; and iii) *References*, as a (soft) recommended prerequisite.

The relationships specification mainly affect the planning process. The only item the scheduling is concerned with is the *Required resources*, but the definition given in the standards is very vague and does not provide details on the availability of resources, their capacity, quantity or usage cost, all of them key features for the scheduling process. Our authoring tool is designed to overcome this limitation and improve the quality of the LOs by modeling new planning and scheduling meta-data annotations.

Learning Objects as Planning Actions

Each LO is modeled as an action with duration, a student's profile, dependency relationships as preconditions, and outcomes as effects. For instance, given the LO named ITEM-LO11 of Fig. 1, the PDDL action is:

```
(:durative-action ITEM-LO11 ;;LearningObject 1.1
   :parameters (?s - student)
   :duration (= ?duration 4)
   :condition (and (at start (= (ITEM-LO11_done ?s) 0))
                   (at start (= (Start_ITEM-LO1_done ?s) 1))  ;;IsPartOf
                   (or (at start (= (ITEM-LO12_done ?s) 1))   ;;IsBasedOn
                       (at start (= (ITEM-LO13_done ?s) 1)))) ;;IsBasedOn
   :effect (and (at end (increase (ITEM-LO11_done ?s) 1))
                (at end (increase (ITEM-LO11_value ?s) 75));;adaptation
                (at end (increase (COST-computer0) 10))))  ;;resource cost
```

The action consists of four entries: parameters, duration, conditions and effects. The only parameter[1] is the student who will execute the action. The three

[1] Although our tool generates grounded actions for all students, i.e. actions with no parameters, we include the parameter student here for simplicity matters.

remaining entries are extracted from the meta-data specification. Atomic LOs such as ITEM-LO11 require only one action, whereas aggregated objects like ITEM-LO1 in Fig. 1 require two dummy actions, to start and end the hierarchical structure, respectively. Dummy actions are instantaneous, and the real duration of the aggregation depends thereby on the execution of its aggregated actions.

The information encoded in the PDDL action is represented through numeric functions, which allows us to deal with a broader domain of values and keep different levels of achievement for the LOs. In this example, all functions "_done" use a simple 0/1 value to limit whether that LO has been done or not. But, it is also possible to work with other values such as 0, 0.25, 0.5, 0.75, 1, which could be used to represent different competence levels. Numeric values provide a great expressiveness and are particularly useful to model adaptation, costs, rewards and multi-objective metrics. For instance, one effect of the above action ITEM-LO11 is to increase the value "(ITEM-LO11_value ?s)" in 75. This can be interpreted as increasing a particular student's competence value in 75 to reach a learning score, or simply as an evaluation mark (75 out of 100). The value 75 can be fixed or student-dependent. For example, a higher value (e.g. a better mark or learning reward) will denote a LO which is more appropriate for the student's learning style. It is important to note that these values always increase monotonically. Using exclusively increasing effects means that once a student has attained a competence value through the execution of one or several LOs, no further LOs can decrease such a value; that is, executing a LO cannot result in "forgetting" what the student attained with previous LOs. More formally, this monotonically incremental process means that the effects of a LO do not negatively interact with other LOs' effects. Analogously, the cost of a numeric resource (e.g (COST-computer0)) is also a monotonically increasing value which will depend on the resource itself, the student or the action that is using it. On the other hand, using a numeric representation also allows the user to optimize the plan according to different multi-objective metrics, such as minimize the makespan, the length or the resource cost of the plan, maximize the students' rewards, or any combination of them. These numeric features are very valuable in P&S but they are not listed in the e-learning meta-data specification. We can overcome this lack by modeling the features with our authoring tool [6], thus making it possible to consider more realistic scenarios and obtain more useful and successful solutions.

3 Planning and Scheduling Learning Routes

Fig. 2 depicts the architecture of our system. The LO repository contains the objects to design the courses. The authoring tool is used to extract the meta-data labeling of the LOs, to enrich the LOs by adding new P&S features and to automatically generate the course as a P&S problem. Once the personal characteristics (personal profile, resource availability and temporal constraints), background and preferences of the students are encoded, the planning stage calculates the learning route, i.e. plan, per student, as a sequence of LOs. This plan can be given to the student *as is* or can be used as an input for the scheduler when additional temporal/resource constraints are present. In such a case, the scheduler attempts to find a feasible schedule for the plan. If this is not possible, the scheduler backtracks (dotted line of Fig. 2) and calls the planner to have a new plan and repeat the process.

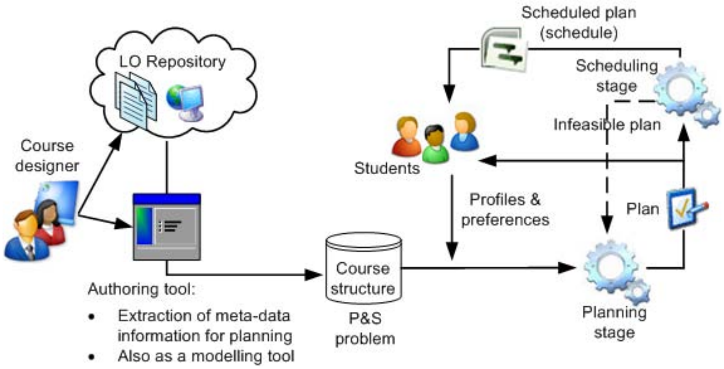


Fig. 2. Overall schema of the system

3.1 Planning Stage

We have implemented a planner that performs a two-phase process, inspired in the Graphplan's philosophy [1] but with significant differences. First, it generates a kind of numeric relaxed planning graph that also plays the role of a reachability analyzer, ruling out the actions that cannot be executed because their preconditions never hold. Second, it uses a classical goal-driven backward strategy to extract the learning route from the planning graph by regressively adding the profile-dependent actions that satisfy the goals.

Generation of the relaxed planning graph. Our planning graph is a multi-level directed graph but, unlike Graphplan's planning graph, a level is indexed by a timestamp `t` and it contains only one numeric state and the actions that can end at `t`. The relaxation we apply is that actions for the same student may overlap; a clear simplification because students cannot execute more than one action (LO) at a time. Our planning graph is entirely numeric and it is straightforwardly generated in polynomial time, more easily than Graphplan's, as shown in Algorithm 3.1. A state is formed by the numeric functions and their values. For instance, if we focus on the action `ITEM-LO11` of Section 2 with only one student `s1`, `state[0]={⟨ITEM-LO11_done s1,0⟩,⟨Start_ITEM-LO1_done s1,0⟩...⟨COST-computer0,0⟩}`. These values change over time by the effects of actions, so at each time point `t` the state comprises the values of the functions at `t`, i.e. the goals satisfied at time `t`.

```
$\mathtt{t} \leftarrow 0$
initialize $\mathtt{state}[\mathtt{t}]$ with the profiles & preferences of all students
while $\exists$ new actions that can be applied in $\mathtt{state}[\mathtt{t}]$ do
  for all $\mathtt{a_i} \mid \mathtt{a_i}$ can be applied in $\mathtt{state}[\mathtt{t}]$ do
    update $\mathtt{state}[\mathtt{t} + \mathtt{duration}(\mathtt{a_i})]$ with effects($\mathtt{a_i}$)
    update $\mathtt{actions\_end}[\mathtt{t} + \mathtt{duration}(\mathtt{a_i})]$ with $\mathtt{a_i}$
  $\mathtt{t} \leftarrow$ next level in the planning graph
```

Algorithm 1. Numeric relaxed planning graph generation

The while loop in Algorithm 3.1 proceeds until no more new actions can be executed (this is another difference with respect to Graphplan), which means that actions that are not present in the structure actions_end are unreachable, and thus irrelevant for the problem. In an e-learning setting, actions cannot be executed indefinitely (usually each LO is executed at most once, but in any case there is a limited number of possible executions). In consequence, the relaxed graph contains all the possible plans, thus guaranteeing its completeness. If the problem goals are not satisfied in the last state of the relaxed planning graph, the problem has no solution. Otherwise, we can guarantee that a solution exists, at least from a planning point of view (without considering the scheduling constraints), and then call the search stage.

Search: Extracting the plan. A backward process extracts the plan from the relaxed planning graph. Starting out from the last state in the graph, for each numeric goal we select the most appropriate action with respect to the problem metric to achieve that goal, as shown in Algorithm 2. The key points of this algorithm are steps 6–8, which encode the main differences with respect to a traditional plan extraction. In steps 6–7, we choose first the best $\mathtt{state}[\mathtt{t}']$, according to the metric to be optimized, that satisfies the current goal and, second, the best action $\mathtt{a_i}$ that *partially* supports the goal at $\mathtt{t}'$. For example, if the metric is to minimize the makespan of the plan, the algorithm chooses the shortest $\mathtt{t}'$ that satisfies goal $\mathtt{g_i}$; if the metric is to maximize a reward, the algorithm chooses the level $\mathtt{t}'$ in which the reward takes the highest value. Note that supporting a goal $\langle \mathtt{g_i}, \geq, \mathtt{75} \rangle$ whose initial value is 0 would require at least three supporting actions with an effect $(\mathtt{increase}, \mathtt{g_i}, \mathtt{30})$.

Finally, in steps 12–14 the algorithm includes more actions in case of maximization. This is a capability that most modern planners lack. Once all the original problem goals are attained, we create dummy goals, which are fictitiously supported by the remaining actions, to force the planner to add them into the plan, thus increasing the metric. For instance, if one student desires to obtain the highest score possible, the planner will include as many complementary actions as possible. Note again that this process always finishes because of the limited number of action executions (in e-learning, repeating the same LO many times does not scale up the learning outcome, so unlimited repetitions are not generally allowed).

```
1: $\mathtt{plan} \leftarrow \emptyset$
2: $\mathtt{t} \leftarrow$ last timestamp of the relaxed planning graph
3: initialize $\mathtt{goals}[\mathtt{t}]$ with the problem (learning) goals $\langle \mathtt{g_i}, \mathtt{binary\_comparator_i}, \mathtt{v_i} \rangle$,
   e.g. $\langle \mathtt{(ITEM\text{-}LO11\_value\ s1)}, \geq, \mathtt{90} \rangle$
4: while $\mathtt{goals}[\mathtt{t}] \neq \emptyset$ do
5:   extract $\langle \mathtt{g_i}, \mathtt{c_i}, \mathtt{v_i} \rangle$ from $\mathtt{goals}[\mathtt{t}]$
6:   if $\langle \mathtt{g_i}, \mathtt{c_i}, \mathtt{v_i} \rangle$ is satisfied in $\mathtt{state}[\mathtt{t'}] \mid \mathtt{t'} \leq \mathtt{t}$ then
7:     select best $\mathtt{a_i} \in \mathtt{actions\_end}[\mathtt{t'}] \mid \mathtt{a_i} \notin \mathtt{plan}$ that partially supports $\langle \mathtt{g_i}, \mathtt{c_i}, \mathtt{v_i} \rangle$
8:     update $\mathtt{v_i}$ of $\langle \mathtt{g_i}, \mathtt{c_i}, \mathtt{v_i} \rangle$ with effects($\mathtt{a_i}$) ;;partial support
9:     update $\mathtt{goals}[\mathtt{t'} - \mathtt{duration}(\mathtt{a_i})]$ with preconditions($\mathtt{a_i}$)
10:    update $\mathtt{plan}[\mathtt{t'} - \mathtt{duration}(\mathtt{a_i})]$ with $\mathtt{a_i}$
11:  update $\mathtt{t}$ with the latest timestamp with pending goals
12: if (metric to optimize = maximize) and ($\exists$ actions to be planned) then
13:  update $\mathtt{goals}$[latest timestamp] with a dummy goal $\langle \mathtt{fg_i}, \geq, \mathtt{1} \rangle$
14:  goto step 4
```

Algorithm 2. Extraction of a plan from the relaxed planning graph

When the algorithm finishes, the structure `plan` contains the actions that form the learning route. Note that this route is a parallel plan with overlapping actions for the same student. From a planning standpoint, this plan is optimal because of the following statements. First, all values increase monotonically and actions do not interact negatively, so there is no need to reason on mutexes, which indirectly guarantees that a locally optimal solution is also a global one. Second, the relaxed planning graph is complete and encapsulates all possible solutions. Third, the search process always extracts the state/action that is optimal with respect to the metric. Finally, when maximizing, the algorithm includes as many actions as possible until reaching the highest metric value. However, the plan needs to be sequentialized before being executed by each student, and this process can shed as a result a non-optimal plan with respect to the makespan (if this is involved in the metric). That is, a parallel plan which is optimal with respect to the makespan may not be optimal after being sequentialized.

From an e-learning standpoint, the process of finding a tailored learning route finishes here but, if we want a more realistic scenario with plans for students sharing temporal and resource constraints, we need a scheduling stage.

Table 1. Formulation of the elements of a plan + scheduling constraints as a CSP

Element	Formulation in the CSP model
action $\mathtt{a_i}$	$\mathtt{a_i} \in [0, \infty)$, which represents the start of $\mathtt{a_i}$
causal link $\mathtt{a_i} \rightarrow \mathtt{a_j}$	$\mathtt{a_i} + \mathtt{duration}(\mathtt{a_i}) \leq \mathtt{a_j}$
deadline for $\mathtt{a_i}$	$\mathtt{a_i} + \mathtt{duration}(\mathtt{a_i}) \leq \mathtt{deadline_i}$
ordering constraint $\mathtt{a_i} \prec \mathtt{a_j}$	$\mathtt{a_i} + \mathtt{duration}(\mathtt{a_i}) < \mathtt{a_j}$
synchronization (collaboration) of $\mathtt{a_i}$ among students $\mathtt{s1}, \mathtt{s2} \ldots \mathtt{s_n}$	$\mathtt{a_i\_s1} = \mathtt{a_i\_s2} = \ldots = \mathtt{a_i\_s_n}$
resource ($\mathtt{R_j}$) capacity constraint	$\sum_{\forall \mathtt{a_i} \mathit{that\ overlaps}} \mathtt{use}(\mathtt{a_i}, \mathtt{R_j}) \leq \mathtt{capacity}(\mathtt{R_j})$

3.2 Scheduling Stage

This stage uses a standard CSP (Constraint Satisfaction Problem) solving process to check whether the temporal and resource constraints hold in the plan. The plan is formulated as a CSP [5,6], including variables, domains and constraints, and it is extended with other scheduling constraints, as shown in Table 1. The first block in the table models the planning elements, i.e. actions and causal links. The second block models the scheduling constraints: deadlines, ordering constraints, synchronization constraints for collaborative activities that must happen at the same time, and resource capacity constraints to prevent overlapping actions from exceeding the max capacity of a resource[2].

The solving process creates a Temporal Constraint Network (TCN), checks the path consistency and assigns values to all variables [4]. Variables (actions) are heuristically selected in a chronological order, as they are executed in the plan. When an inconsistency occurs, we apply backjumping rather than chronological backtracking, thus avoiding unfruitful backtracking and speeding up performance [4]. Note that an optimal plan (from a planning perspective) may become non-optimal after adding scheduling constraints. In particular, a plan may include more tightly constrained actions and be scheduled as a worse solution than another plan with a worse initial quality —again, this can happen only if the metric includes the makespan. Further, a plan may become unfeasible at the scheduling stage because of the constraints that have been ignored during planning (e.g., a deadline on the length of the plan after including synchronization constraints, or a shared resource with scarce availability). If the scheduling process fails to find a feasible scheduling, the process goes back to the planning stage to find an alternative plan and repeats the cycle, as indicated in Fig. 2.

4 Experimental Results

In this section we show the performance of our system in an on-line AI course[3], in IMS-MD format, that contains 172 LOs with *IsPartOf*, *Requires* and *IsBasedOn* relations, and LOs particulary adequate for specific learning styles. Our aim is to analyze the quality of the plans and the scalability of the system, both when applying only planning (`P`) and planning and scheduling (`P&S`). We have conducted two experiments: minimize the plan makespan, and maximize the learning reward of the student. In both cases, we have defined problems with 1, 2, 4, 8. . . 256 students with different profiles and run the experiments with the options `P` and `P&S`. The results are shown in Fig. 3.

Minimization takes less time than maximization as plans contain fewer actions. Particularly, in `min-makespan` the number of actions in the whole plan is 77, 154, 308. . . 19712 for 1, 2, 4. . . 256 students, respectively. In `max-reward` the

[2] Our CSP formulation models each student as a resource with capacity=1 and so the final plan will never contain parallel actions for the same student.

[3] We thank Daniel Borrajo and his team for providing us this example course.

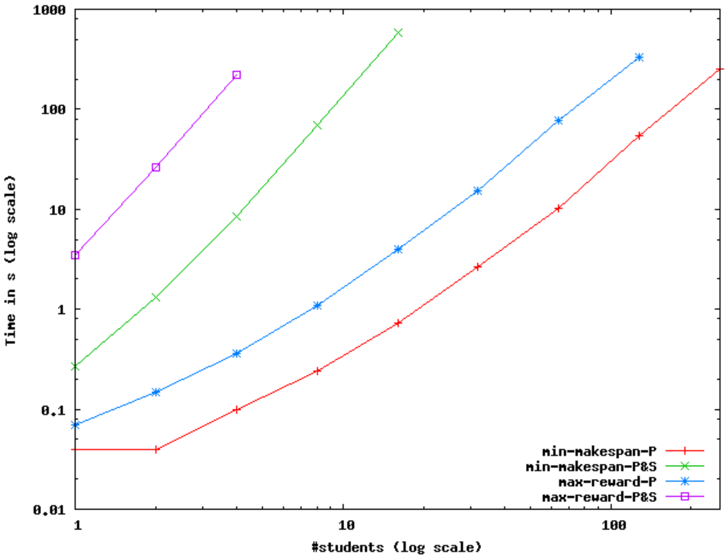


Fig. 3. Experimental results for two different metric optimization and P/P&S. All tests were censored after 600s. Note the log scale in both axis.

values are 163, 326, 652...41728 for the same number of students[4]. This difference with respect to the plan size is slightly relevant when doing only Planning, but it becomes more significant when doing Planning&Scheduling because of the complexity of the CSP. In particular, our CSP can handle up to 1500-2000 variables (actions) in a reasonable time, but more than this entails a very expensive process which exhausts the allocated time. This also happens in Planning, but it scales up better; in `min-makespan` a plan for 256 students is found in the given time, but in `max-reward` the max number of supported students is 128 within that time. We can conclude that our system can cope well with more than 100-150 students in Planning, and around 10-20 students, i.e. an order of magnitude less, in Planning&Scheduling —still a reasonable number when dealing with scheduling constraints, such as group interaction or collaboration.

5 Conclusions

The contribution of this paper are: i) an automated compilation that maps LO's standard meta-data specification to PDDL planning domains, ii) a planning and scheduling approach that automatically constructs routes tailored to the students' needs and preferences, iii) an entirely numeric planning approach inspired in Graphplan, and iv) a constraint programming plan formulation to include e-learning

[4] It is important to highlight that most current planners cannot deal with these problems, or simply solve problems up to 10 students, because they are entirely numeric.

scheduling constraints. Our future work includes employing this approach in some adaptive learning systems to better analyze its effectiveness.

Our approach is open, flexible and also applicable to other scenarios, such as Web services composition, with these requirements: data are monotonically increasing and can be encoded as numeric values, planning actions do not interact negatively, there are scheduling constraints that need complex reasoning, and a multi-objective optimization may be required.

Acknowledgments. This work has been supported by the Spanish MICINN under projects TIN2008-06701-C03-03 and Consolider Ingenio 2010 CSD2007-00022, the Valencian Prometeo project 2008/051 and Vicerrectorado de Investigacion de la UPV by PAID-05-09 4365-20100087.

References

1. Blum, A., Furst, M.: Fast planning through planning graph analysis. Artificial Intelligence 90, 281–300 (1997)
2. Brase, J., Painter, M.: Inferring meta-data for a semantic web peer-to-peer environment. Educational Technology and Society 7(2), 61–67 (2004)
3. Castillo, L., Morales, L., Gonzalez-Ferrer, A., Fdez-Olivares, J., Borrajo, D., Onaindia, E.: Automatic generation of temporal planning domains for e-learning. Journal of Scheduling (in press, 2010)
4. Dechter, R.: Constraint Processing. Morgan Kaufmann, San Francisco (2003)
5. Garrido, A., Arangu, M., Onaindia, E.: A constraint programming formulation for planning: from plan scheduling to plan generation. Journal of Scheduling 12(3), 227–256 (2009)
6. Garrido, A., Onaindia, E., Sapena, O.: Planning and scheduling in an e-learning environment. a constraint-programming-based approach. Engineering Applications of Artificial Intelligence 21(5), 733–743 (2008)
7. Gerevini, A., Long, D.: Plan constraints and preferences in PDDL3. In: Proc. International Planning Competition (IPC-ICAPS-2006), pp. 7–13 (2006)
8. IMS: IMS: global learning consortium (2008), http://www.imsglobal.org
9. Kontopoulos, E., Vrakas, D., Kokkoras, F., Bassiliades, N., Vlahavas, I.: An ontology-based planning system for e-course generation. Expert Systems with Applications 35(1-2), 398–406 (2008)
10. LOM: Draft standard for learning object metadata. IEEE (2002), http://ltsc.ieee.org/wg12/files/IEEE_1484_12_03_d8_submitted.pdf (rev. February 16, 2005)
11. Ullrich, C.: Course generation based on HTN planning. In: Proc. 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems, pp. 74–79 (2005)
12. Vassileva, J.: Dynamic course generation. Communication and Information Technologies 5(2), 87–102 (1997)

A Fuzzy ANP Model for Evaluating E-Learning Platform

Soheil Sadi-Nezhad[1], Leila Etaati[1], and Ahmad Makui[2]

[1] Islamic Azad University, Science and Research Branch, Faculty of engineering,
Department of systems & Industrial Engineering, Tehran, Iran
[2] Industrial Engineering Department, Iran University of Science and Technology,
Tehran Iran
sadinejad@hotmail.com, l.etaati@srbiau.ac.ir

Abstract. In recent past, there have been a great number of e-learning platforms, presented and used in various universities. Beside this there have been many efforts made to evaluate these systems which include a number of evaluation techniques and factors. However there may be some interrelations between these factors that must be considered during assessment. In this paper we present a Fuzzy Analytical Network Process (FANP) model for evaluating e-learning systems which can help us assume dependencies between criteria. This ANP model is based on the criteria and sub-criteria which were introduced by previous researchers. Also interrelations between the factors are considered from their suggestions. Therefore for deriving preference ratio among decision makers and to reach a global argument among them, we use Group Fuzzy Preference Programming. Finally we will show the general assessment techniques and some obtained results using our model to evaluate the three existing e-learning platforms which are employed in some universities and compare the results of ANP and AHP models.

Keywords: Fuzzy ANP, Evaluation E-Learning systems.

1 Introduction

E-learning is basically a web-based system that makes information available to learners. Also, it uses telecommunication technology to deliver information for education and training. The great advantage of E-learning includes liberating interaction between learners and instructors, or learners and learners, without limitation of time and space through the asynchronous and synchronous learning network models [19]. Methods of learning have been changed by improvement in information technology and the developing uses of electronic learning. Internet has the most important impact on creating asynchronous learning. Also by increasing the speed of information transfer, use of electronic learning is rising dramatically. Many companies use E-learning platforms to train their employees over the internet that causes reduction of the cost and time of training their employees. Although e-learning has been developing for so many years, the need for assessing the effectiveness of such learnings become more important. To evaluate an E-learning system, many factors must be considered during the assessment process which affect such system.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 254–263, 2010.

This paper establishes an extended approach to assessing E-learning platforms by considering some interrelationship between subfactors. For this reason we use Analytical Network Process (ANP) instead of Analytical Hierarchical Process (AHP). For determining decision maker's preference about comparison among factors we use Fuzzy Preferences Programming (FPP) analysis. The remainder of this paper is organized as follows: in the following sections, related literature is presented, in section three the methodologies used are explained and factors influencing the learner's satisfaction in E-learning environment with ANP application are discussed. Section four demonstrates a real case with three E-learning systems and finally in the last section the result and some further works are presented.

2 Related Works

In 2001, Soong [5] identified some Critical Success Factors (CSF) for favorable usage of on-line course resources. Wang [3] used a survey, generating items, collecting data, and validating the multiple-item scale for developing a comprehensive model and tools for measuring learner satisfaction with asynchronous e-learning systems. Finally he suggested four main criteria and 13 sub-criteria for evaluating E-learner satisfaction. In another research Hassan M-Selim [6], collected some data on measuring learner satisfaction about E-learning systems. He found out about the four main categories in Wang's survey and used confirmatory factor modeling approach to assess the criticality of the measures included in each CSF category. In the same year in Thailand Gwo-Hshiung Tzeng [7] used several approaches such as Factor analysis; fuzzy integral; DEMATEL and multiple criteria decision making (MCDM) to outline a hybrid MCDM evaluation model for e-learning effectiveness; this was done by considering an interrelation between criteria. Moreover, in 2009 Cheng- Mei [8] Hsu established a two step study (Delphi technique and heuristic method) to create an evaluation scale for web-based learning platforms.

Furthermore there are many methods to evaluate E-learning systems. There are some studies that use quality techniques to assess E-learning platforms such as evaluation method based on a *quality assurance* model [24]. Emre-Alptekin [2] applied the Quality Function Deployment (QFD) approach to assure quality of different e-learning products provided by the higher educational institutions in Turkey. Jill W. Fresen[4] used online quality management system (QMS) in the E-Education Unit at the University of Pretoria, South Africa. Also many other researches used MCDM approach for evaluating E-learning systems. AHP proposed by Saaty [20], has been widely used among other MCDM techniques for bench marketing of E-learning systems [9],[1],[10],[11],[12]. However in a real situation some criteria can have an impact on other criteria. For example Ihsan-Yu¨ksel [14] in his research proposed some relations between SWOT elements in strategy matrix. He wanted to choose the best strategy for different situations by considering such interrelationships.

3 The Methodology

To evaluate E-learning system by using Fuzzy Network Process, some steps must be taken. The following steps are required to evaluate E-learning systems in this research.

1- Select a Suitable assessment model
2- Determine interrelations between sub criteria
3- Create network structure
4- Create a pairwise comparison questionnaire based on ANP
5- Collect questionnaires and obtain comparisons
6- Evaluate by applying three scenarios
   Evaluate by applying *Network by content scenario.*
   Evaluate by applying *AHP scenario.*
   Evaluate by applying *Content & Learner Interface scenario.*
7- Compare all the results together

In the first step we select a suitable model for evaluating E-learning systems from previous researches. After that, we consider the interrelationship between its criteria. At the third stage, according to the above mentioned relationship based on ANP application a network structure and a pairwise comparison questioner, is created. For achieving this, an AHP questionnaire with fuzzy scale is made. Then, these questionnaires are distributed among students and managers of E-learning systems to obtain the preferences. For evaluating E-learning systems we consider three different scenarios described in section 2.2. At the end of the process, the result of each scenario is compared with others and final evaluation is obtained.

3.1 E-Learning Factors

We use the hierarchy structure for evaluating WELS which has been proposed by Wang [3]. This model has four main dimensions including: *Learner Interface, Learning Community, System Content and Personalization.* Each dimension has its criteria, for example, System Content has three criteria including: *Up-to-date content, sufficient content, and Useful content.* As mentioned before, Gwo- Hshiung [7] presented some criteria and their interrelationships among them. One of them is *Instruction material* that has three criteria, namely, *Level of instructional material*, *Readable* and *Update frequently* which *Update Frequently Material* has been affected by two other criteria, *Readable* and *Level of Instruction*. This relationship demonstrates for having readable (useable) or sufficient content; all contents must update frequently. In our model we can demonstrate these relations as in Figure 1.

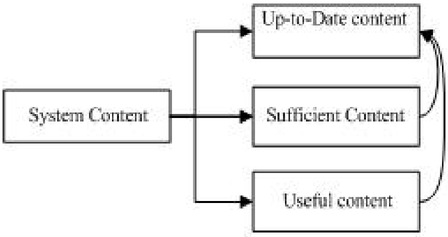


Fig. 1. Interrelationships between System content criteria

Moreover; there are some debates on learner interface. Learner interface allows the learner to access the information in E-learning system [16]. Learner interface is the same as user interface in a software program which allows the user to use software. According to Wang model, it has four criteria such as: *ease of use*, *user friendly*, *ease of understanding*, and *operational stability*. Nigel-Bevan [17] in his research on usability defines *ease of use* as a term for *Usability* that was defined in ISO/IEC 9126-1. Hence we can use *Usability* instead of *ease of use*. Furthermore, there are some criteria that have the most impact on usability. ISO/IEC FDIS 9126-1 quality model introduced 4 items that can affect usability, understandability, learn ability, operability and attractiveness (user-friendly).

As a result, *ease of use* has the same subcriteria as *Usability* dimension. Thus, *ease of understanding* (learn ability), *user-friendly* (attractiveness) and *operational stability* (operability) can affect *ease of use* criterion. These relations are illustrates in figure 2.

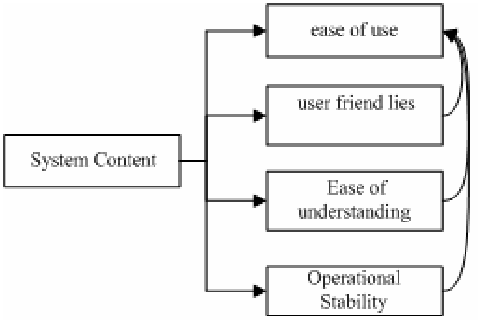


Fig. 2. Interrelationships between Learner Interface's criteria

3.2 Fuzzy ANP Application

ANP developed by Thomas L. Saaty [22], provides a way to input judgments and measurements to derive ratio scale priorities for the distribution of influence among the factors and groups of factors in the decision. The well-known decision theory, the Analytical Hierarchy Process (AHP) is a special case of the ANP. Both the AHP and the ANP derive ratio scale priorities by making paired comparisons of elements on a common property or criterion [15]. Many decision-making problems cannot be structured hierarchically because they have some dependency among their factors. So structuring a problem with functional dependencies that allows feedback among clusters is considered to be a network system. Each element in such network is called *cluster*. Figure 3 shows the final network structure for evaluating E-learning systems. For solving an ANP structure there are four steps to be followed. These steps include:

Step1: *Creating Network Structure, which shows interrelationship between clusters*. These relations can be obtained through brainstorming by experts or studying previous researches. In this paper we created a network structure based on previous researches.

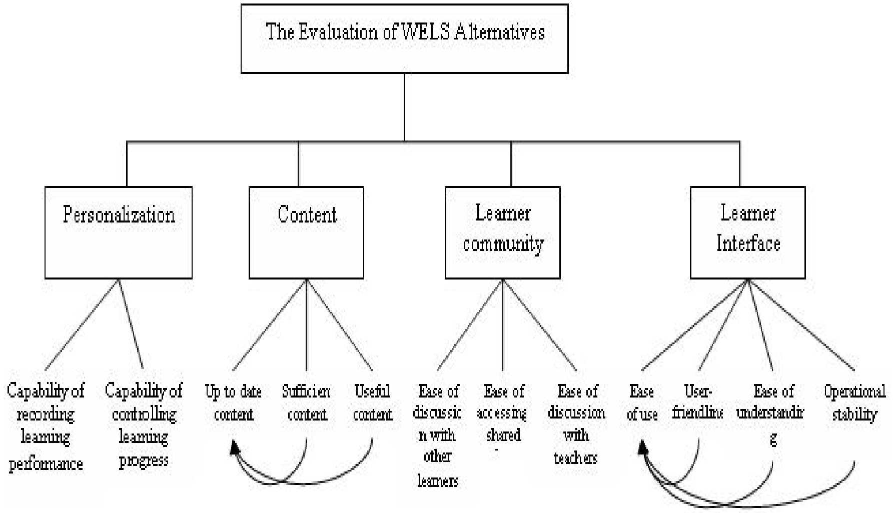


Fig. 3. Final Network structure for Evaluating E-learning systems

Step2: *Creating pairwise comparison matrix and priority of vectors.* Pairs of decision elements at each cluster are compared with respect to their importance towards their control criteria. There are some scales to determine the importance of one cluster on others. Also, for helping the decision makers to decide between pairs of elements, and improving their accuracy of decision makers, we use linguistic variable weight method as is shown in table 1[18].

Table 1. Linguistic value look-up table

Fuzzy language	*the mean of fuzzy numbers*
Equal important	1
Weak important	2
Fairly strong important	3
Very strong important	4
Absolute important	5

Step3: *Capture comparison result from questionnaires.* This model has four dimensions; hence, for comparing the affection of each pair of elements on the main goal, we use questionnaires which have been filled out by decision makers. Furthermore, for each dimension and its criteria, we define some questionnaires for reflecting the affection of criteria on each dimension. Also, there are two more relations that show the interrelationship between criteria. Finally we have 12 main criteria by which all the alternatives compared.

Step4: *Creating a super matrix that is similar to Markov chain process [14].* To identify global priorities in E-learning system with its interdependence, we create a super matrix. We consider three main scenarios for evaluating E-learning systems.

First of all is *AHP scenario* which has a super matrix like matrix 1. In this scenario we did not consider any interrelationships between criteria.

$$\begin{array}{lcccc} goal & 1 & 0 & 0 & 0 \\ factors & w_{Factor\_Goal} & 1 & 0 & 0 \\ subFacto & 0 & w_{Subfacto\_Factor} & 0 & 0 \\ Alternati & 0 & 0 & w_{Alternativ e\_Subfactor} & 1 \end{array} \qquad (1)$$

The Second scenario or *Network by content senario* has relevancy between content's criteria. Likewise, *Network by Content & Learner Interface Senario* as the third one had relevancy between content's criteria and learner interface's criteria. Thus, the super matrix shown in matrix 2 is suitable for both of these scenarios.

$$\begin{array}{lcccc} goal & 1 & 0 & 0 & 0 \\ factors & w_{Factor\_Goal} & 1 & 0 & 0 \\ subFact & 0 & w_{Subfacto\_Factor} & w_{Subfactor\_Subfactor} & 0 \\ Alternat & 0 & 0 & w_{Alternative\_Subfactor} & 1 \end{array} \qquad (2)$$

Because pairwise comparisons are not crisp, hence we use fuzzy preference programming to capture the weight of each matrix. In addition, to reach a global argument between decision makers, we use Group Fuzzy Preference Programming (GFPP) for deriving weight of a matrix which has been proposed by Mikhailov et al. [13]. By applying pairwise comparison between criteria with respect to their higher row, for each comparison from decision makers, we get triangular fuzzy number (TFN) like this: $a_{jl} = (l_{ij}, m_{ij}, u_{ij})$ where l_{ijk} and u_{ijk} are the lower and the upper bounds, representing the scope of the fuzziness of the fuzzy number, and m_{ijk} is core of the fuzzy number, corresponding to the maximum degree of membership, equal to one. Using the concept of α *-level sets* or α -cuts (Zimmerman [21]), for a given α -level, the interval version of the fuzzy preference programming (FPP) method tries to find a crisp priority vector $W = (w_1, w_2, ..., w_n)^T$, which satisfies approximately all interval constraints:

$$l_{ijk}(\alpha) \overset{\sim}{\leq} \frac{w_i}{w_j} \overset{\sim}{\leq} u_{ijk}(\alpha) . \qquad (3)$$

With solving the weighted GFPP model (4), the crisp priority vector W and the objective function C which is an indicator that measures the overall consistency of the judgments are obtained.

$$\begin{array}{l} \text{Max } C = \sum_{k=1}^{K} l_k \lambda \\ \text{s.t. } d_q \lambda_k + w_i - u_{ij}(\alpha) w_j \leq d_q \cdot \\ d_q \lambda_k - w_i + l_{ij}(\alpha) w_j \leq d_q , \quad \sum_{i=1}^{n} w_i = 1 , \\ w_i > 0 , i = 1,2,..., n ; \lambda_k > 0 , k = 1,2,..., K , q = 1,2,..., 2m \cdot \end{array} \qquad (4)$$

When the interval judgments are consistent, the maximum value of C is greater or equal to one. For inconsistent judgments C takes a value between one and zero that depends on the degree of inconsistency and the values of the tolerance parameters d_q .In addition l_k demonstrates the weight of decision maker k. In this paper we set l_k =1 (for all $k = 1,2,..., K$) and it means there are no differences between the priorities of all DMs. Here we have some pairwise matrixes which are enforced by GFPP model. Finally for each matrix we have preference vectors that are shown in equation 5.

$$w_{Factor\_Goal} = \{ w_1, w_2, \ldots, w_n \}, w_{Subfactor\_Factorl} = \{ w_1, w_2, \ldots, w_j \}, \tag{5}$$
$$w_{Subfactor\_Subfactorl} = \{ w_1, w_2, \ldots, w_i \}, w_{Alternative\_Subfactorl} = \{ w_1, w_2, \ldots, w_m \}.$$

At this time we have a super matrix that containes the above vectors and is shown in equation 3. For finall assessment,we apply equation 6 on the vectors.

$$w_{Subfactor\_Factorl} = w_{Subfactor\_Factorl} \times w_{Subfactor\_Subfactorl}$$
$$w_{Subfactor\_Subfactorl} = w_{Factor\_Goal} \times w_{Subfactor\_Factorl} \tag{6}$$

Equation 7. shows finall evaluation of E-learning system.

$$w_{Alternative\_Subfactorl} = w_{Subfactor\_Factorl} \times w_{Alternative\_Subfactorl} . \tag{7}$$

In the following section we represent a real case that has been done by *Fuzzy ANP* model.

4 Real Cases

To derive preference structure, and evaluate three E-learning systems in Iran, a survey was carried out. Data were gathered from students, the managers of these systems and also some software developers who were involved in developing E-learning systems. They filled out ANP questionnaires which were similar to AHP questionnaires. However for deriving preference structure we had a total number of 25 decision makers. About 50 percent of them were students of E-learning systems who were studying Information technology, 30% were engaged in developing E-learning systems and finally 20% of them were managers of E-learning systems.

After gathering questionaries ,the above methodology was applied to them. We had 25 decision makers who filled out all the questions. In this step we used three discussed scenario to capture preference among dimension and criteria.In this step the three main matrixes are determined (w_{Factor_Goal} , $w_{Subfactor\_Factorl}$ and $w_{Subfactor\_Subfactorl}$). The result of this preference among dimensins is shown in Table 2.

Table 2. Weights of dimensions and criteria

Criteria	Weight AHP scenario	Weight Network by Content	Weight Network by Content & Learner interface
Ease of use	0.146131	0.146131	0.146131
User-friendliness	0.093102	0.093102	0.116537
Ease of understanding	0.054445	0.07788	0.054445
Operational stability	0.068145	0.02124	0.068145
Up-to-date content	0.1798	0.1798	0.1798
Sufficient content	0.078786	0.122344	0.122344
Useful content	0.10324	0.10324	0.10324
Ease of discussion with Other learners	0.03416	0.03416	0.03416
Ease of discussion with teachers	0.049532	0.049532	0.049532
Ease of accessing shared data	0.072761	0.072761	0.072761
Controlling learning progress	0.08175	0.08175	0.08175
Recording learning performance	0.02725	0.02725	0.02725

By applying the above methodology on the three E-learning systems that are used in famous universities in Iran, final evaluation is performed. Two decision makers who were familiar with these three E-learning systems, judged the systems. Finally a matrix of 12 * 3 was created for ($w_{Alternative\_Subfactorl}$). By applying fuzzy ANP procedure, final evaluation is achieved, as illustrated in table 3.

Table 3. Evaluation result

University	***Scenario1*** Weight AHP	***Scenario 2*** Network by Content	***Scenario 3*** Network by Content & Learner interface
AUT	0.3723	*0.368*	*0.3831*
IUST	0.3723	*0.4037*	*0.4271*
REISU	0.5206	*0.5519*	*0.5284*

Table 3 shows that REIS University has the best score in all scenarios. AUT and IUST in the first scenario have the same grade but in the third and the second scenarios IUST is more applicable. All the above procedures were carried out by using Excel solver and a software program written in VB6.

5 Conclusions

The need for evaluating E-learning systems introduces the need to use various methodologies for assessment. In this paper we propose a MCDM approach by using ANP and fuzzy preference programming to evaluate E-learning systems. This approach considers some interrelations between E-learning criteria that is solved by ANP procedure .In this case we consider three scenarios and finally compare the result of applying these scenarios on three different E-learning systems. There can be other interrelation between criteria that were not considered in this paper. Also, for future researches we can consider priorities of decision makers in Group Fuzzy preference programming. Although the result shows that REIS University is the best of the three mentioned scenarios, but it also shows that regarding interrelations between criteria and applying ANP models can bring changes to the final results.

References

1. Mahdavi, H.F.: A heuristic methodology for multi-criteria Evaluation of web-based E-learning systems Based on User Satisfaction. Journal of Applied science 24(8), 4603–4609 (2008)
2. Alptekinl, E., Isiklar, G.: An Application of Quality Unction Deployment: Evaluation of E-learning Products in Turkey. In: 35th International Conference on Computers and Industrial Engineering (2004)
3. Wang, Y.-S.: Assessment of learner satisfaction with asynchronous electronic learning systems. Journal of Information & Management 41, 75–86 (2003)
4. Fresen, J.-W., Boyd, L.G.: Caught in the web of quality. International Journal of Educational Development 25, 317–331 (2005)
5. Soong, B., Chuan-Chan, H.: Critical success factors for on-line course resources. Computers & Education 36, 101–120 (2001)
6. Selim, H.M.: Critical success factors for e-learning acceptance: Confirmatory factor models. Computers & Education 49, 396–413 (2007)
7. Tzeng, H., Chiang, C.H.: Evaluating intertwined effects in e-learning programs: A novel hybrid CDM model based on factor analysis and DEMATEL. Journal of Expert Systems with Applications 32, 1028–1044 (2007)
8. Hsu, C.M., Chu, Y.: Development of design criteria and evaluation scale for web-based earning platforms. International Journal of Industrial Ergonomics 39, 90–95 (2009)
9. Colace, F., De Santo, M.: 36th Annual Evaluation Models for E-Learning Platform: an AHP approach Frontiers in Education Conference (2006)
10. Ru.en, Hsiang, Y.: Evaluation of the criteria and effectiveness of distance e-learning with consistent fuzzy preference relations. Journal of Expert Systems (2009)
11. Tzeng, G.H.: Hierarchical MADM with fuzzy integral for evaluating enterprise intranet web sites. Journal of Information Sciences 169, 409–426 (2005)
12. Shee, D.Y., Wang, Y.S.: Multi-criteria evaluation of the web-based e-learning system: A methodology based on learner satisfaction and its applications. Journal of Computers & Education 50, 894–905 (2008)
13. Mikhailov, L., Sadinezhad, S., Didehkhani, H.: Weighted Prioritization Models in the Fuzzy Analytic Hierarchy Process (submitted paper 2009)

14. Yüksel, I.: Using the analytic network process (ANP) in a SWOT analysis – A case study for a textile firm. Journal of Information Science 177, 3364–3382 (2007)
15. Saaty, R.W.: Decision Making in Complex Environments, Instructions on how to use the Super Decisions software (2003)
16. Anderson, T.: Theory and Practice of Online learning, pp. 46–53. E-book printed Athabasca University (2004)
17. Evan, N.: International Standards for HCI and Usability. International Journal of Human Computer Studies 55, 533–552 (2004)
18. Cheng, C.H.: Evaluating attack helicopters by AHP based on linguistic variable Weight. European Journal of Operational Research 116, 423–435 (1999)
19. Sum, P.c., Tsai, R.J.: what drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction. Journal of computer & Education 50, 1183–1202 (2003)
20. Saaty, T.L.: The Analytical Hierarchy process. RWS publication (1980)
21. Zimmermann, H.-J.: Fuzzy Set Theory and its Applications. Kluwer, Dordrecht (1990)
22. Saaty, T.L.: The Analytic Network Process: Decision Making with Dependence and Feedback. RWS Publications (1996)

An Ontology-Based Expert System and Interactive Tool for Computer-Aided Control Engineering Education

Isaías Garcia*, Carmen Benavides, Héctor Alaiz,
Francisco Rodríguez, and Ángel Alonso

Department of Electrical and Systems Engineering, Universidad de Leon,
Escuela de Ingenierías, Campus de Vegazana, C.P. 24071 León, Spain
{isaias.garcia,carmen.benavides,hector.moreton,
francisco.sedano,angel.alonso}@unileon.es

Abstract. This paper shows the experience in the development of an ontology-based expert system for the field of control engineering education. The particular characteristics of the knowledge in this domain and some of the conceptualization strategies used are shown. Also the prototype expert system built is described, showing how the ontology-based nature of the software supposes a benefit for people learning design techniques in the field of control engineering.

Keywords: Expert Systems, Computer-Aided Control Engineering Education Tools, Engineering Ontologies, Knowledge Engineering, Knowledge Representation.

1 Introduction

Computers have always been a valuable tool in engineering domains. Within control engineering, they have been used as a tool in every phase of a system's life cycle by means of the so-called CACE software (Computer Aided Control Engineering. Today, control engineering is demanding a shift in software applications toward highest levels of abstraction in the representation of the information [1]. The semantic content of the complex engineering knowledge must be explicitly stated and separated from code in the software applications in order to facilitate the interactions among machines and between humans and machines.

There is also a recent concern in the control engineering community about education and computer-aided education and training [2]. As a result, the research on control education software is very active [3], [4]. But there is a problem regarding this kind of software: there is a gap between the graphical, multimedia-enriched presentation of the information and the underlying theoretical concepts. This is so because the control concepts are not conveniently represented in the computer. The interpretation

* Corresponding author.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 264–273, 2010.

of the things happening in the user interface is left to the user, without any help from the software.

The work presented in this paper shows the experience on the development of an ontology-based expert system to be used as a computer aided education tool for the field of control engineering design. The aim is to fill the aforementioned gap between the elements in the user interfaces and the underlying theoretical concepts. The paper is organized as follows: Section 2 introduces the domain of application and the objectives of the expert system to be built. Section 3 is devoted to the presentation of the ontology, along with some conceptualization examples. Section 4 describes the ontology-based user interface, stressing the interaction between the application, the ontology and the user. Finally, section 5 presents some conclusions and future work.

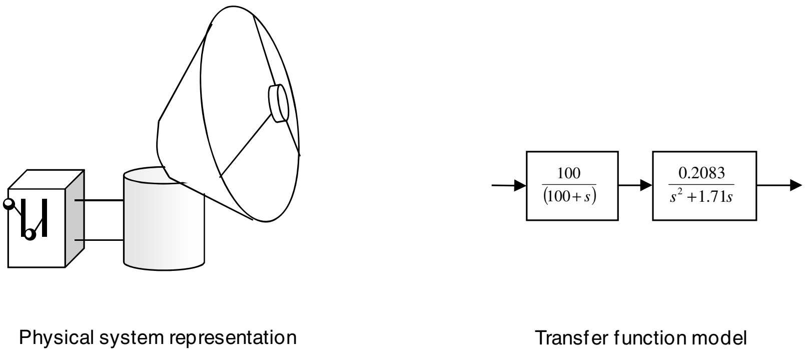
2 Domain of Study and Objectives of the Application

2.1 The Domain of Study

Control engineering deals with the analysis of the dynamic behaviour of systems (of any type) and the design of devices that, conveniently attached to the initial system, will make the resulting new system to perform better. The prominent characteristic of control engineering is the use of mathematical models of the physical systems on which the analysis and design tasks are carried out. There are many different models expressed at different levels of representation and with different objectives.

The model used in this work is the so-called transfer function model, where the dynamic behaviour of the system is represented by a polynomial quotient resembling the relationship among the output and the input signals (those polynomial quotients are called transfer functions).

Each of the subsystems that build a physical system has its own transfer function model. This transfer function is usually represented by a block containing the mentioned polynomial quotient. The blocks are arranged in a graph (a block diagram) where connections among them resemble the connections of the actual physical (sub)systems. Figure 1 shows an example of the physical representation of an antenna positioning system, including its corresponding transfer function block diagram.



Physical system representation

Transfer function model

Fig. 1. Transfer function model for the antenna positioning system

The design in control engineering is motivated by the necessity to alter the dynamic behaviour of a given system (for example, in order to control the position or the speed at which the antenna goes to the required position). In order to achieve this, a new element must be added to the system. Also, the current position of the antenna must be measured and compared to the desired value for the position by means of a feedback link, as shown in figure 2. The feedback loop and the extra element (called the controller) are the responsible for implementing the control of the antenna positioning system. Design in control is the process of choosing the appropriate controller to be added to the system. This element is also a dynamic system and so has its own transfer function system model of the form:

$$K \cdot \frac{(s-z_1)\cdot(s-z_2)}{(s-p_1)\cdot(s-p_2)} \tag{1}$$

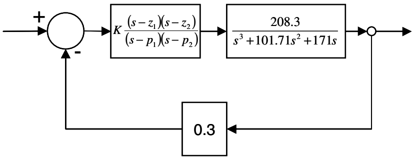

+
-
$K\frac{(s-z_1)(s-z_2)}{(s-p_1)(s-p_2)}$
$\frac{208.3}{s^3+101.71s^2+171s}$
0.3

Fig. 2. Controller and negative feedback loop

The design process is then the process of finding the suitable values for K and for z_1, z_2, p_1 and p_2, which are real numbers representing the roots of the numerator and the denominator of the transfer function of the controller.

The roots of the polynomials in the transfer function model of a system are very important and, in fact, the dynamic behaviour of that system can be obtained from the values of these roots. Control design techniques arrange these roots in the Argand diagram, representing complex numbers with the real component in the abscissa axis and the imaginary one in the ordinates. The design process tries to locate the roots of the total system transfer function in such positions that the obtained behaviour matches the one desired. The roots of the denominator, called the poles of the system, are the ones that determine the kind of evolution that the output signal will have.

In this design method, the parameters in the controller are adjusted to move the roots of the total system to the desired positions. The usual way to help in this task is representing the variation of the roots of the total system for variations in the gain of the controller, K. This graphical representation is called the root locus of the system and it offers lots of insight into the system and the effects of the different designs obtained (and that's why the design method is called "the root locus method").

2.2 Objectives of the Application

The main objective of this work is to build a prototype expert system based on a domain ontology to help in the learning of the task of controller design. The application will, at runtime, accept the problem description and design requirements, perform the design process automatically and present to the user the whole design process in a convenient way, including all the intermediate steps. It must also be able to answer questions the user may have about the design process or the meaning of the concepts that may have appeared.

3 Description of the Ontology and Its Conceptual Structures

From the point of view of the knowledge lying inside, an ontology is built upon two main structures:

- One reflecting the static knowledge, comprising the concepts found in the domain and the relations among these concepts.
- One storing the dynamic knowledge, comprising the conceptualization of the tasks describing the problem solving aspects of the domain.

3.1 The Static Knowledge of the Ontology

The starting point in control engineering practice is the mathematical, transfer function model of each of the systems involved in the design problem and their connections in the block diagram representing the total system. Every reasoning process and every design decision has to do with and can be obtained from the block diagram and what it contains.

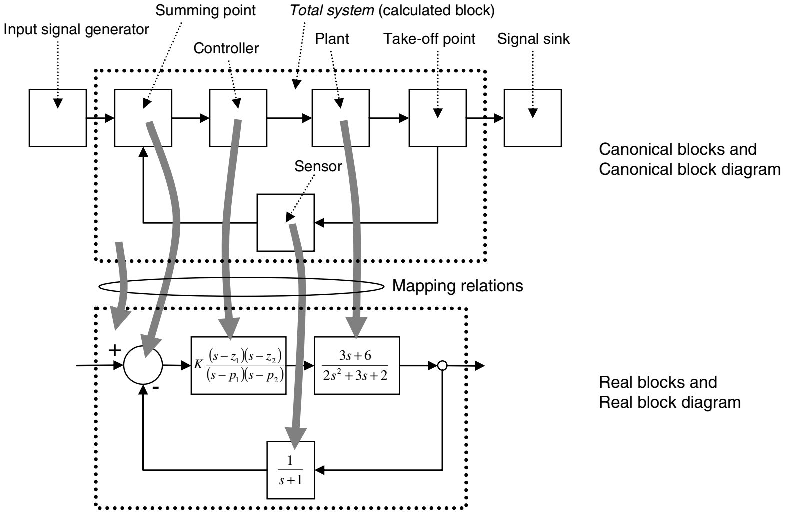


Fig. 3. Canonical and real block representations

The basic conceptual elements are then the blocks, along with their interconnection diagram, and the transfer functions inside them. Block diagrams have been represented by means of a graph of blocks with directed binary relations. Because knowledge about control engineering design can be stated without any real data about their transfer functions there must be a way to refer to the blocks in a general way. Thus, these blocks in the diagram have been represented with canonical block instances that will eventually be mapped to corresponding real blocks (figure 3).

In order the design process can evolve, some insight into the systems must be obtained and some decision must be taken based on this insight. Control engineering obtains all the relevant characteristics for gaining this insight from the polynomials of the transfer functions models. And so, an example of expression that may be found will be:

If the type of the Plant is greater than 1 then……

In this simple statement there are a number of important issues from the point of view of the conceptualization. First, it is worth noting how the canonical block "Plant" (that can be seen in figure 3) is used in the expression to represent design knowledge. Second, it can be seen how the conditions established over the values of the characteristics of the (sub)systems guide the analysis and design processes and decisions. Here, the characteristic "type" is applied to the block concept “Plant”. The transfer function of the system “Plant” will be obtained from the mapping described above. Then, the characteristic “type” must be obtained from the data stored in that transfer function. In order for this to be possible, the definition of how the type of a system can be obtained from its transfer function must be represented in the ontology.

The most basic conceptual structure is the one describing real and complex numbers. Then, polynomials and polynomial quotient concepts are built and, with the help of these structures, the transfer function models are also represented in the ontology. Concept composition is also a very useful conceptual schema used in the ontology. By using it, the concept “poles” may be defined as “the roots of the polynomial in the denominator of the transfer function model”.

Quantitative and qualitative characteristic play also a central role in the conceptual structure of the ontology. Quantitative characteristic may be defined with the help of other ontology concepts or may be represented by the function to call in order to get its value (as, for example, the case of the rising time of a given transfer function model). Qualitative characteristics are built by associating intervals in the possible values of quantitative characteristics or mathematical expressions to qualitative names as, for example, in the case of the stability of a system.

As well as the ones presented, there are some other important conceptualizations in the ontology not described here, like, for example:

- Representation of graphical objects.
- Communication with numerical calculus software, like MATLAB or MAPLE, including the function call and data type conceptualizations for bidirectional automatic data type conversion.
- The structures representing complex expressions on the characteristics.
- The representation of the controller structure, containing the design parameters.
- The representation of the different stages of the design process.
- The description of subsets of a set of concepts defined by fulfilling some characteristic value restrictions.

3.2 The Dynamic Knowledge of the Ontology

Dynamic knowledge comprises the knowledge structures describing the actions and procedures that describe the problem solving strategies (in this case, the representation of analysis and design processes involved in the root locus method).

As has previously seen, design knowledge in control engineering comprises different kinds of reasoning (procedural, heuristic, graphical, etc). The design task itself has an iterative nature, because each of the designs obtained must be tested against the model (by numerical simulation). If the design fails to satisfy all the initial design requirements then a new design must be obtained, and so on.

The design task may be divided into subtasks, many of them procedural but also some heuristic ones. The conceptualization of the dynamic knowledge is based on two main components:

- The description of the general architecture of the different stages found in the design task, including the control knowledge to go through these stages at the adequate moments and conditions.
- The description of each of the tasks and subtasks to be arranged as specified in the previous description.

The design evolves by a successive set of "design decisions". The design decisions are related most of the times with the assignation of a value to the parameters in the controller as described in section 2.1, though they can also be related to the structure of the controller, the suitability of a certain design method, etc.

Most of the tasks and control mechanisms are built by means of rules. Rules corresponding to the dynamical knowledge are organized in the ontology at three levels for task-subtask grouping and execution control purposes. The bottom level describes subtasks. Then a second level organizes and groups the subtasks into design tasks like “proportional gain design” or “lead zero design”. The conceptualization of tasks and subtasks at both levels is comprised of information about what preconditions must be satisfied in order the task can be executed, the concepts necessary to carry out the task and the list of actions that perform the given task.

These blocks defining design tasks are controlled by a higher control level also described by rules. This higher level controls the execution of the different design tasks. The conceptualizaton of the higher level rules consists of the preconditions necessary to achieve a given, lower level, task.

This structure of task blocks and control level, along with the rules they contain, are all automatically translated into JESS (Java Expert System Shell) [5] to perform the actual reasoning. The control of task execution is accomplished by asserting specific facts devoted to the control mechanism.

The condition part of the rules has the proper structure to be able to match the structure of triples comprising the knowledge base. The rule firing process may produce new facts regarding the evolution of the design process. These facts are used by the higher level rules to guide the processes of design and redesign.

In order the system can to reason about previous designs, and also show to the user the design rationale, some kind of account about time and evolution of concepts into

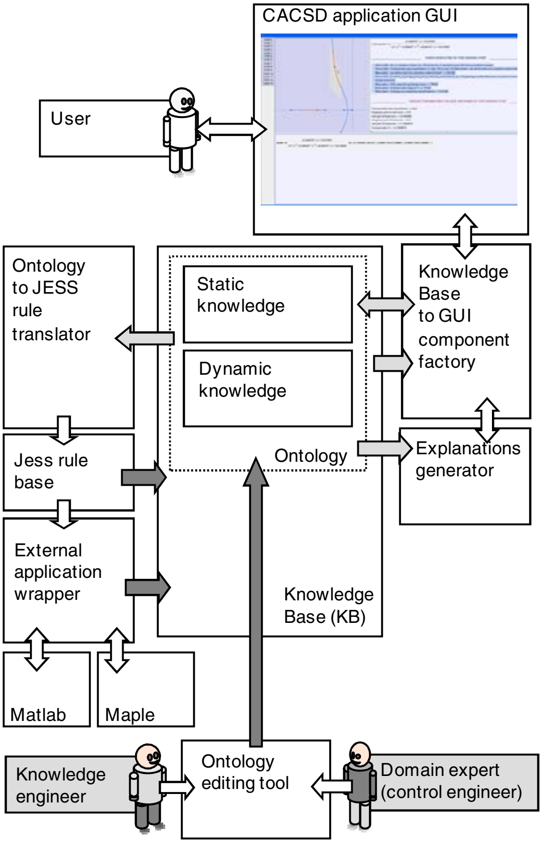


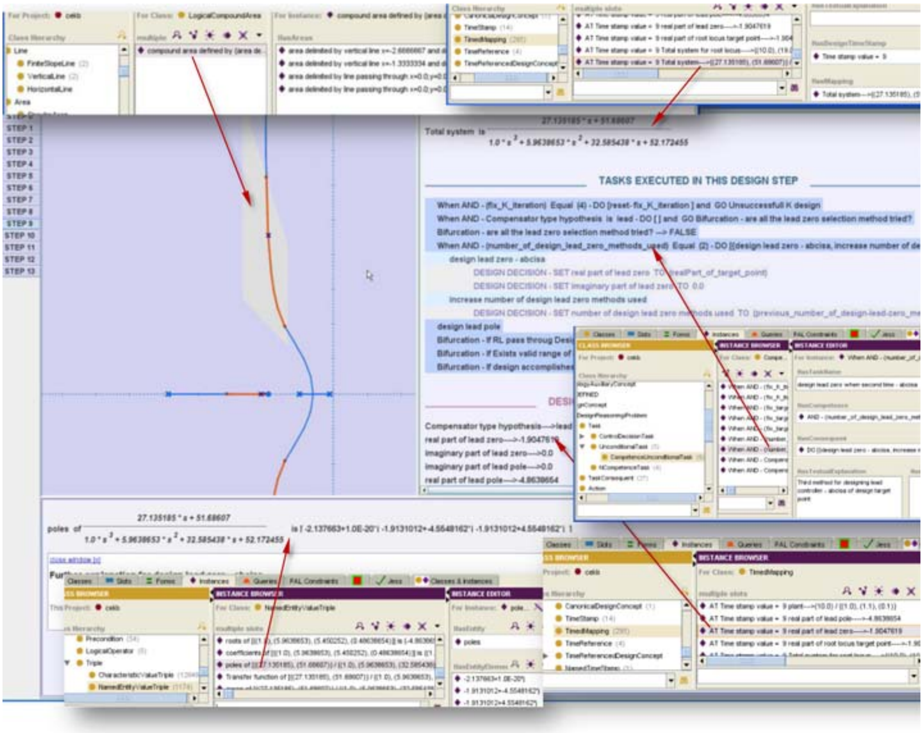
Fig. 4. Expert system and application schemas

time must be represented. This has been accomplished by introducing a time stamp and a series of mappings among concept instances and time stamps. Each time some design decision is taken the time stamp is incremented and new triples are generated. This strategy allows building the information necessary for the re-engineering tasks.

Figure 4 shows the schema of the overall system: the ontology, the expert system and the graphical application for the user. First, the domain expert and the knowledge engineer build the ontology structures reflecting the design of controllers. Then, these structures are translated into a rule base with Jess syntax. This forms an expert system able to solve control design problems. When some problem data are entered into the system, the rules in the rule base are fired until a solution is found. A huge amount of data is generated and stored into the knowledge base. Finally, the user can see and interact with the whole design process by using the graphical user interface, as explained in the next section.

4 Outline of the Graphical Application

Once the expert system has solved a given problem, a huge amount of data must be displayed to the user, this is done by using a very rich graphical user interface where the different steps of the design process are shown and where the user may ask questions or explanations about any of the elements appearing in the interface.



Total system is $\frac{27.135185 * s + 51.68607}{1.0 * s^3 + 5.9638653 * s^2 + 32.585438 * s + 52.172455}$

TASKS EXECUTED IN THIS DESIGN STEP

When AND - (fix_K_iteration) Equal (4) - DO [reset-fix_K_iteration] and GO Unsuccessfull K design
When AND - Compensator type hypothesis is lead - DO [ ] and GO Bifurcation - are all the lead zero selection method tried?
Bifurcation - are all the lead zero selection method tried? —> FALSE
When AND - (number_of_design_lead_zero_methods_used) Equal (2) - DO [[design lead zero - abcisa, increase number of de
design lead zero - abcisa
DESIGN DECISION - SET real part of lead zero TO (realPart_of_target_point)
DESIGN DECISION - SET imaginary part of lead zero TO 0.0
increase number of design lead zero methods used
DESIGN DECISION - SET number of design lead zero methods used TO (previous_number_of_design-lead-zero_me
design lead pole
Bifurcation - If RL pass throug Desi
Bifurcation - If Exists valid range of
Bifurcation - If design accomplishe

Compensator type hypothesis—>lead
real part of lead zero—>-1.9047619
imaginary part of lead zero—>0.0
imaginary part of lead pole—>0.0
real part of lead pole—>-4.8638654

poles of $\frac{27.135185 * s + 51.68607}{1.0 * s^3 + 5.9638653 * s^2 + 32.585438 * s + 52.172455}$ is [-2.137663+1.0E-20*i -1.9131012+4.5548162*i -1.9131012+4.5548162*i]

Fig. 5. Application GUI and ontology bound concepts

As can be seen on figure 5, the application generates a tab panel for each of the design steps that have been performed. The application window (for each of the tabs) has three main areas: The top left is a graphical panel where the graphical concepts are shown. The top right panel shows the names of the different systems involved in the design and their corresponding transfer functions for this design step. It also shows the tasks executed during that design step as well as the actions performed and design decisions taken for each of these tasks. Finally, the bottom panel is the place where the explanations resulting from the user's questions and interactions are displayed.

The most important characteristic of this application is the fact that every element in the GUI, from the terms in the text labels to the curves and points in the graphical panel, are the representation of an ontology concept and is linked to it, as represented by the arrows in figure 5. This binding of elements in the GUI to ontology concepts allows the automatic, on the fly, generation of menu items when the user interacts with one of these elements. The ontology structure is also the source for the explanations that are presented to the user. These explanations may be of different nature:

- Explanations about the definition of a term.
- Explanations about the reasons behind some assertion appearing in the GUI.
- Explanations about the design process, for example the reason for the execution of a given task or how a given parameter has been calculated.

5 Conclusions and Future Work

This work has presented the experience in the process of building an ontology for the field of control engineering design and an application for educational purposes based on this ontology. The benefits of ontologies for control engineering software construction come from the representation of static, definitional concepts on the one hand and dynamical knowledge on the other. The former benefit is due to the ability to represent and explain the control engineering language that exists on top of other scientific languages like mathematics. The latter benefit is produced by the possibility of storing the design rationale found in the tasks of analysis and design.

The ontology was built by using a frame-like formalism and the tool Protégé [6]. An automatic converter from the frame structures to the JESS syntax was also built in order to execute the definitional and the dynamical structures in the ontology. As future works, it should be attempted the possibility of building the ontology by using OWL [7] as the formalism. Besides using OWL, a rule language must also be used to complement the language and build the structures that can't be expressed with OWL. In this sense, Semantic Web Rule Language (SWRL) (Antoniou *et. al.*, 2005) is the most appealing candidate.

Some tests with a group of students enrolled in a basic course of control engineering have been carried out, with very promising results. The students were given the application and three solved root locus design problems, and were said to freely interact with the system. The most valued items were the following ones:

- The novelty of the approach.
- The advantages of having the whole design process organized and described in terms of tasks and design decisions, with a tab representing a design step.
- The possibility of seeing how and why the different controller parameters are calculated.
- The convenience of having the root locus plotted, and the possibility of showing different graphical elements on demand.
- The flexibility of the free interactivity.
- The possibility of obtaining explanations about why a given task was performed, or about definitions of concepts.

The only drawback pointed by the user was the relative simplicity of the provided exercises or the impossibility of getting a printed copy of the design process description.

References

1. Murray, R.M., Astrom, K.J., Boyd, S.P., Brockett, R.W., Stein, G.: Future Directions in Contol in an Information-Rich World. IEEE Control Systems Magazine 23(2), 20–33 (2003)
2. Antsaklis, P., Basar, T., DeCarlo, R., McClamroch, N.H., Spong, M., Yurkovich, S.: Report on the NSF/CSS Workshop on New Directions in Control Engineering Education. IEEE Control Systems Magazine 22(2), 53–58 (1999)
3. Dormido, S.: Control Learning: Present and Future. In: Proceedings of the 15th IFAC World Congress, Barcelona, Spain (2002)
4. Guzmán, J.L., Åström, K.J., Dormido, S., Hägglund, T., Piguet, Y.: Interactive Learning Modules for PID Control. In: Proceedings of the 7th IFAC Symposium on Advances in Control Engineering Education (2006)
5. Friedman-Hill, E., et al.: JESS (Java Expert System Shell) (2009), http://herzberg.ca.sandia.gov/jess
6. Gennari, J., Musen, M.A., Fergerson, R.W., Grosso, W.E., Crubézy, M., Eriksson, H., Noy, N.F., Tu, S.W.: The Evolution of Protégé: An Environment for Knowledge-Based Systems Development. Technical Report SMI-2002-0943, SMI, Stanford University (2002)
7. McGuinness, D.L., van Harmelen, F. (eds.): W3C consortium. OWL Web Ontology Language Overview, W3C Recommendation, February 10 (2004), http://www.w3.org/TR/owl-features/
8. Antoniou, G., Viegas Damásio, C., Grosof, B., Horrocks, I., Kifer, M., Małuszyński, J., Patel-Schneider, P.F.: Combining Rules and Ontologies. A survey. In: Małuszyński, J. (ed.) Deliverable I3-D3 (2005), http://rewerse.net/deliverables/m12/i3-d3.pdf

Exploiting Taxonomical Knowledge to Compute Semantic Similarity: An Evaluation in the Biomedical Domain

Montserrat Batet[1], David Sanchez[1], Aida Valls[1], and Karina Gibert[2]

[1] Universitat Rovira i Virgili
Department of Computer Science and Mathematics
Intelligent Technologies for Advanced Knowledge Acquisition Research Group
Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain
{montserrat.batet,david.sanchez,aida.valls@urv.cat}
[2] Universitat Politècnica de Catalunya
Department of Statistics and Operations Research
Knowledge Engineering and Machine Learning group
Campus Nord, Ed.C5, c/Jordi Girona 1-3, E-08034 Barcelona, Catalonia, Spain
karina.gibert@upc.edu

Abstract. Determining the semantic similarity between concept pairs is an important task in many language related problems. In the biomedical field, several approaches to assess the semantic similarity between concepts by exploiting the knowledge provided by a domain ontology have been proposed. In this paper, some of those approaches are studied, exploiting the taxonomical structure of a biomedical ontology (SNOMED-CT). Then, a new measure is presented based on computing the amount of overlapping and non-overlapping taxonomical knowledge between concept pairs. The performance of our proposal is compared against related ones using a set of standard benchmarks of manually ranked terms. The correlation between the results obtained by the computerized approaches and the manual ranking shows that our proposal clearly outperforms previous works.

Keywords: Semantic similarity, Ontologies, Biomedicine, Data mining.

1 Introduction

Semantic similarity quantifies how words extracted from documents or textual descriptions are alike. Similarity is typically understood as a degree of taxonomical proximity. For example, *bronchitis* and *flu* are similar because both are disorders of the respiratory system.

The assessment of semantic similarity has been a very active trend, with many direct applications such as, word-sense disambiguation [1], document categorization or clustering [2], ontology learning [3] or information retrieval [4].

In the biomedical field, similarity measures can improve the performance of Information Retrieval tasks [5]. In this domain, other authors have applied semantic similarity measures to discover similar protein sequences [6] or to the

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 274–283, 2010.

automatic indexing and retrieval of biomedical documents (e.g. the PubMed digital library) [7].

In general, semantic similarity computation is based on the estimation of semantic evidence observed in some knowledge source. That is, background knowledge is needed in order to measure the degree of similarity between a pair of concepts.

Domain-independent approaches [8,9,10] typically rely on WordNet [11], which is a freely available lexical database that describes and structures more than 100,000 general English concepts, which are stored as an ontology. An ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for combining terms and relations to define extensions to the vocabulary [12]. However, in specific domains it is more appropriate to use domain ontologies that have been built to describe precisely and completely the information related to a certain domain of knowledge. In biomedicine, there exist a growing number of ontologies that organize medical concepts into hierarchies and semantic networks like the Unified Medical Language System (UMLS) of the US National Library of Medicine. SNOMED-CT is one of the largest sources included in the UMLS. This ontology contains a number of medical concepts interrelated by different conceptual hierarchies corresponding to different scopes (procedures, substances etc.) (see section §3).

In the past, some classical similarity measures have been adapted to the biomedical domain [5] by exploiting medical ontologies (SNOMED-CT). In this paper, we expand this study to other classical similarity measures [13,14] based on the exploitation of the ontology's geometric model. Considering the limited performance obtained by previous attempts applied to the biomedical domain, we present a new method which is able to overpass them when evaluated against a set of biomedical and domain independent standard benchmarks.

The rest of the paper is organized as follows. Section 2 presents some similarity computation paradigms and the way in which they have been used in the past to deal with biomedical concepts. Section 3 presents classical similarity measures based on the ontology's geometric model and how can they be adapted to use SNOMED-CT as ontology. Section 4 introduces a new measure aimed to provide a better performance than previous approaches in the biomedical field. In section 5, all the presented measures are evaluated using a set of standard benchmarks composed by biomedical terms and general domain independent words. The final section will present the conclusions of this study and some lines of future work.

2 Related Work

In the literature, we can distinguish different approaches to compute semantic similarity between concepts according to the techniques employed and the knowledge exploited to perform the assessment. First, there are unsupervised approaches in which semantics are inferred from the information distribution of terms in a given corpus [15,16]. Statistical analysis and shallow linguistic parsing are used to measure the degree of co-occurrence between terms which is used as an estimation of similarity [17]. These measures need a corpus as general as possible in order to estimate social-scale word usage. However, due to their completely unsupervised nature and the lack of semantic analysis over the text, they offer a limited performance, specially when dealing with concrete domain such as biomedicine [5]. This is motivated by the lack of domain coverage of a general domain corpus and the difficulty of compiling a big enough domain corpus to obtain robust statistics.

Other trends exploit structured representations of knowledge as the base to compute similarities. Typically, subsumption hierarchies, which are a very common way to structure knowledge [18], have been used for that purpose. The evolution of those basic semantic models has given the origin to ontologies in which many types of relationships and logical descriptions can be specified to formalize knowledge [5]. In the biomedical field, many domain ontologies are available, being SNOMED-CT and MeSH some of the most successful examples.

From the similarity point of view, there exist ontology-based measures which combine the knowledge provided by an ontology and the Information Content (IC) of the concepts that are being compared. IC measures the amount of information provided by a given term from its probability of appearance in a corpus. Consequently, infrequent words are considered more informative than common ones. Based on this premise, Resnik [8] proposed to estimate the similarity between two terms as the amount of information they share in common. In a taxonomy, this information is represented by the Least Common Subsumer (LCS) of both terms. So, the computation of the IC of the LCS results in an estimation of the similarity of the subsumed terms. The more specific the subsumer is (higher IC), the more similar the subsumed terms are, as they share more information. Several variations of this measure have been developed [9,10].

These measures have been adapted by Pedersen et al. [5] to the biomedical domain by using SNOMED-CT as ontology and a source of clinical data as corpus. Those measures can be affected by the availability of the background corpus and their coverage with respect to the evaluated terms. Data sparseness (i.e. the fact that not enough data is available for certain concepts to reflect an appropriate semantic evidence) is the main problem [10]. Certain degree of pre-processing is needed with these corpora as they are typically unstructured natural language texts.

Without relying on a domain corpus, other approaches consider taxonomies and, more generally, ontologies, as a graph model in which semantic interrelations are modeled as links between concepts. Several measures have been developed to exploit this geometrical model, computing concept similarity as inter-link distance (also called Path Length) [13,19,14]. This idea was applied to the MeSH (*Medical Subject Headings*) semantic network [19] in order to improve the information retrieval by ranking document from MEDLINE, a corpus made up of abstracts of biomedical journal articles. Taking a similar approach, several authors [20,21] developed measures based on path finding in the UMLS hierarchy.

These measures have been applied to the biomedical domain by using SNOMED-CT as knowledge source [5]. The advantage of this kind of measures is that they only use a domain ontology as the background knowledge, so, no corpus with domain data is needed. As they only exploit the geometrical model of the ontology, no pre-processing is needed and, computationally, they are very efficient. Consequently, in this paper we focus on this kind of measures and study their behavior when they are applied to the biomedical field using SNOMED-CT as ontology.

3 Semantic Similarity Measures Based on the Taxonomical Structure

In an is-a hierarchy, the simplest way to estimate the distance between two concepts c_1 and c_2 is calculating the shortest *Path Length* connecting these concepts (i.e. the minimum number of links) [19].

$$pL(c_1,c_2) = min \ \# \ of \ is{-}a \ edges \ connecting \ c_1 \ and \ c_2 \tag{1}$$

Several variations of this measure have been developed such as the one proposed by Wu and Palmer [13]. They propose a Path Length measure that also takes into account the depth of the concepts in the hierarchy (2).

$$sim_{w\&p}(c_1,c_2) = \frac{2*N_3}{N_1+N_2+2*N_3}, \tag{2}$$

where N_1 and N_2 is the number of is-a links from c_1 and c_2 respectively to the Least Common Subsummer (LCS) c, and N_3 is the number of is-a links from c to the root ρ of the ontology. It scores between 1 (for similar concepts) to 0. If the taxonomy has multiple inheritance, the shortest path is selected.

Leacock and Chodorow [14] proposed a measure that considers both the shortest path between two concepts (in fact, the number of nodes N_p from c_1 to c_2) and the depth D of the taxonomy in which they occur (3).

$$sim_{l\&c}(c_1,c_2) = -\log N_p/2D \tag{3}$$

Classical approaches use those measures relying on WordNet [11] as the ontology to obtain the similarities between terms. However, due to the limited WordNet's coverage of biomedical terms [22], the performance obtained when comparing those specific domain concepts is poor [5]. So, as stated in the previous section, they have been applied to the biomedical domain by exploiting SNOMED-CT instead of WordNet.

SNOMED-CT (*Systematized Nomenclature of Medicine, Clinical Terms*) is an ontological/terminological resource distributed as part of the UMLS and it is used for indexing electronic medical records, ICU monitoring, clinical decision support, medical research studies, clinical trials, computerized physician order entry, disease surveillance, image indexing and consumer health information services. It contains more than 311,000 active concepts with unique meanings and formal logic-based definitions organized into 18 overlapping hierarchies: clinical findings, procedures, observable entities, body structures, organisms, substances, pharmaceutical products, specimens, physical forces, physical objects, events, geographical environments, social contexts, linkage concepts, qualifier values, special concepts, record artifacts, and staging and scales. Each concept may belong to one or more of those hierarchies by multiple inheritance. Concepts are linked with approximately 1.36 million relationships. Such a complete domain description makes it specially attractive to estimate the similarity between a pair of terms.

4 A New Measure to Compute the Semantic Similarity

In path length measures the minimum path length between a concept c_i and a concept c_j is taken as the sum of paths between each of the concepts and their LCS, so only ascendant is-a relations are considered to compute the path. In fact, the set of concepts encountered in these paths is, except the LCS, the set of non-shared superconcepts (upper concepts of a concept) of the pair of compared ones. That is, the amount of non-shared superconcepts is taken as an indication of distance. As it has been said, in those measures, in case of multiple inheritance from many is-a hierarchies, all the possible paths between two concepts are calculated but only the shortest one is selected. In consequence, the path length does not considers the full set of non-common superconcepts.

For complex taxonomies, such as SNOMED-CT, with thousands of interrelated concepts with several overlapping hierarchies that classify concepts and an extensive use of multiple inheritance, this kind of measures wastes a great amount of knowledge. For this reason, it seems reasonable that a measure that takes into account all the possible superconcepts of the evaluated concepts could provide more accurate assessments.

So, in the same sense as for the path length measures, we propose to consider all the non shared superconcepts as an indication of distance, that is, in case of multiple inheritance, evaluating all the superconcepts available for all the possible taxonomical paths. However, considering only the non-shared knowledge, we are not able to distinguish from concept pairs with very few or even no superconcepts in common from other ones which share more knowledge. For example, in Fig. 1 the number of non-common superconcepts of concepts c_1 and c_2, and concepts c_3 and c_4 is equal, and thus the distance between them are equal too. However, it makes sense that the distance between c_1 and c_2 is lower than between c_3 and c_4 due to the higher amount of shared superconcepts of the pair (c_1, c_2). This means that c_1 and c_2 are more specific terms that share more is-a relations in the taxonomy.

In order to take into account also the common knowledge, we define our measure as the relation between non shared and shared information, which is computed by dividing the non-common superconcepts by the sum of common and non-common superconcepts.

Given that our measure evaluates the amount of shared and non-shared ontological superconcepts as an indication of common and non-common information,

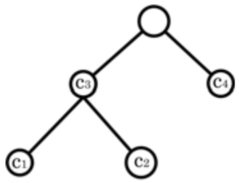


Fig. 1. Taxonomy example

conceptually, it follows a similar principle as information content measures, but performing the assessment in an intrinsic manner (i.e. concepts are more similar if they share more information -superconcepts- and vice versa). In order to fully capture the notion of IC and considering that the amount of explicit data available in a repository for a given concept is not linear to its IC [17], the inverted logarithm function is introduced to smooth assessments, obtaining the final similarity measure (4).

Formally, we define the full concept hierarchy or taxonomy (H^C) of concepts (C) of an ontology as a transitive is-a relation $H^C \in C \times C$, and we define $\mathcal{A}(c_i)$ as the set of ancestors of c_i including c_i itself, so that $\mathcal{A}(c_i) = \{c_j \in C | c_j \text{ is superconcept of } c_i\} \cup \{c_i\}$. Our measure can be defined as follows.

Definition:

$$sim(c_1, c_2) = -\log_2 \tfrac{|\mathcal{A}(c_1) \cup \mathcal{A}(c_2)| - |\mathcal{A}(c_1) \cap \mathcal{A}(c_2)|}{|\mathcal{A}(c_1) \cup \mathcal{A}(c_2)|} \qquad (4)$$

This definition introduces a desired penalization to those cases in which the number of shared superconcepts is small. Using the previous example (Fig.1) the similarity between concepts provides an approximation of the real situation. The result is bigger as bigger is the common information, and vice versa.

$$sim(c_1, c_2) = -\log_2 \frac{4-2}{4} = -\log_2 0.5 = 1$$

$$sim(c_3, c_4) = -\log_2 \frac{3-1}{3} = -\log_2 0.66 = 0.6$$

In the next section, the results obtained with the proposed measure and those presented in section 3 are compared, showing that considering all the common and non-common information between a pair of concepts results in more accurate estimation of semantic similarity in the biomedical domain.

5 Evaluation

The most common way of evaluating similarity measures is by using a set of word pairs whose similarity has been assessed by a group of human experts and computing their correlation with the results of the computerized measures.

For the biomedical domain, Pedersen et al. [5], in collaboration with Mayo Clinic experts, created a set of 30 word pairs referring to medical disorders. Their similarity was assessed in a scale from 1 to 4 by a set of 9 medical coders who were aware about the notion of semantic similarity and a group of 3 physicians who were experts in the area of rheumathology. For each pair, the averaged scores for each group of experts is presented in [5]. The correlation between physician judgements was 0.68 and between the medical coders was 0.78.

We used the same benchmark to evaluate the measures presented in this paper, using also SNOMED-CT as the domain ontology. Note that the term pair *"chronic obstructive pulmonary disease"* - *"lung infiltrates"* was excluded from the test bed as the later term was not found in the SNOMED-CT terminology.

Table 1. Correlations obtained for each measure against Physicians, Coders and both

Measure	Physician	Coder	Both
Path Length	0.33	0.39	0.39
Wu and Palmer	0.29	0.36	0.35
Leacock and Chodorow	0.45	0.58	0.55
Our measure	**0.60**	**0.79**	**0.73**

As some of the measures involved in the test compute similarity (Wu and Palmer and Leacock and Chodorow) and others evaluate dissimilarity (Path Length), for a consistent comparison, all the results have been converted into similarity values by changing the sign. Note that this conversion does not affect the result of the evaluation, since a linear transformation of the values will not change the magnitude of the resulting correlation coefficient.

The correlation values between the obtained results of the different similarity measures with respect to the human expert scores (including physicians, coders and the averaged scores of both) are presented in Table 1.

Considering the correlation values between human experts (0.68 for physicians and 0.78 for coders), it can be seen that Path Length-based measures offer a limited performance with correlations smaller than 0.45 and 0.59 respectively. These results show that poor results are obtained when estimating semantic similarity from the minimum inter-concept path in complex domain ontologies, such as SNOMED-CT, where multiple paths between concepts from several overlapping taxonomies are available.

On the other hand, our measure correlates significantly better than Path Length-based measures, being quite close or even better to the correlation between human manual evaluation: 0.60 vs 0.68 in the case of physicians, and 0.79 vs 0.78 with respect to medical coders.

In order to test the performance of our measure from a more general point of view, we also have evaluated each measure using a set of classical benchmarks [8,23] and WordNet[11] ontology as background. Concretely, Miller and Charles [23] proposed a benchmark which consists a set of 30 domain-independent word pairs. Resnik [8] replicated the experiment in order to obtain more accurate ratings by giving a group of experts the same set of noun pairs. Resnik computed how well the rating of the subjects in his evaluation correlated with Miller and Charles ratings. The average correlation over the 10 subjects was 0.884. This value is considered the upper bound to what one could expect from a machine computation on the same task [8]. The results for both benchmarks are presented in Table 2, showing a similar behavior as for the biomedical setting.

Table 2. Correlations obtained for each measure using general benchmarks

Measure	Resnik	Miller&Charles
Path Length	0.69	0.66
Wu and Palmer	0.82	0.81
Leacock and Chodorow	0.84	0.80
Our measure	**0.88**	**0.84**

6 Conclusions

In this paper, we studied the behavior of several ontology-based semantic similarity measures exploiting the geometrical model of ontologies when applied to the biomedical domain. The main advantage of those measures is that they do not rely on a domain corpus in order to extract semantic evidence. This is especially interesting in domains such as biomedicine in which the access to the huge amounts of data is typically difficult due to the sensitivity of medical information.

The main drawback is that their performance completely depends on the degree of completeness, homogeneity and coverage of the semantic links represented in the ontology. A priori, massive ontologies such as SNOMED-CT with thousands of interrelated concepts with a high degree of taxonomic specialization are a good knowledge source to apply those measures [1]. However, for other more specific domain ontologies with a limited scope, the graph model may be partial, which may bias the results of these measures. [24].

Even using a wide ontology like SNOMED-CT, classical approaches based on Path Length have shown a poor performance. Due to the inherent complexity of taxonomical links modeled in that ontology, with relationships of multiple inheritance between concepts, the computation of the minimum path between a pair of concepts only represents a partial view of the modeled knowledge.

In this paper, a new measure that takes into account the ratio between the shared and non-shared taxonomically related concepts of the compared pair of concepts is analyzed.

As shown in the evaluation, this measure is able to extract a robust semantic evidence from highly complex ontologies in biomedicine. The consideration of non common information between concepts and its relative importance with the common information provides a more accurate estimation of the semantic similarity. The results clearly outperform the classical similarity measures exploiting the taxonomic network complexity of SNOMED-CT. In fact, the correlation obtained by our approach with respect to human expert judgements is quite near to the maximum upper-bound (the inter-expert agreement, both for medical coders and physicians), showing the reliability of the obtained results. In addition, our measure has been also tested with domain independent benchmarks and using WordNet as ontology. In those experiments our approach also provided the best performance.

After this initial study, we plan to evaluate our approach with other medical ontologies such as UMLS or MeSH, and using other medical benchmarks [25]. In addition, a comparison with other measures based on other paradigms of semantic similarity like those based on the IC (instead of those based on the path length) could be interesting. Furthermore, other non-taxonomic relationships available in those ontologies can be also considered in the future as an statement of concept relatedness. Finally, tests with more reduced domain ontologies might show the degree of dependency of the quality of the results with respect to the ontology coverage.

Acknowledgments. This work has been partially supported by the Universitat Rovira i Virgili (2009 AIRE-04) and the DAMASK Spanish Project (TIN2009-11005) and the Spanish Government (PlanE, Spanish Economy and Employment Stimulation Plan). Montserrat Batet is also supported by a research grant provided by the University Rovira i Virgili.

References

1. Resnik, P.: Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research 11, 95–130 (1999)
2. Cilibrasi, R.L., Vitányi, P.M.: The Google similarity distance. IEEE Transaction on Knowledge and Data Engineering 19(3), 370–383 (2006)
3. Sanchez, D., Moreno, A.: Learning non-taxonomic relationships from web documents for domain ontology construction. Data Knowledge Engineering 63(3), 600–623 (2008)
4. Lee, J., Kim, M., Lee, Y.: Information retrieval based on conceptual distance in is-a hierarchies. Journal of Documentation 49(2), 188–207 (1993)
5. Pedersen, T., Pakhomov, S., Patwardhan, S., Chute, C.: Measures of semantic similarity and relatedness in the biomedical domain. Journal of Biomedical Informatics 40, 288–299 (2007)
6. Lord, P., Stevens, R., Brass, A., Goble, C.: Investigating semantic similarity measures across the gene ontology: the relationship between sequence and annotation. Bioinformatics 19(10), 1275–1283 (2003)
7. Wilbu, W., Yang, Y.: An analysis of statistical term strength and its use in the indexing and retrieval of molecular biology texts. Computers in Biology and Medicine 26, 209–222 (1996)
8. Resnik, P.: Using information content to evalutate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 95), Montreal, Canada, pp. 448–453 (1995)

9. Lin, D.: An information-theoretic definition of similarity. In: Shavlik, J.W. (ed.) Proceedings of the 15th International Conference on Machine Learning (ICML 98), Madison, Wisconson, USA, pp. 296–304. Morgan Kaufmann, San Francisco (1998)
10. Jiang, J., Conrath, D.: Semantic similarity based on corpus statistics and lexical taxonomy. In: Proceedings of the International Conference on Research in Computational Linguistics, September 1997, pp. 19–33 (1997)
11. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998), http://www.cogsci.princeton.edu/~wn/
12. Neches, R., Fikes, R., Finin, T., Gruber, T., Senator, T., Swartout, W.: Enabling technology for knowledge sharing. AI Magazine 12(3), 36–56 (1991)
13. Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proceedings of the 32nd annual Meeting of the Association for Computational Linguistics, New Mexico, USA, pp. 133–138. Association for Computational Linguistics (1994)
14. Leacock, C., Chodorow, M.: WordNet: An electronic lexical database. In: Combining local context and WordNet similarity for word sense identification, pp. 265–283. MIT Press, Cambridge (1998)
15. Etzioni, O., Cafarella, M., Downey, D., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Unsupervised named-entity extraction form the web: An experimental study. Artificial Intelligence 165, 91–134 (2005)
16. Landauer, T., Dumais, S.: A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review 104, 211–240 (1997)
17. Lemaire, B., Denhiére, G.: Effects of high-order co-occurrences on word semantic similarities. Current Psychology Letters - Behaviour, Brain and Cognition 18(1) (2006)
18. Gómez-Pérez, A., Fernández-López, M., Corcho, O.: Ontological Engineering, 2nd printing. Springer, Heidelberg (2004)
19. Rada, R., Mili, H., Bichnell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics 9(1), 17–30 (1989)
20. Caviedes, J., Cimino, J.: Towards the development of a conceptual distance metric for the UMLS. Journal of Biomedical Informatics 37, 77–85 (2004)
21. Nguyen, H., Al-Mubaid, H.: New ontology-based semantic similarity measure for the biomedical domain. In: IEEE conference on Granular Computing, pp. 623–628 (2006)
22. Burgun, A., Bodenreider, O.: Comparing terms, concepts and semantic classes in wordnet and the unified medical language system. In: Proc. of the NAACL 2001 Workshop: WordNet and other lexical resources: Applications, extensions and customizations, Pittsburgh, PA, pp. 77–82 (2001)
23. Miller, G., Charles, W.: Contextual correlates of semantic similarity. Language and Cognitive Processes 6(1), 1–28 (1991)
24. Cimiano, P.: Ontology Learning and Population from Text. Algorithms, Evaluation and Applications (2006)
25. Hliaoutakis, A., Varelas, G., Voutsakis, E., Petrakis, E.G.M., Milios, E.E.: Information retrieval by semantic similarity. Int. J. Semantic Web Inf. Syst. 2(3), 55–73 (2006)

Estimating Class Proportions in Boar Semen Analysis Using the Hellinger Distance

Víctor González-Castro[1,⋆,⋆⋆], Rocío Alaiz-Rodríguez[1],
Laura Fernández-Robles[1], R. Guzmán-Martínez[2], and Enrique Alegre[1]

[1] Dpto. de Ingeniería Eléctrica y de Sistemas
victor.gonzalez@unileon.es
[2] Servicio de Informatica y Comunicaciones
University of León, Campus de Vegazana s/n, 24071 León, Spain

Abstract. Advances in image analysis make possible the automatic semen analysis in the veterinary practice. The proportion of sperm cells with damaged/intact acrosome, a major aspect in this assessment, depends strongly on several factors, including animal diversity and manipulation/conservation conditions. For this reason, the class proportions have to be quantified for every future (test) semen sample. In this work, we evaluate quantification approaches based on the confusion matrix, the posterior probability estimates and a novel proposal based on the Hellinger distance. Our information theoretic-based approach to estimate the class proportions measures the similarity between several artificially generated calibration distributions and the test one at different stages: the data distributions and the classifier output distributions. Experimental results show that quantification can be conducted with a Mean Absolute Error below 0.02, what seems promising in this field.

1 Introduction

Artificial insemination techniques provide great advantages in the veterinary field. On the one hand, they allow farmers to work with a reduced number of animals, saving both time and money. On the other hand, it makes possible to get better individuals each generation. Companies that sell semen samples to farmers need to guarantee that they will be optimal for fertilization. There is a direct relationship between sperm fertility and the state of the acrosome: a sample containing a high percentage of spermatozoa with a damaged acrosome when is recollected will no be useful for fertilizing purposes. This assessment is traditionally carried out manually, using stains, which makes this process tedious, time-consuming, costly and what is more important, non objective.

There are some works that automatically characterize images of spermatozoa according to their membrane integrity by means of texture descriptors (e.g. [2]). Features are evaluated in terms of the achieved classification accuracy. In this

⋆ This work has been partially supported by the research project DPI2009-08424 from the Spanish Ministry of Education and Science.

⋆⋆ Corresponding author.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 284–293, 2010.

field, however, the aim is to estimate the proportion of damaged cells with no concerns about the individual classification of each one.

Unlike a typical supervised learning problem, the class prior probabilities estimated from the labeled training data cannot be considered representative of future samples since they are subject to vary due to factors like the animal/farm variability, or the manipulation and conservation conditions. It is well known that a mismatch between the test (real) class prior probabilities and those for which the classifier has been optimized, leads to suboptimal solutions. Different works have tackled this problem (e.g. [11,10]) from several perspectives, but always with the goal of improving the individual classification performance. Linguistics [5] or medicine [12] are just some fields where it has been applied.

To the best of our knowledge, only a few works cope with the problem of estimating the *a priori* probabilities – the actual class distribution – of unlabeled data sets (also known as quantification). They mostly aim to the analysis of a company's technical support logs where there are changes in trends[8]. Some previous work has also been conducted on veterinary applications [1] with promising results evaluated, though, on very small data sets.

The techniques proposed in the literature to estimate the class proportions are either based on the classifier confusion matrix [8] or on the posterior probability estimations provided by the classifier [1]. Forman has also explored a method based on the estimation of the class conditional probability densities [8], but it turned out to be outperformed by simple methods that rely on the confusion matrix.

When there is a shift in class prior probabilities between training and test sets, the data distributions as well as the *a posteriori* probability distribution also change. Our proposal to estimate the class distributions is based on indirectly quantifying the similarity between distributions (comparing the test set distribution with the distribution of several generated labeled sets). A distributional divergence metric (Hellinger Distance [7]) is applied at different stages of the classification process: (1) between data distributions and (2) between the *a posteriori* probability distributions.

The goal of this paper is: (a) to explore an information theoretic approach to automatically quantify the class distribution in a given semen sample and (b) to evaluate some quantification methods and check whether or not reliable estimations can be achieved for this specific application.

In this work, we consider an image data set of boar sperm samples and use a back-propagation neural network as a classifier.The rest of this paper is organized as follows: Sections 2 and 3 present the class distribution quantification methods assessed in this work. Experimental results are shown in Section 4 and finally Section 5 summarizes the main conclusions.

2 Class Distribution Estimation

Consider a binary classification problem with a calibration labeled data set $T = \{(\mathbf{x}^k, d^k), k = 1, \ldots, K\}$ where $\mathbf{x}^k$ is a feature vector, d^k is the class label with

$d \in \{0,1\}$[1] and $\widehat{Q}_i = P\{d = i\}$ is the class prior probability estimated from T. A classifier can be built using the available data in two steps: Firstly, it computes a soft output $\widehat{y}^k$ and based on it, makes the final hard decision $\widehat{d}^k \in \{0,1\}$.

Consider now an unlabeled test data set $U = \{(\mathbf{x}^l), l = 1, \ldots, N\}$ with unknown class distribution P_i and the decisions $\widehat{y}^l$ and $\widehat{d}^l$ provided by the classifier for each instance in the data set U. The naive approach to estimate its class distribution is to count the labels assigned by the classifier, what is referred as Classify & Count (CC) in [8]. It is considered here as a baseline for comparison purposes.

A brief description of proportion estimation methods based on the confusion matrix and the posterior probability estimation are provided in Sects. 2.1 and 2.2, respectively. The technique based on the distance between probability distributions is presented in Sect. 3.

2.1 Methods Based on the Confusion Matrix

Classifier performance can be summarized by its confusion matrix. Based on it and the ratio of labels assigned by the classifier to each class, different techniques have been proposed in [10,8] to estimate the unknown proportions of an unlabeled data set. Basically, the estimations are obtained by solving a following system of two (in a binary case) linear equations with respect to $\widehat{P}_i$.

The solution of the equation system, however, can be non consistent with the basic probability laws (i.e, values outside the interval $[0,1]$) as it has been highlighted in [8]. In a binary problem, it is suggested to clip the negative values to zero and fix the probability of the other class to one. This solution may be not satisfactory for real practical binary applications, though and moreover, there is no straightforward solution for general multi-class problems. As in [8] we will refer to this method as Adjusted Count (AC). A related method, Median Sweep (MS), computes several confusion matrices for different classification thresholds and finally, the class proportion is given as the median of the estimations derived from each confusion matrix

2.2 Estimation Of Priors Based on Posterior Probabilities

In [8] the application of techniques based on the posterior probability estimation is discarded arguing that, under a change in class prior probabilities, the classifier is not optimal anymore. Therefore, any estimation derived from the posterior probabilities given by the classification model would not be reliable.

This problem can be overcome with the following strategy. Given a model whose outputs y_i provide estimates of posterior probabilities, Saerens et al. [10] proposes an iterative procedure based on the EM algorithm in order to adjust the classifier outputs for the new deployment conditions without re-training the classifier. This is carried out by indirectly computing the new class prior probabilities, which is the goal in our work.

[1] Class-1 will also be denoted as the positive class and class-0 as the negative one.

We will denote this iterative approach as the Posterior Probability (PP) method, and we refer the interested reader to [1] for more details.

3 Hellinger Distance to Estimate Proportions

As it has been mentioned before, in this work we focus on problems where the class conditional densities are fixed, but the class prior probabilities may shift after the classifier calibration. When this happens, the joint probabilities $p(x, d = 0)$ and $p(x, d = 1)$ also vary and so the unconditional density $p(x)$ and the posterior probabilities $p(d = 0|x)$ and $p(d = 1|x)$.

Figs 1 and 2 show the effects of shifting class distributions on the data distribution $p(x)$ for a binary classification problem where each class is defined by a univariate gaussian distribution. Fig.1 depicts the joint probabilities $p(x, d = 0)$ and $p(x, d = 1)$ for the training dataset with class prior probability (Q_0, Q_1) and the data density $p(x)$. Fig.2 plots the data distribution for the test set when a change in the prior probabilities has taken place. We note that a shift in class proportions from training and test, implies also a significant change in the data distribution. Generating calibration data sets with different prior probabilities

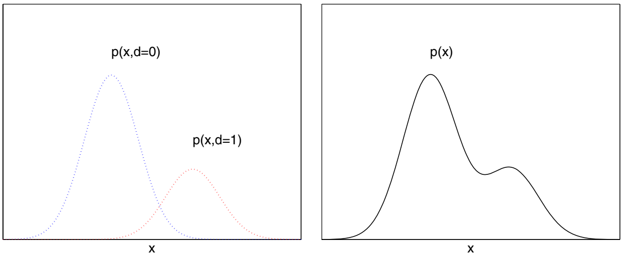


Fig. 1. Training data. Joint probabilities $p(x, d = 0)$ and $p(x, d = 1)$ (left) and unconditional density $p(x)$ (right) for prior class probabilities (Q_0, Q_1) equal to $(0.3, 0.7)$.

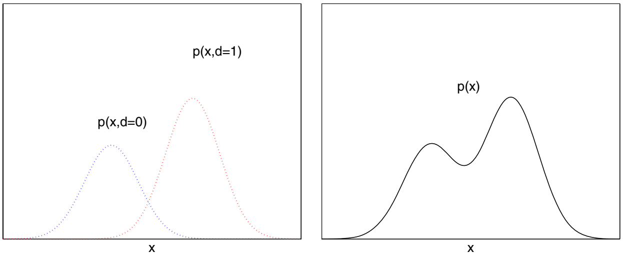


Fig. 2. Test (future) data. Joint probabilities $p(x, d = 0)$ and $p(x, d = 1)$ (left) and unconditional density $p(x)$ (right) for prior class probabilities (P_0, P_1) in the test set equal to $(0.6, 0.4)$.

and measuring differences between the calibration and the test data distribution would allow to detect these changes and therefore, estimate the new class proportions.

The Kullback-Leibler divergence as well as the χ^2 measure and the Hellinger distance are particular cases of the family of f-Divergences [7] that measure distributional divergence. Unlike the KL divergence or χ^2 measure that are both asymmetric, and not strictly distance metrics, the Hellinger Distance (HD) has interesting properties that make it appealing for our purpose. Recently, it has been receiving attention in the machine learning community in order to detect failures in classifier performance due to shifts in data distributions [6]. In particular, Cieslak and Chawla have shown that the HD measure is very effective in detecting breakpoints in classifier performance due to shifting class prior probabilities. In this work we address the problem of class distribution estimation following a HD-based approach.

The HD between two probability density functions $q(\mathbf{x})$ and $p(\mathbf{x})$ can be expressed as

$$H(q,p) = \sqrt{\int \left(\sqrt{q(\mathbf{x})} - \sqrt{p(\mathbf{x})}\right)^2 dx} \tag{1}$$

where HD is non negative and bounded (it ranges from 0 to $\sqrt{2}$) and is symmetric, i.e., $H(q,p) = H(p,q)$. Additionally, it is defined for whatever value of $p(x)$ and $q(x)$ and does not make any assumptions about the distributions themselves.

Similarity between the training data distribution and future distributions can be measured with HD by converting them into binned distributions with a probability associated with each of the b bins. The HD between the training data T and the unlabeled test data U with n_f features is then calculated as

$$H(T,U) = \frac{1}{n_f} \sum_{f=1}^{n_f} H_f(T,U) \tag{2}$$

where the distance between T and U according to feature f is computed as

$$H_f(T,U) = \sqrt{\sum_{i=1}^{b} \left(\sqrt{\frac{|T_{f,i}|}{|T|}} - \sqrt{\frac{|U_{f,i}|}{|U|}} \right)^2} \tag{3}$$

Note that b is the number of bins, $|T|$ the total number of training examples and $|T_{f,i}|$ the number of training examples whose f feature belongs to bin i. Similarly, $|U|$ and $|U_{f,i}|$ correspond to the same statistics for the test set.

Let us go back to the problem presented previously with its test data distribution depicted in Fig.2. It corresponds to a test set with class prior probabilities $P_1 = 0.4$ and $P_0 = 0.6$. Fig.3 plots the Hellinger distance between this test data distribution and different data distributions obtained from the available training data set. These calibration data sets differ in the class distributions (from $Q_1 = 0$ to $Q_1 = 1$). Note that the minimum HD is achieved for that calibration data set with the same a priori probabilities as the test set ($Q_1 = 0.4$).

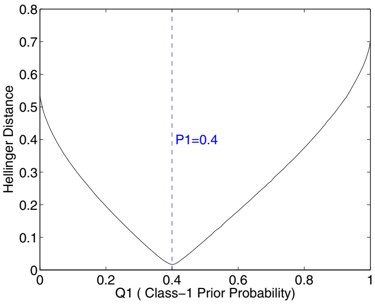


Fig. 3. Hellinger distance between the test data distribution and different calibration data distributions. The dashed line is the class-1 prior probability of the test data set.

In this work we address the problem of estimating the class proportions of a new unlabeled data set by finding the calibration data distribution for which the distance with the test data distribution is minimum. These artificially generated distributions can be extracted from the available training data set either by stratified sub-sampling, over-sampling or weighting the examples accordingly.

In real practical applications we usually face the problem of data sparseness. It is not uncommon to have a training data set that is not fully representative in all regions of the nf (number of features) dimensional space. In these cases, the curve in Fig. 3 (obtained from a large enough data set) turns noisy like the ones represented in Fig.4. This has been partially solved by downsampling the resultant curve and estimate the HD for a given training data set distribution by computing the median among that value and the nearest four points.

Note that we can measure the difference between a calibration set (with known labels) and the test set either by computing the HD between both data distributions (see Fig.4, left) or by computing the HD between the outputs assigned by a classifier that provides a posteriori probability estimates (see Fig.4, right).

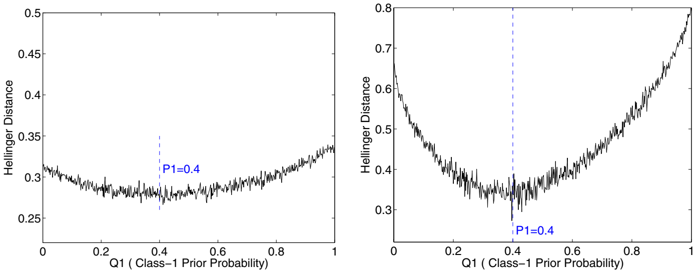


Fig. 4. Sperm cell data set. Hellinger distance (HD) between a test set and different calibration sets. HD between data distributions $p(x)$ (left). HD between posterior probability distribution (right).

Finding the differences between the classifier output distribution (the posterior probabilities) simplifies the problem, because distributional divergences are measured with data defined in a one dimensional space (for a two class problem) or in a $L - 1$ space for a general multi-class problem with L classes.

4 Experimental Results

In this section, several techniques that estimate the class distribution are evaluated in the context of a boar semen quality control application. We assess the performance of three quantifying approaches that rely on the Hellinger distance (HD), as well as AC, MS[2], PP and CC methods.

4.1 Sperm Cell Data Set

Experiments are carried out with images of boar sperm samples. The image data acquisition was conducted at CENTROTEC and under the guidance of researchers of the Faculty of Veterinary Sciences from the University of León. We use a data set with 1861 instances: 951 damaged and 910 intact spermatozoon heads. An example of these two acrosome states are shown in Fig. 5.

We use texture descriptors derived from the Discrete Wavelet Transform (DWT) to characterize the images. 20 features per image are derived from the co-occurrence matrix using the Wavelet Co-occurrence Features (WCF) [3]. For further details we refer the interested reader to [9].

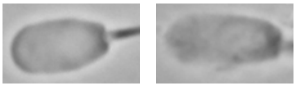


Fig. 5. Grey level images of intact (left) and damaged (right) acrosomes of boar sperm

4.2 Neural Network Classifier

Classification was carried out by means of a back-propagation Neural Network with one hidden layer and a logistic sigmoid transfer function for the hidden and the output layer. Learning was carried out with a momentum and adaptive learning rate algorithm. It is well known that classifier outputs provide estimates of class posterior probabilities when training is carried out minimizing some loss functions such as the mean square error used in this work [4].

Data were normalized with zero mean and standard deviation equal to one. The neural network architecture as well as the number of training cycles were determined by 10-fold cross validation. A two-node hidden layer network with 400 training cycles lead to the optimal configuration evaluated in terms of the overall misclassification rate, which was 4.27%.

[2] According to the methods based on the confusion matrix, we will only show the results for the AC method, since AC and MS presented very similar performance.

4.3 Performance Metrics

The mismatch between the real class distribution and the estimation provided by the different approaches assessed in this work is measured by means of the Mean Absolute Error (MAE) and the Mean Relative Error (MRE).

MAE focuses on the class of interest (class-1, positive or damaged cell class) and is defined as the absolute value of the difference between its actual prior probability and the estimated one: $MAE(\mathbf{P}, \widehat{\mathbf{P}}) = |P_1 - \widehat{P_1}|$. MRE measures the importance of the error and is defined as follows: $MRE(\mathbf{P}, \widehat{\mathbf{P}}) = |P_1 - \widehat{P_1}|/P_1$.

4.4 Quantification of the Damaged Sperm Cells

The following experiment was designed to assess the quantification methods CC, AC, PP, and three methods based on the Hellinger distance: HD of the inputs to the classifier (HDx), HD of the input data computing the median of the points in the curve (Median HDx) and HD of the outputs of the classifier (HDy).

Performance has been evaluated for a proportion of class-1 examples lower than 0.5 and a fixed set size (600 instances for training and 300 for test). The ratio of images from class-1 in the test set goes from 0.05 to 0.50 in 0.05 steps. For each scenario, results are the average of 5 training sets randomly extracted from the data set and, for each training set, 4 test sets were randomly extracted among the remaining examples. The confusion matrices required for AC were estimated exclusively from the training set by 50-fold cross validation, as suggested in [8].

Table 1. MAE and MRE errors of the HDx, Median HDx and HDy25 methods

	MAE			MRE (%)		
Ptest	HDx	Median HDx	HDy25	HDx	Median HDx	HDy25
0.05	0.039	0.024	**0.009**	77.32	48.56	**18.60**
0.10	0.032	0.026	**0.010**	31.70	26.36	**10.00**
0.15	0.023	0.022	**0.007**	15.33	14.42	**4.92**
0.20	0.021	0.022	**0.008**	10.38	11.16	**4.01**
0.25	0.029	0.029	**0.011**	11.40	11.41	**4.56**
0.30	0.031	0.036	**0.010**	10.22	12.13	**3.43**
0.35	0.025	0.027	**0.014**	7.24	7.74	**4.09**
0.40	0.027	0.032	**0.021**	6.75	8.04	**5.30**
0.45	0.044	0.033	**0.017**	9.80	7.27	**3.82**
0.50	0.071	0.054	**0.023**	14.23	10.87	**4.66**

In order to evaluate the quantification methods CC, AC, PP and HDy, the neural network described in Sect. 4.2 has been used as classifier. It has been trained with sets which have 25% class-1 elements – the center of our interval of interest –, so they will be referred to as CC25, AC25, PP25, and HDy25. The number of bins taken in the Hellinger-based methods has been fixed to 60.[3]

[3] This parameters was not very sensitive, as long as it was not too high so that there were few or none examples in each bin.

Results using the Hellinger-based methods (HDx, Median HDx and HDy25) are gathered and compared in Table 1. As it can be seen, the estimations given by HDy25 achieved the lowest error, in terms of MAE and MRE no matter the test class distribution. HDy25 allows to estimate the proportions with a MAE at the most of 0.023. Experimental results corroborate that computing the divergence between the output distributions yields better estimations than directly measuring raw data distribution divergences. A problem of data scarceness in the twenty-dimensional input space that does not appear in the one-dimensional output space may be the main reason.

With regard to CC, AC, PP and HDy25 (compared in Table 2), all methods provided satisfactory estimations in terms of absolute errors. Thus, for a test set with a ratio of damaged cells of 0.30, HDy achieves a MRE of 3,43% whereas PP and AC get a value around 4% what sounds very reasonable for the application field. It is remarkable that the absolute errors are quite stable across the different test distributions. It is also noticeable that the PP method works better for low probabilities, while AC method performs better in balanced distributions, and the HDy method is the best in the intermediate range of the interest interval.

Table 2. MAE and MRE of the HDy25, CC25, AC25 and PP25 methods

	MAE				MRE (%)			
Ptest	HDy25	PP25	AC25	CC25	HDy25	PP25	AC25	CC25
0.05	0.009	**0.006**	0.008	0.018	18.60	**12.45**	16.90	36.33
0.10	0.010	**0.008**	0.009	0.013	10.00	**7.64**	8.88	12.73
0.15	**0.007**	0.009	0.010	0.009	**4.92**	6.13	6.56	6.21
0.20	**0.008**	0.011	0.011	0.009	**4.01**	5.39	5.43	4.33
0.25	**0.011**	0.012	0.012	**0.011**	4.56	4.72	4.66	**4.36**
0.30	**0.010**	0.012	0.012	0.015	**3.43**	4.06	4.00	4.91
0.35	**0.014**	0.015	**0.014**	0.021	4.09	4.21	**3.87**	6.06
0.40	0.021	**0.017**	**0.017**	0.027	5.30	4.26	**4.22**	6.80
0.45	0.017	0.018	**0.016**	0.034	3.82	4.04	**3.61**	7.44
0.50	0.023	0.018	**0.017**	0.040	4.66	3.64	**3.49**	7.93

5 Conclusions and Future Work

This work tackles the problem of automatically quantifying the proportion of sperm cells with damaged acrosome in a given boar semen sample. We follow a computer vision-based approach that describes each cell by texture features and afterwards they are classified by a neural network.

Our novel proposal to estimate the class distributions is based on measuring the discrepancy between probability distributions (data and classifier output distributions) with the Hellinger distance. Satisfactory results for this application have been achieved with a mean absolute error lower than 0.023 for test semen samples with a proportion of damaged cells that vary from 0.05 to 0.50.

Comparisons of HD with AC,MS and PP show that: (a) all techniques get a significant improvement with respect to the baseline approach CC, (b) there is no single method that outperforms all the others in the whole probability range, (c) PP appears to be the best option for very low probabilities and AC when the test data set is balanced, whereas HD should be the choice for proportions in the middle of the interval.

Results suggest that a hybrid solution that either selects or combines the different estimation methods would allow to improve the total performance. Future work will follow this line as well as the study of visual tools that enable the analysis in a two dimensional space of the vast amount of performance results.

References

1. Alaiz-Rodríguez, R., Alegre, E., González-Castro, V., Sánchez, L.: Quantifying the proportion of damaged sperm cells based on image analysis and neural networks. In: SMO'08: Proc. of the 8th conf. on Simulation, modelling and optimization, pp. 383–388 (2008)
2. Alegre, E., González-Castro, V., Suárez, S., Castejón, M.: Comparison of supervised and unsupervised methods to classify boar acrosomes using texture descriptors. In: Proceedings ELMAR-2009, September 2009, pp. 65–70 (2009)
3. Arivazhagan, S., Ganesan, L.: Texture classification using wavelet transform. Pattern Recogn. Lett. 24(9-10), 1513–1521 (2003)
4. Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Oxford (1996)
5. Chan, Y.S., Ng, H.T.: Estimating class priors in domain adaptation for word sense disambiguation. In: ACL-44: Proc. of the 21st Int. Conf. on Computational Linguistics, pp. 89–96 (2006)
6. Cieslak, D., Chawla, N.: A framework for monitoring classifiers performance: When and why failure occurs? Knowl. Inf. Syst. 18(1), 83–108 (2009)
7. Csiszar, I., Shields, P.: Information Theory and Statistics: A Tutorial (Foundations and Trends in Communications and Information The). Now Publishers Inc. (December 2004)
8. Forman, G.: Quantifying counts and costs via classifcation. Data Min. Knowl. Disc. 17(2), 164–206 (2008)
9. González, M., Alegre, E., Alaiz, R., Sánchez, L.: Acrosome integrity classification of boar spermatozoon images using dwt and texture techniques. In: International Conference VipIMAGE 2007. Taylor & Francis, Abington (2007)
10. Saerens, M., Latinne, P., Decaestecker, C.: Adjusting a classifier for new a priori probabilities: A simple procedure. Neural Comput. 14, 21–41 (2002)
11. Xue, J.C., Weiss, G.M.: Quantification and semi-supervised classification methods for handling changes in class distribution. In: Proc. of the 15th ACM SIGKDD int. conf. on Knowledge discovery and data mining, pp. 897–906 (2009)
12. Yang, C., Zhou, J.: Non-stationary data sequence classification using online class priors estimation. Pattern Recogn. 41(8), 8 (2008)

Analysis of the Inducing Factors Involved in Stem Cell Differentiation Using Feature Selection Techniques, Support Vector Machines and Decision Trees

A.M. Trujillo[1], Ignacio Rojas[2], Héctor Pomares[2], A. Prieto[2], B. Prieto[2], A. Aránega[3], Francisco Rodríguez[3], P.J. Álvarez-Aranega[3], and J.C. Prados[3]

[1] FIBAO Bio-Health Research Foundation of Eastern Andalucía
[2] Computer Architecture and Computer Technology Department
[3] Anatomy and Human Embryology Department
University of Granada, 18071, Granada, Spain
irojas@ugr.es

Abstract. Stem cells represent a potential source of cells for regeneration, thanks to their ability to renew and differentiate into functional cells of different tissues. The studies and results related to stem cell differentiation are diverse and sometimes contradictory due to the various sources of production and the different variables involved in the differentiation problem. In this paper a new methodology is proposed in order to select the relevant factors involved in stem cell differentiation into cardiac lineage and forecast its behaviour and response in the differentiation process. We have built a database from the results of experiments on stem cell differentiation into cardiac tissue and using this database we have applied state-of-the-art classification and predictive techniques such as support vector machine and decision trees, as well as several feature selection techniques. The results obtained are very promising and demonstrate that with only a reduced subset of variables high prediction rates are possible.

1 Introduction

Stem cells are a special type of undifferentiated cells that are capable of dividing indefinitely without losing its properties and get specialized cells through the process known as cell differentiation [1] [2]. In human and higher mammals, stem cells sustain tissues composed of cells that have a limited lifespan and lose the ability of self-renewal such as blood and epidermis cells, by providing new cells differentiated.

Stimuli that can induce the differentiation of the stem cells into the cardiac lineage are very broad and varied [3]. In this context, growth factors play an important role in cell differentiation. These factors are proteins that bind to specific receptors on the cell surface leading to the activation of cell proliferation or differentiation.

Many articles are focused on experimental studies about stem cell differentiation [4] [5] [6] [7], in all of them the role of the growth factors is specially emphasized. Moreover, there are different mathematical models applied in this field of biomedicine that offer a description of the stem cell differentiation problem and simulate the effects of the growth factors on this process [8] [9] [10] [11]. The main limitation of these

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 294–305, 2010.

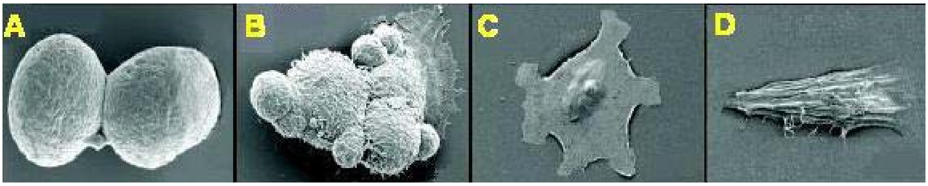


Fig. 1. Different stages that occur in the stem cell differentiation process into cardiomyocytes: (A,B) start differentiation process, (C) cardiopoietic cell and (D) cardiomyocyte cell

models is the absence of a detailed study and description of the variables or factors considered and the influence of them on the differentiation of the stem cells. This generates a great diversity and variability in the experimental and clinical results. Therefore, it is necessary the development of predictive models that facilitate the design of experiments and contribute to the understanding of the stem cells' behaviour.

This paper proposes the combined use of feature selection methods and the application of several so-called intelligent techniques to predict the differentiating response of various types of stem cells into cardiac lineage when they are stimulated with exogenous chemical and physical factors. The rest of the paper is organized as follows: in Section 2, we present the database we have built and on which all our experiments are based. Section 3 deals with different techniques we have implemented to select those features of the database that provide more information about the differentiation process. Sections 4 and 5 present the results obtained when using decision trees and support vector machines, respectively. Finally, some conclusions are drawn in Section 6.

2 Database

The database used in the predictive process has been built within the research group CTS-107 of the Junta of Andalucia "New Technologies for Biomedical Research", led by Professor Antonia Aránega, who is authorized to work with human embryonic stem cells by the national bioethics committee.

The database is composed of all relevant data that may guide the differentiation of stem cells into cardiac lineage. The input variables are all parameters concerning the morphological and functional differentiation, resulting from the action of external factors, including the cellular elements involved, the inducing factors, patterns of gene expression and proteomics, etc. The output variable has been generated by averaging three quality levels which weight the functionality and appropriate morphology of the cells after differentiation and the expression of markers which are characteristic of myocardial cells. This output takes values between 0 and 10, where the value 0 is the worst level of quality and the value 10 is the highest possible quality in the differentiation process.

To develop the database, this research group has obtained information from scientific papers published in recent years in relation to the differentiation of stem cells into the cardiac lineage, using the National Institute of Health (NIH) Pubmed database. The final database is made up by 282 samples of data, consisting of 14 input variables (see Table 1) and one output variable.

Table 1. Set of features/input variables that makes up the database

V_1. Origin of species	V_8. Cultivation Medium
V_2. Stem cell type	V_9. Serum
V_3. Source extraction	V_{10}. Support growth
V_4. Cultivation time until use	V_{11}. Target organism
V_5. Using just only one growth factor	V_{12}. Overall quality of the initial stem cells
V_6. Using two growth factor	V_{13}. Number of patients
V_7. Using three growth factor	V_{14}. Treatment time

With the aim of simplifying the processing of these data by the feature selection and learning algorithms, the input variables of the initial database have been transformed into quantitative variables classified into categories. The output variable has been interpreted from two viewpoints: as a gradual or multi-class output, where the quality of differentiation is expressed with values between 0 and 10, and a binary output, which only determines whether the process of directed differentiation has been successful or not. The binary interpretation of the output variable has been carried out in order to provide medical research groups a first approximation to determine whether the differentiation process has been successfully completed or not. In this way, the gradual output allows us to specify the quality of the differentiation process achieved.

Therefore, in this paper the proposed methodology for selecting the relevant features or input variables, and the classification or predictive process, has been carried out for both the binary output variable and the multi-class output variable.

3 Feature Selection

In stem cell differentiation problems, feature selection is a key previous phase. This process allows us to find the most significant factors in the system. In this section a robust methodology for feature selection is proposed to find the relevant variables in the stem cell differentiation process, for the two types of classifications, binary and multi-class classifications.

Several well-known feature selection techniques can be applied to the initial data set: Correlation-based feature selection (CFS) methods [14], Information Gain (IG) [15], Gain Ratio (GR) [15], Symmetrical Uncertainty (SU) [15], the Chi-Squared test, Consistency based feature selection methods [16], the OneR algorithm [17] and Relief [18].

CFS and Consistency based feature selection methods only provide a subset of features that has higher predictive power in the classification process. Other methods generate an ordered ranking of features depending on a certain weight assigned to each of them, according to the specific criteria followed by every particular algorithm. So, to deal with all of the selection methods in the same way, the subsets of variables obtained by CFS and the Consistency base feature selection algorithm have been associated a weight and an equal-distributed ranking in the interval [0, 1].

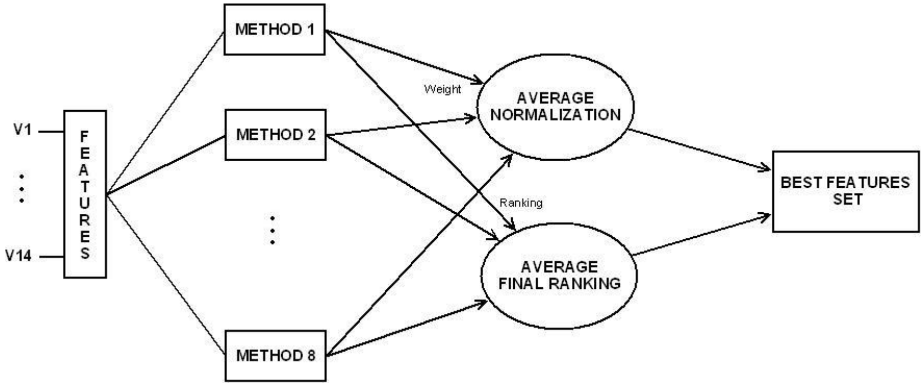


Fig. 2. Methodology proposed for feature selection

Table 2. Summary of results obtained by our proposed feature selection method

Binary Feature Selection		Multi-Class Feature Selection	
Ranking Index	Weight Index	Ranking Index	Weight Index
V_3	V_3	V_7	V_7
V_7	V_7	V_3	V_3
V_6	V_6	V_{10}	V_6
V_5	V_5	V_6	V_{10}
V_2	V_2	V_5	V_5
V_{10}	V_{10}	V_1	V_2
V_1	V_1	V_2	V_1
V_{11}	V_{11}	V_9	V_9
V_9	V_9	V_{11}	V_{11}
V_8	V_8	V_8	V_8
V_4	V_4	V_{12}	V_{14}
V_{14}	V_{14}	V_{13}	V_{12}
V_{13}	V_{13}	V_{14}	V_{13}
V_{12}	V_{12}	V_4	V_4

The procedure proposed in this section combines the 8 algorithms we have just mentioned and calculate the average importance of each of the 14 input variables according to two criteria: weight or importance assigned by the selection method and the ranking position produced by the method.

As can be seen in the left part of Table 2, for binary classification we have obtained exactly the same list of features using both types of criteria. This list emphasizes the importance of variables V_3, V_7, V_6, V_5, V_2 and V_{10}, whose values are above the mean, as the parameters that better determine the behaviour of the system.

For multi-class classification (right part of the table) the lists of features obtained for every criterion are very similar. Despite there is a very tiny difference, it has no influence in the stem cell differentiation problem: the final set of variables above the mean is identical in both cases V_7, V_3, V_6, V_5 and V_{10}. In Fig. 3 a graphical representation of the importance values obtained for each feature under both criteria is shown.

From a biomedical point of view, these conclusions suggest that the stem cell type, the source of extraction, the growth factors and the type of medium used in the experiments are the key issues in the differentiation process of the stem cells into cardiac tissue.

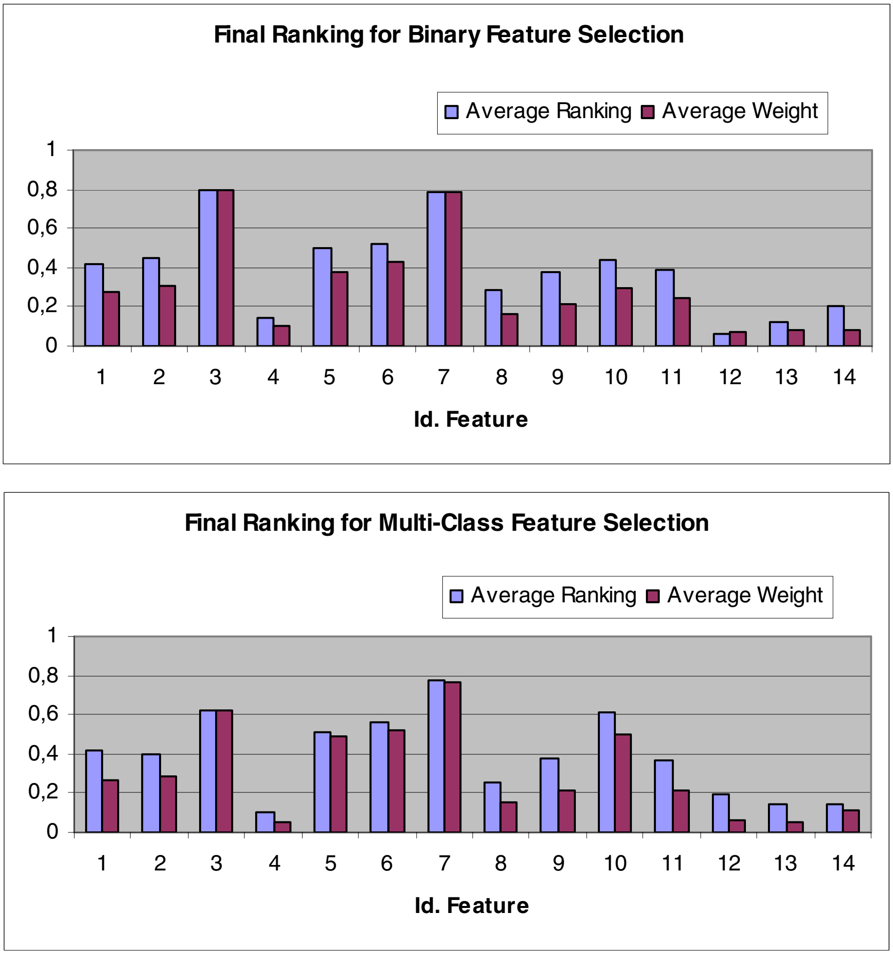


Fig. 3. Importance values for each feature under the two criteria: ranking and weight associated. All values are normalized in the range [0, 1].

These results provide valuable information about the stem cell differentiation process, because they determine the influence of different factors on the process. Thus, research groups that work on specific stem cell differentiations may focus their experiments on those variables that best describe the behaviour of the system under consideration.

Finally, based on this feature selection method, we could construct predictive models to provide information to medical research groups about the problem of stem cell differentiation. These models are expected to provide good accuracy and easily interpretable information. In the next sections, two predictive algorithms used in classification problems are proposed: decision trees and support vector machines.

4 Decision Trees

The characteristic of decision trees is their ability to transform a complex decision-making process in a series of simple decisions and thereby, providing an easily interpretable model [19]. This tree structure is composed of leaves and nodes: tree nodes represent input variables used in the problem under consideration and leaves are associated with a finite collection of values of the input variables. The tree is grown starting from the most meaningful variable and proceeding with other variables at lower nodes of the tree.

Many methods have been developed to construct decision trees from collections of data. The most popular is the Interactive Dichotomizer 3, ID3, proposed by Quinlan [20]. Models based on this approach use the concept of entropy to evaluate the discriminatory power of each attribute or variable input. The main problem of ID3 is that it can only deal with categorical (symbolic) data. So, more advanced versions emerged such as Classification and Regression Trees (CART) [21] and C4.5 [13], which are able to process both symbolic and non-symbolic data (continuous).

In our case, the database consists of a mixture of numerical and symbolic values. Therefore, we have decided to use as classifier a C4.5 decision tree. This algorithm examines the normalized information gain (difference in entropy) that results from choosing an input variable for splitting the data. The attribute or input variable with the highest normalized information gain is the one used to make the decision.

Other decisions we have adopted for building the decision trees are the use of binary division for symbolic variables and not using any error reduction during the pruning process. The first option improves the quality of the predictive model since our database is made up by a majority of symbolic variables. With the second option we could obtain models with a lower average absolute error, but at the expense of increasing the size of the generated tree, which makes it more difficult the interpretation of the model by the group of medical researchers.

The ranking of features obtained in section 3 has been applied for the creation of the decision trees. So, decision trees with different complexities have been generated according to the number of features used in its creation. More specifically, a first model was created using the first two features of the list. Subsequently, another new model was created making use the first three features of the list, and so on until the final model in which the complete set of input variables is involved. Fig. 4 shows an example of decision tree generated using the four most significant variables.

From a practical point of view, we can estimate that a decision tree classifier is good when there is a suitable balance between the success rate, which determines the quality of the model, and the size of the tree, since trees with a medium or small size represent easily interpretable graphical models and also provide valuable information about the biological problem.

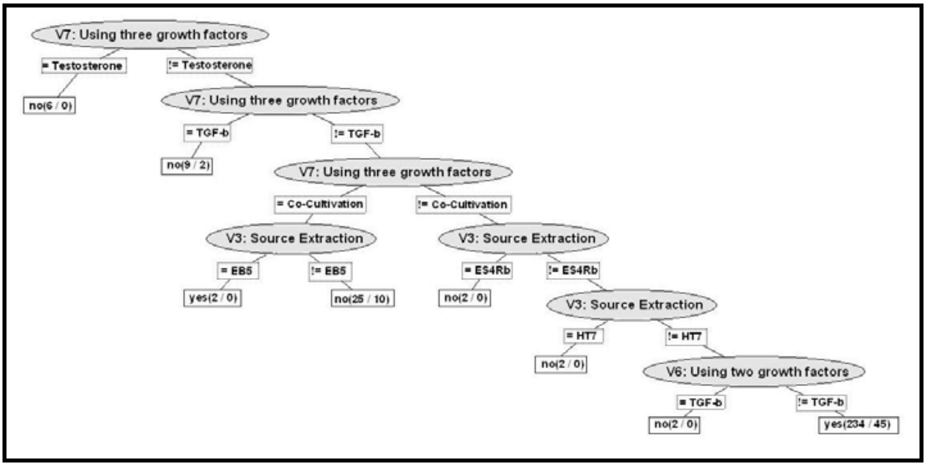


Fig. 4. Tree structure for the C4.5 decision tree generated with the four best ranked features using binary classification: V_3, V_7, V_6 and V_5. Values in parentheses in the leaf nodes of the tree determine the number of samples classified in that node and which of them are wrongly classified.

Table 3. Summary of results obtained using decision trees

	Binary Classification			Multi-Class Classification		
#Features	Success Rate Training	#Leaves	Tree Size	Success Rate Training	#Leaves	Tree Size
2	79.07%	7	13	43.97%	13	25
3	79.78%	8	15	52.12%	23	45
4	---	---	---	58.15%	28	55
5	---	---	---	55.31%	22	43
6	80.49%	9	17	57.80%	24	47
7	---	---	---	59.57%	28	55
8	81.91%	13	25	63.82%	45	89
9	84.04%	17	33	60.99%	29	57
10	83.68%	14	27	60.58%	29	57
11	---	---	---	65.95%	43	85
12	84.39%	16	31	69.14%	47	93
13	---	---	---	69.85%	52	103
14	84.75%	18	35	63.12%	56	111

The results obtained are summarized in Table 3. In the binary classification case, on some occasions the decision trees generated with different number of features are practically identical due to the fact that the algorithm itself rejects the use of certain variables in order to maintain a good success rate and not increase the complexity of the generated model. This issue also affects to that fact that we don't get better results after using a certain number of features to create the models.

In the multi-class classification case, the algorithm obtains less success rate than that obtained for binary classification, due to the complexity of the problem now to be solved. These trees have a large size, which means that their interpretation will be more complex. However, these trees offer more detailed information about the quality of the stem cell differentiation process into the cardiac lineage, because they not only determine whether the cellular differentiation has been successful or not, as in the case of trees created for binary classification, but also they specify the quality of the result.

Finally, it should be mentioned that, as expected, trees created using the best ranked features present higher rates of success; this reveals the importance of a powerful feature selection process, especially, in biological problems such as stem cell differentiation.

5 Support Vector Machines

A Support Vector Machine (SVM) [22] is a supervised learning algorithm developed by Vladimir Vapnik and his co-workers at AT&T Bell Labs in the mid 90's. Since their inception, SVMs have continuously been shown to outperform many prior learning algorithms in both classification and regression applications. In fact, the elegance and the rigorous mathematical foundations from optimization and statistical learning theory have propelled SVMs to the very forefront of the machine learning field within the last decade. A SVM is specifically formulated to build an optimal separating hyperplane in such a way that the margin of separation between two classes is maximized:

$$\text{minimize}\left(\|\vec{\mathrm{w}}\|^2+\gamma\sum_{i=1}^{N}\xi_i\right)$$

subject to

$$y_i\left(\vec{w}'\cdot\phi\left(\vec{x}_i\right)+b\right)\geq 1-\xi_i,\ \ \xi_i\geq 0,\ \text{i=1...N}$$

where N is the number of I/O samples of the form $\left(\vec{x}_i, y_i\right)$, $\vec{w}$ and b are the parameters defining the hyperplane, γ is a regularization parameter and ϕ a non-linear mapping for the input variables. Please, refer to [23] for a more detailed explanation about this computational paradigm.

SVMs are becoming popular in a wide variety of biological applications e.g. in classification and validation of cancer tissue samples using microarray expression data [24], prediction of mitochondrial proteins [25], species recognition [26], prediction of protein structural classes [27], etc.

We have used gaussian kernels (i.e. our non-linear mapping will be a gaussian) and we have optimized the non-linear parameters of the model using five-fold cross-validation in order to obtain the best model among all possible that determine the behaviour of the system. The subset of variables obtained after the feature selection procedure has been used for the evaluation of the SVM predictive models for binary and multi-class classification. In the same way, the generation procedure for the different SVM models, using the list of variables obtained of feature selection process was identical to that performed for decision trees.

Table 4. Summary of result for support vector machine

	Binary Classification		Multi-Class Classification	
#Features	Success Rate Training	#Support Vectors	Success Rate Training	#Support Vectors
2	77.65%	173	48.22%	262
3	83.68%	155	51.41%	262
4	82.26%	171	56.73%	243
5	83.68%	179	58.51%	242
6	85.46%	200	62.41%	251
7	87.94%	214	61.70%	251
8	90.78%	217	67.73%	269
9	94.32%	241	80.14%	264
10	95.39%	243	75.53%	262
11	97.87%	237	83.33%	264
12	96.03%	258	84.04%	264
13	98.96%	239	84.04%	264
14	98.93%	242	87.58%	262

For a detailed analysis of how this type of learning and classification algorithm works in biological problems, we have done a study based on the success rate of the model and its complexity, that is, the number of support vectors. In this case, the model does not provide an intuitive and interpretable graphical representation to researchers, that is, the model acts as a black box.

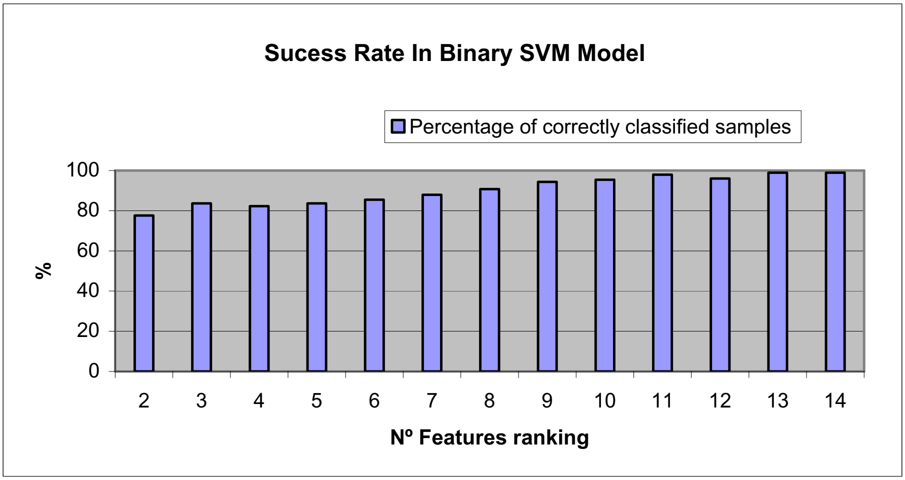


Fig. 5. Success rate for SVM predictive models created for binary and multi-class classifications. For their creation it has been used a variable number of features from the ranking order generated by the feature selection method.

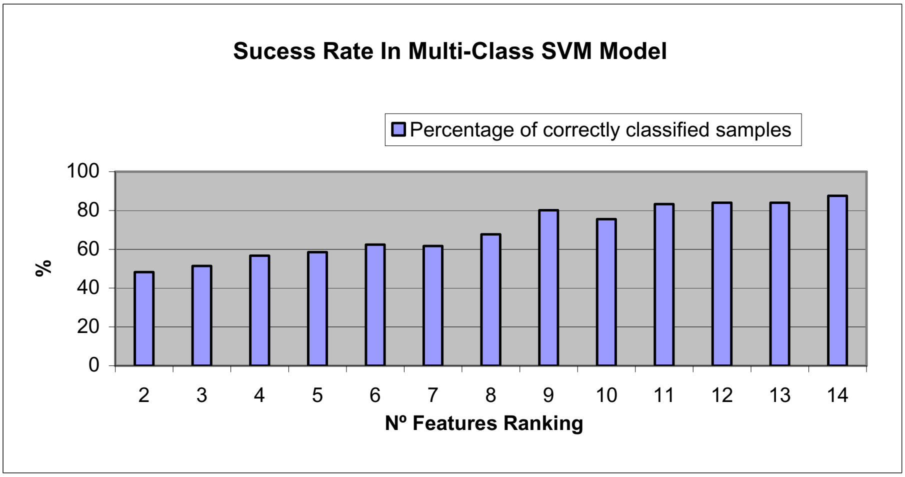


Fig. 5. (*continued*)

SVM models created for binary classification have a higher number of samples correctly classified even when using only the best ranked feature produced in the selection process. In general, as the number of features involved in creating the model increases, the model presents a greater capacity for learning. The peak is reached with the full set of input variables, resulting in a success rate close to 99%. But it should be noted that in these cases, the parameters that make up the model can be overfitting the training samples and the model might not be able to generalize the acquired knowledge to classify new samples.

In the case of SVM models developed for multiple-class classifications, the number of samples correctly classified is lower because of the much higher complexity of the problem. Even so, it can be concluded that the learning ability of the model increases with the number of features involved in its creation. It obtained promising results using only the first nine features of the ranking.

6 Conclusions

In this paper the application of feature selection methods and predictive learning algorithms is proposed to find out which factors are most influential in the differentiation of stem cells and reveal some information about the behaviour of the cell system.

The procedure for selecting those most important input variables is based on the combination of eight feature selection algorithms. The results obtained by applying this procedure provide valuable information about the stem cell differentiation process, because they determine the influence in the process of different variables such as source of extraction, growth factors and the type of medium.

The predictive learning algorithms applied are good tools to predict the behaviour of stem cell differentiation, depending on the molecules and chemicals that act as exogenous inducing factors. A decision tree provides predictive results of acceptable quality in medical research about cell stem differentiation, because the success rate reached around 80%. In addition, it generates a graphical representation of the prediction by allowing medical research groups obtain information and conclusions about the problem in question. SVM models are capable of modelling with better fidelity the behaviour of the stem cell differentiation process it; however they don't provide any representation to enable medical research groups to obtain information about the predictive process. The results suggest that using only the first features of the ranking obtained after selection, the success rate is close to 90%; this is a great contribution for a study field in the area of regenerative medicine such as stem cell differentiation where there is at present very few information available.

The methods and algorithms presented in this paper help to reduce the variety and variability in clinical and experimental results obtained. Besides, the use of such models in the medical field can improve the economic and personal resources of the research groups working on specific stem cell differentiation. The final and more important benefit will be to allow the development of specialized therapies which could improve the quality of life, survival and even rebuild the health of a large number of patients suffering from diseases that could be treated with regenerative cell therapy.

References

1. Baksh, D., Song, L., Tuan, R.S.: Adult mesenchymal stem cells: characterization, differentiation, and application in cell and gene therapy. J. Cell. Mol. Med. 8, 301–316 (2004)
2. Khademhosseini, A., Zandstra, P.W.: Engineering the in vitro cellular microenvironment for the control and manipulation of adult stem cell responses. In: Turksen, K. (ed.) Adult Stem Cells, pp. 289–314. The Humana Press Inc., Totowa (2004)
3. Kuo, C.K., Tuan, R.S.: Tissue engineering with mesenchymal stem cells. IEEE Eng. Med. Biol. Mag. 22, 51–56 (2003)
4. Johnstone, B., Hering, T.M., Caplan, A.I., Goldberg, V.M., Yoo, J.U.: In vitro chondrogenesis of bone marrow-derived mesenchymal progenitor cells. Exp. Cell Res. 238, 265–272 (1998)
5. Bosnakovski, D., Mizuno, M., Kim, G., Ishiguro, T., Okumura, M., Iwanaga, T., Kadosawa, T., Fujinaga, T.: Chondrogenic differentiation of bovine bone marrow mesenchymal stem cells in pellet cultural system. Exp. Hematol. 32, 502–509 (2004)
6. Tsuchiya, K., Chen, G., Ushida, T., Matsuno, T., Tateishi, T.: The effect of coculture of chondrocytes with mesenchymal stem cells on their cartilaginous phenotype in vitro. Mater. Sci. Eng. C 24, 391–396 (2004)
7. Mauch, R.L., Yuan, X., Tsuan, R.S.: Chondrogenic differentiation and functional maturation of bovine mesenchymal stem cells in long-term agarose culture. Osteoarthritis Cartilage 14, 179–189 (2006)
8. Till, J.E., McCulloch, E.A., Siminovitch, L.: A stochastic model of stem cell proliferation, based on growth of spleen colony-forming cells. Proc. Natl. Acad. Sci. U.S.A. 51, 29–36 (1964)
9. Nielsen, L.K., Papoutsa kis, E.T., Miller, W.M.: Modeling ex-vivo hematopoiesis using chemical engineering metaphors. Chem. Eng. Sci. 53, 1913–1925 (1998)
10. Bailon-Plaza, A., van der Meulen, M.C.H.: A mathematical framework to study the effects of growth factor influences on fracture healing. J. Theor. Biol. 212, 191–209 (2001)

11. Hentschel, H.G.E., Glimm, T., Glazier, J.A., Newman, S.A.: Dynamical mechanisms for skeletal pattern formation in the vertebrate limb. Proc. R. Soc. Lond. B 271, 1713–1722 (2004)
12. Boser, B., Guyon, I., Vapnik, V.: A Training Algorithm for Optimal Margin Classiers. In: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, COLT, pp. 144–152 (1992)
13. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
14. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning (2000)
15. Hall, M.A., Smith, L.A.: Practical feature subset selection for machine learning. In: Proceedings of the 21st Australian Computer Science Conference, pp. 181–191 (1998)
16. Dash, M., Liu, H., Motoda, H.: Consistency Based Feature Selection. Knowledge Discovery and Data Mining. Current Issues and New Applications, 98–109 (2007)
17. Buddhinath, G., Derry, D.: A Simple Enhancement to One Rule Classification, Department of Computer Science & Software Engineering. University of Melbourne, Australia
18. Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin based feature selection – Theory and algorithm. School of Computer Science and Engineering, Interdisciplinary Center for Neural Computation. The Hebrew University, Jerusalem
19. Safavian, S.R., Landgrebe, D.: A survey of decision tree classifier methodology. IEEE Trans. Syst., Man, Cybern. 21, 660–674 (1991)
20. Quinlan, J.R.: Induction on decision trees. Machine Learning 1, 81–106 (1986)
21. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth and Brooks/Cole, Monterey (1984)
22. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
23. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons, Inc., New York (1998)
24. Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D.: Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16(10), 906–914 (2000)
25. Kumar, M., Verma, R., Raghava, G.P.S.: Prediction of Mitochondrial Proteins Using Support Vector Machine and Hidden Markov Model. J. Biol. Chem. 281(9), 5357–5363 (2006)
26. Fagerlund, S.: Bird species recognition using support vector machines. Hindawi Publishing Corp., New York (2007)
27. Li, Z.-C., Zhou, X.-B., Lin, Y.-R., Zou, X.-Y.: Prediction of protein structure class by coupling improved genetic algorithm and support vector machine. Springer, Heidelberg (2008)

An Environment for Data Analysis in Biomedical Domain: Information Extraction for Decision Support Systems

Pablo F. Matos[1], Leonardo O. Lombardi[2], Thiago A.S. Pardo[3], Cristina D.A. Ciferri[3], Marina T.P. Vieira[2], and Ricardo R. Ciferri[1]

[1] Department of Computer Science, Federal University of São Carlos - São Carlos/SP, Brazil
[2] Faculty of Mathematical and Nature Sciences,
Methodist University of Piracicaba - Piracicaba/SP, Brazil
[3] Department of Computer Science, University of São Paulo - São Carlos/SP, Brazil
pablo_matos@dc.ufscar.br, lolombardi@unimep.br,
{taspardo,cdac}@icmc.usp.br, mtvieira@unimep.br,
ricardo@dc.ufscar.br

Abstract. This paper addresses the problem of extracting and processing relevant information from unstructured electronic documents of the biomedical domain. The documents are full scientific papers. This problem imposes several challenges, such as identifying text passages that contain relevant information, collecting the relevant information pieces, populating a database and a data warehouse, and mining these data. For this purpose, this paper proposes the IEDSS-Bio, an environment for Information Extraction and Decision Support System in Biomedical domain. In a case study, experiments with machine learning for identifying relevant text passages (disease and treatment effects, and patients number information on Sickle Cell Anemia papers) showed that the best results (95.9% *accuracy*) were obtained with a statistical method and the use of preprocessing techniques to resample the examples and to eliminate noise.

Keywords: Classification, information extraction, knowledge discovery, biomedical domain.

1 Introduction

In the biomedical domain there are a lot of electronic documents that report experiments involving patients who have some kind of disease, describing the treatment adopted, the number of patients enrolled in the treatment, which symptoms and risk factors are associated with the disease, and if the treatment has interfered positively or negatively in the patient's health. The experiments are reported in several magazines and journals, e.g., *American Journal of Hematology*, *Blood*, and *Haematologica.* Researchers and doctors are not able to process this huge number of documents to extract key information related to some issues of interest.

These documents are in unstructured format, that is, in plain textual form. It is necessary to transform this information from unstructured to structured format in order to

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 306–316, 2010.

submit it to an automatic knowledge discovery process. For this purpose, an environment/framework called **I**nformation **E**xtraction and **D**ecision **S**upport **S**ystem in **Bio**medical domain (IEDSS-Bio) is proposed in this paper. This environment is under development and aims at supporting the expert in making decisions, by extracting relevant information from biomedical documents, storing the information in a data warehouse, and mining interesting knowledge from it.

After presenting the general environment architecture and its data flow, this paper focuses mainly on the adopted information extraction process, more specifically, on the task of identifying text passages/sentences that contain the relevant information. A case study on identifying sentences about disease and treatment effects and patients number from Sickle Cell Anemia papers is reported, showing that a high *accuracy* (95.9%) is achieved with a statistical machine learning method and the use of preprocessing techniques to resample the examples and to eliminate noise.

This paper is organized as follows. Section 2 presents the theoretical foundation about text and data mining, and information extraction. Section 3 reviews the main related work. Section 4 presents the IEDSS-Bio environment and some examples of the mined knowledge. Section 5 describes the information extraction module. Section 6 discusses the experiments and Section 7 concludes the paper.

2 Theoretical Foundation

Text mining [1] refers to the process of extracting useful information from documents in unstructured format by identifying knowledge and exploiting patterns. Texts are converted into structured format to use data mining techniques.

Data Mining is the application of specific algorithms for extracting patterns from data [2]. One of the most popular data mining tasks is the discovery of association rules, which aims at finding items that frequently occur together in the data. An association rule is an implication of the form $A => B$, where A and B are item sets [3]. The implication means that databases tuples (transactions) satisfying A are likely to satisfy B. *Support* and *confidence* measures are used to indicate relevant rules [3].

Traditionally, data mining algorithms are applied to data that are gathered in a single table. The data gathering process from multiple tables through joints or aggregation may cause loss of meaning or information and may have a high computational cost [4]. Multi-relational data mining methods search for patterns that involve multiple tables (relations) from a relational database, maintaining the data separately in the data mining process. In the biomedical database it is adequate the use of multi-relational data mining due to the properties of the data.

Regarding information extraction, Cohen and Hunter [5] present two approaches: the rule-based approach and the machine learning approach. The first one uses some kind of knowledge; the second one uses classifiers to separate sentences or documents. Krauthammer and Nenadic [6] and Ananiadou and McNaught [7] present a third approach: the dictionary-based approach, which uses information from a dictionary to assist in the identification of terms or entities in the text. These approaches are the three predominant ones for knowledge extraction in the biomedical domain.

The dictionary-based approach has the advantage of storing information related to a particular area and makes possible the identification of terms such as names of gene and protein. Some problems of this approach are the limited number of names in the dictionary, the change of names, which generates a low *recall*, and the short names, which generate false positives and decrease the *precision* [8].

The rule-based approach has some disadvantages: to delay the construction of systems significantly, to reduce the adaptability of rules to another system, and to remove terms that do not match the predefined patterns. However, generally it has performed better than other approaches [7].

The advantages of using the machine learning approach are the domain-independence and high quality prediction. The main problems related to the machine learning algorithms are the need for large amount of representative training data [7] and for (possible) retraining with the advent of new data.

3 Related Work

In the biomedical literature, there are studies that extract information from abstracts or full papers about gene or protein mainly, using a combination of the three approaches previously discussed. Most of these studies extract information from MEDLINE abstracts [9, 10, 11, 12]. The works that extract information from full text papers have different goals, namely: to extract information [13], to populate a database [14], or to highlight the sentences in accordance with a user query [15]. Some of the above works use *Part-Of-Speech tagging* [9, 10, 13, 16], while some do not [11, 12, 14, 15].

Among the studies, [16] produced the highest *precision* (72.5%) and *recall* (50.7%), using a combination of machine learning, dictionary-based and rule-based approaches. Using the same approaches to extract information from abstracts, higher values were obtained, respectively, 85.7% and 66.7% [13]. This difference shows the particularity in extracting information from full papers. According to Cohen and Hersh [17], AbGene [13] is the most successful rule-based approach for the recognition of gene and protein in biomedical texts.

4 An Environment for Data Analysis

The IEDSS-Bio environment is shown in Fig. 1. It is divided into two major components as shown in the dotted rectangles:

- *Conversion and Extraction component*: aims at converting different formats and extracting relevant information from scientific documents to store it in a biomedical database (**1**);
- *Data Analysis component*: aims at identifying patterns from the biomedical domain by applying data warehouse and data mining techniques (**2**).

From the papers available in PDF format, the *Convert module* handles PDF files and converts them into the semi-structured XML format. The information contained in the XML document is organized in hierarchical levels such as *section* » *page* » *paragraph* » *sentence* to make the extraction processing easier. Next, the *Information Extraction module* processes the information from the XML document to extract the most relevant information from the sentences. Fig. 1 shows that the expert may interfere in the *Converter module*, verifying the quality of the generated document and suggesting changes. In Section 5 the *Information Extraction* (IE) *module* is explained in details.

After the extraction of relevant information from the XML document, the extracted information is stored in the biomedical database by the *Persistence Layer module*. The *Data Extraction and Integration* (DEI) *module* is responsible for extracting data from the biomedical database through the *Persistence Layer* and for populating the *Data Warehouse* (DW).

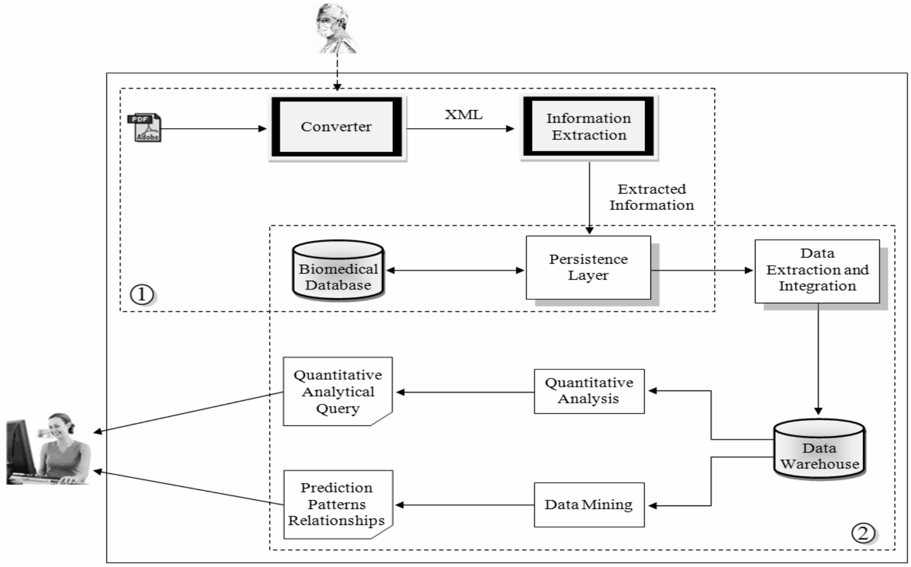


Fig. 1. Environment for data analysis

The data stored in the DW may be used in two kinds of processes in order to support data analysis. In the first one, the DW may be consulted for obtaining quantitative analysis. For instance, it is possible to answer the question "*How many patients had clinical improvement and were treated with the hydroxyurea drug?*" In the second one, the DW may be used by data mining algorithms aiming at identifying interesting patterns in the data. For instance, it is possible to find out that "*A significant amount of patients under treatment with the hydroxyurea drug tend to have marrow depression*".

The biomedical database of Fig. 1 has been populated with treatments, positive and negative effects, and number of patients extracted from experiments reported in the papers. Fig. 2 shows parts of some of the tables that compose the database. The joint analysis of this information may lead to the discovery of useful information for decision-making. However, the following features in the data need to be adequately addressed, requiring a new approach for the data mining algorithm to be used:

1. The number of patients involved is not explicitly reported in many circumstances (which had undergone certain treatments, which had certain effects or certain symptoms), requiring some inference on the data; only after this process, the data may be stored in the database;

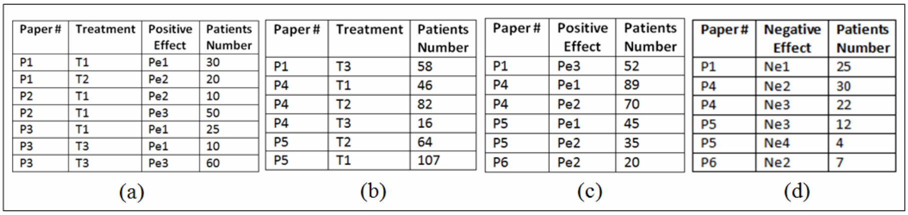
2. The information about treatments and (negative or positive) effects was stored in two ways in the database:
   - in a table that lists treatment and effect caused by it, when the paper states clearly that information (table (a) in Fig. 2);
   - in separate tables, if the association between cause and effect is not mentioned in the paper (tables (b), (c) and (d) in Fig. 2).

Even though this information is located in different tables and do not maintain explicit links between each other, frequent co-occurrences in the data may indicate that relationships exist between them. An example would be a significant number of experiments reporting the use of a specific treatment and the occurrence of certain effect. Therefore, it is important to analyze such information. The multi-relational data mining technique is suitable for this purpose.

Fig. 2 illustrates a subset of stored data. Considering $hydroxyurea = T2$, $lower\_annual\_rates\_of\_crises = Pe2$ and $marrow\_depression = Ne3$, the following rule could be found by jointly analyzing the tables (b), (c), and (d):

> *Rule 1: hydroxyurea => lower_annual_rates_of_crises, marrow_depression [support = 10%; confidence = 80%]*

where values of *support* and *confidence* are just for illustration and do not refer to an actual generated rule.



Paper#	Treatment	Positive Effect	Patients Number
P1	T1	Pe1	30
P1	T2	Pe2	20
P2	T1	Pe2	10
P2	T1	Pe3	50
P3	T1	Pe1	25
P3	T3	Pe1	10
P3	T3	Pe3	60

(a)

Paper#	Treatment	Patients Number
P1	T3	58
P4	T1	46
P4	T2	82
P4	T3	16
P5	T2	64
P5	T1	107

(b)

Paper#	Positive Effect	Patients Number
P1	Pe3	52
P4	Pe1	89
P4	Pe2	70
P5	Pe1	45
P5	Pe2	35
P6	Pe2	20

(c)

Paper#	Negative Effect	Patients Number
P1	Ne1	25
P4	Ne2	30
P4	Ne3	22
P5	Ne3	12
P5	Ne4	4
P6	Ne2	7

(d)

Fig. 2. Treatment_Positive_Effect (a), Treatment (b), Positive_Effect (c), and Negative_Effect (d) tables

5 Information Extraction Module

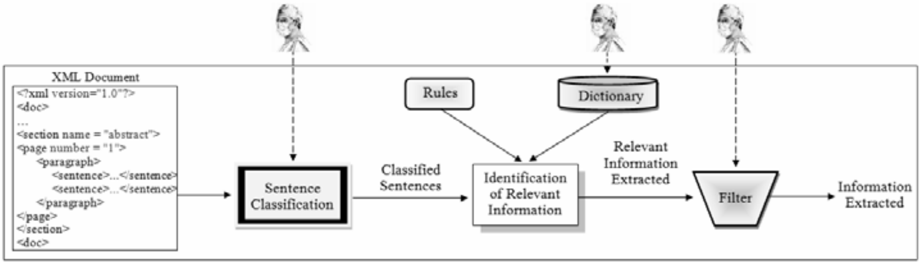
The information extraction of biomedical domain uses a combination of three approaches:

1. Machine learning: used for sentence classification;
2. Rule-based: formal method to specify text patterns, used to extract information from sentences classified in the previous step;
3. Dictionary-based: a manually built dictionary (containing terms from the scientific papers) stores information related to the biomedical domain in order to assist the construction of rules, used to increase the *precision* and *recall* in the identification of such information.

Fig. 3 shows the adopted process for extracting information from the biomedical domain. The sentences from each paper processed by the *IE module* are classified using machine learning techniques. Each classified sentence is analyzed through regular expressions (rules) to identify relevant information. The dictionary of terms from the biomedical domain is used to assist the construction of the regular expressions. An expert may validate the information extracted (filter) before it is stored in the database. The expert may also evaluate the classified sentences and add new terms to the dictionary.

Formally, consider $S = \{s_1, s_2, \dots, s_n\}$ the set of sentences of training and $C = \{c_1, c_2, \dots, c_n\}$ the set of predefined classes. The classification aims at identifying a function f that maps each sentence s_i to one of the predefined classes c_j: $f\ (s_i) = c_j$, where $i = 1, \dots, n$ and $j = 1, \dots, m$.

For each class c_j a set of regular expressions re_{θ} is defined in order to obtain relevant information. The symbol θ denotes one of the following concepts: the number of patients involved in the experiment under different circumstances (total patients, patients who have had positive or negative effects, patients that have completed or not the treatment), the treatment adopted, the positive or negative effects and the symptoms of the patient.



```
<?xml version="1.0"?>
<doc>
...
<section name = "abstract">
<page number = "1">
    <paragraph>
        <sentence>...</sentence>
        <sentence>...</sentence>
    </paragraph>
</page>
</section>
<doc>
```

Fig. 3. Information extraction module

The sentence classification model is built from the documents (papers) suggested by experts. The model consists of structuring the sentences in an attribute-value matrix whose rows represent the sentences and the columns the attributes (n-grams). Each sentence is associated to a class. Each cell in the matrix indicates the presence/frequency or absence of the corresponding attribute in a sentence. The data may undergo stopwords removal, stemming, and attribute selection processes.

The *IE module* identifies the relevant information of the classified sentences through regular expressions and the dictionary. The following example shows one possible regular expression:

$\text{re}_{\text{negative\_effect\_patients}}$ = (⟨negative effect⟩)(.*)([0–9]+)percent(.*)(⟨individual⟩)

The ⟨negative effect⟩ and ⟨individual⟩ are terms that are stored in the dictionary. The following sentences are examples represented by this regular expression:

"Respiratory failure was documented in 13 percent of patients."

"Neurologic events occurred in 11 percent of patients."

"Neurologic complications developed in 22 percent of the adults in our study."

These sentences express the percentage of patients who suffered from some negative effects. Although the information is different, the sentence structures are similar.

Regular expressions are used to cover different ways of expressing the same information. Some challenges were faced, although regular expressions provide resources that allow dealing with different writing styles:

1. In some cases, it is necessary to interpret the sentence to understand it, which is beyond the scope of regular expressions. One example is shown in the following sentence: *"Fewer patients assigned to hydroxyurea had chest syndrome (25 vs. 51, P < 0.001), and fewer underwent transfusions (48 vs. 73, P = 0.001)."*

   The author compares the results of two treatments to which the patients were submitted. He cites the number of patients who had chest syndrome and those who underwent transfusions for each treatment. Given the way he reports these numbers, the information requires interpretation of other information mentioned in the paper to identify the treatment that led to each result.
2. In other cases, additional treatment is necessary to obtain the desired information. For instance, in order to estimate the number of patients who suffered from respiratory failure in the following sentence (13%) it is necessary to identify all patients who participated in the experiment, which was mentioned somewhere else in the text: *"Respiratory failure was documented in 13 percent of patients."*

6 Evaluation: A Case Study

Experiments were carried out in the biomedical domain, more specifically on papers about Sickle Cell Anemia [18], which is a genetic and hereditary disease considered as a public health problem. These experiments are specifically on sentence classification from medical scientific papers, dealing with positive and negative effects and the number of patients enrolled in treatments. The goal is to validate the classification task, which is an important step in the information extraction process.

Three questions are intended to be answered here: (1) How do human beings manually perform the sentence classification? (2) Is it feasible to automate the sentence classification task? (3) What kind of classification algorithm performs better in this classification?

To answer the *first* question, it is necessary to know the results obtained by humans in the sentence classification. For this end, the Kappa measure [19] was used to compute the agreement among humans. The achieved results are shown in Section 6.1. After that, it is possible to answer the *second* question by comparing the obtained Kappa value with a nominal scale agreement. To know the answer for the *third* question, six classical machine learning algorithms were chosen. These algorithms are from different paradigms: Support Vector Machine (SVM) and Naïve Bayes (NB) are statistical methods; ID3, J48, Prism, and OneR are symbolic methods – the first two are algorithms for decision tree induction and the last two are algorithms for rules induction. All classification experiments were performed with Weka data mining environment [20]. Default values for the learning parameters were used as they appear in Weka. The Mover classification system [21] is considered as a baseline method. It is a classical tool for text structure classification and is a NB approach to the problem. The results are shown in Section 6.2.

6.1 Annotation Agreement

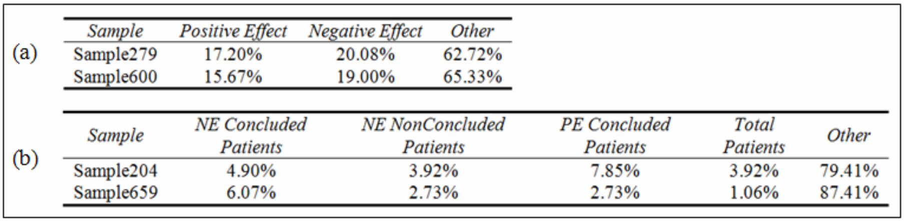
The Kappa measure for the evaluation of annotation agreement among humans was calculated in order to know the difficulty of the sentence classification. The idea is that the more humans agree the better the task is defined and, therefore, the more it may be automated. Humans were divided into two groups: naïve subjects (i.e., that are not experts on the topic) and experts. Table 1 shows the results for a sample of 50 sentences. One may see that there were 3 naïve subjects and 3 experts. In the last row of the table, the annotation agreement was collected for all humans, experts or not. We computed the agreement for positive and negative effect classes, for other possible classes (under the class named as "other"), and for the 3 classes together. An interesting result is that the naïve subjects group has gotten the overall agreement value higher (0.71) than the experts (0.63). According to the Landis and Koch scale [22], the obtained Kappa for all the classes is in the range of substantial agreement (0.61 to 0.81). We may conclude that the classification task is well defined. Therefore, this result answers the *first* and *second* questions mentioned previously, i.e., for this task it is possible to automatically perform the classification.

Table 1. Annotator agreement on 50 sentences

Annotator	*Positive Effect*	*Negative Effect*	*Other*	*All the classes*
3 experts	0.77	0.63	0.52	**0.63**
3 naïve subjects	0.80	0.72	0.60	**0.71**
experts + naïve subjects	0.75	0.66	0.55	**0.65**

6.2 Experiments with Sentence Classification

In this section, the experiments on sentence classification are reported, focusing on the information of interest: effect and patients. Two samples were selected for each one: Sample279 and Sample600 for effect sentences; and Sample204 and Sample659 for patients sentences. The number that identifies each sample also indicates the number of examples/sentences that the corresponding sample contains. The effect sentences were divided into three classes: *Positive Effect*, *Negative Effect* and *Other*. The patients sentences were separated in five classes: *Negative Effect Concluded Patients*, *Negative Effect NonConcluded Patients*, *Positive Effect Concluded Patients*, *Total Patients* and *Other*. *Concluded* and *NonConcluded* refer to the patients that concluded and did not conclude the experiments, respectively. The percentage distribution of the classes for the information of interest may be seen in Fig. 4.



(a)

Sample	*Positive Effect*	*Negative Effect*	*Other*
Sample279	17.20%	20.08%	62.72%
Sample600	15.67%	19.00%	65.33%

(b)

Sample	*NE Concluded Patients*	*NE NonConcluded Patients*	*PE Concluded Patients*	*Total Patients*	*Other*
Sample204	4.90%	3.92%	7.85%	3.92%	79.41%
Sample659	6.07%	2.73%	2.73%	1.06%	87.41%

Fig. 4. Effect (a) and Patient (b) distribution of classes for each sample

The experiments were conducted as follows. First of all, the sentences (for every class) were cleaned by removing commas, parenthesis, etc. Then the attribute-value matrix was constructed using minimum frequency two for selecting the attribute, i.e., attributes whose values happened only once were not considered. The attributes were 1 to 3-grams, and the values were: 1, for the case the n-gram occurs in the sentence, or 0 otherwise. Neither stopwords removal nor stemming were considered.

After that, the six previous machine learning algorithms were used in the experiments. Six kinds of preprocessing were used for each algorithm, generating 36 possible combinations. The preprocessing possibilities were: *No Filter* (NF), *Randomize* (RD)*, Remove Misclassification* (RM), *Resample* (RS), RM followed by RS, and RS followed by RM. NF does not use any kind of filter; RD is used to random the examples; RM is used to eliminate noise; RS is an oversampling method used to balance the examples. The partitioning method 10-fold cross-validation was used to generate the results. It trains the classification model on 9 folds and tests it on the remaining one. The *accuracy* measure (number of correctly classified sentences/number of sentences) was computed.

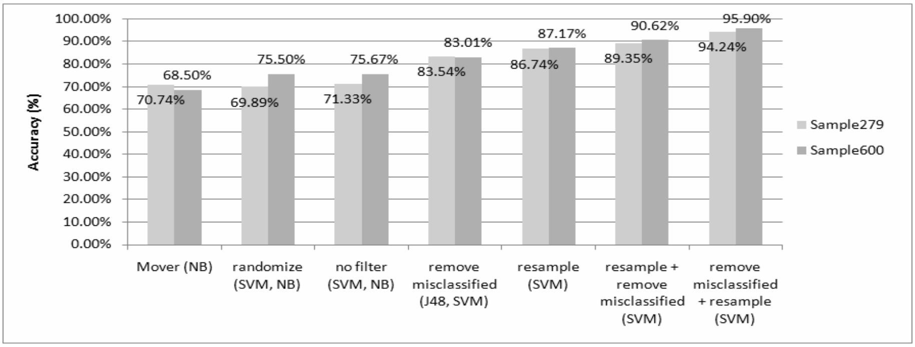


Fig. 5. Better results (*accuracy* measure) obtained in effect class

The results reported in Fig. 5 show that in general the SVM algorithm performed better than the other algorithms. The best results were produced with the use of RM and RS filters, i.e., by removing the mislabeled sentences and resampling the examples in order to establish a uniform rate between the classes. The results obtained from patients class were similar to the effect class and are not shown in this paper. The better results obtained by SVM answer the *third* question in this paper.

Since regular expressions select the sentence parts of interest after sentence classification, it is best suited that the classifier recovers a big number of sentences, even if they are misclassified (high *recall*, but probably low *precision*). A classifier that selects correctly most of the sentences (high *precision*) may recover fewer sentences, and this may injure the information extraction process.

7 Conclusion and Future Work

The environment proposed in this paper – Information Extraction and Decision Support System in Biomedical domain – aims at being a general framework for mining relevant information in the area. Our first experiments on sentence classification (which is a step of the whole process) showed very good results (95.9% *accuracy*) for papers about Sickle Cell Anemia. Therefore, the task of sentence classification in this area is well defined and possible to be automated.

As future work, we plan to investigate the identification of treatment and symptoms information in the text, as well as proceed to the extraction of the relevant sentence pieces for populating our databases (using some of the techniques discussed in this paper). We also envision to investigate the use of parallel processing to optimize the more time-consuming tasks, e.g., the application of data mining algorithms and the analytical query processing. Other biomedical areas may also benefit from our text mining approach. This also remains for future work.

Acknowledgments. The authors acknowledge the support of the following Brazilian research agencies: CAPES, FAPESP, CNPq, and FINEP.

References

1. Feldman, R., Sanger, J.: The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, New York (2007)
2. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From Data Mining to Knowledge Discovery in Databases. AI Magazine 17(3), 37–54 (1996)
3. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
4. Džeroski, S.: Multi-Relational Data Mining: An Introduction. ACM SIGKDD Explorations Newsletter 5(1), 1–16 (2003)
5. Cohen, K.B., Hunter, L.: Getting Started in Text Mining. PLoS Computational Biology 4(1), 1–3 (2008)
6. Krauthammer, M., Nenadic, G.: Term Identification in the Biomedical Literature. Journal of Biomedical Informatics 37(6), 512–526 (2004)
7. Ananiadou, S., McNaught, J. (eds.): Text Mining for Biology and Biomedicine. Artech House, Norwood (2006)
8. Tsuruoka, Y., Tsujii, J.: Improving the Performance of Dictionary-Based Approaches in Protein Name Recognition. Journal of Biomedical Informatics 37(6), 461–470 (2004)
9. Chun, H.-W., Tsuruoka, Y., Kim, J.-D., Shiba, R., Nagata, N., Hishiki, T., Tsujii, J.: Extraction of Gene-Disease Relations from Medline Using Domain Dictionaries and Machine Learning. In: 11th PSB, Hawaii, pp. 4–15 (2006)
10. Mika, S., Rost, B.: NLProt: Extracting Protein Names and Sequences from Papers. Nucleic Acids Research 32(Suppl. 2), 634–637 (2004)
11. Seki, K., Mostafa, J.: A Hybrid Approach to Protein Name Identification in Biomedical Texts. Information Processing & Management 41(4), 723–743 (2005)
12. Hanisch, D., Fundel, K., Mevissen, H., Zimmer, R., Fluck, J.: Prominer: Rule-Based Protein and Gene Entity Recognition. BMC Bioinf. 6(Suppl. 1), S14 (2005)
13. Tanabe, L., Wilbur, W.J.: Tagging Gene and Protein Names in Biomedical Text. Bioinformatics 18(8), 1124–1132 (2002)
14. Bremer, E.G., Natarajan, J., Zhang, Y., DeSesa, C., Hack, C.J., Dubitzky, W.: Text Mining of Full Text Articles and Creation of a Knowledge Base for Analysis of Microarray Data. In: López, J.A., Benfenati, E., Dubitzky, W. (eds.) KELSI 2004. LNCS (LNAI), vol. 3303, pp. 84–95. Springer, Heidelberg (2004)

15. Garten, Y., Altman, R.: Pharmspresso: A Text Mining Tool for Extraction of Pharmacogenomic Concepts and Relationships from Full Text. BMC Bioinf. 10(Suppl. 2), S6 (2009)
16. Tanabe, L., Wilbur, W.J.: Tagging Gene and Protein Names in Full Text Articles. In: Workshop on NLP in the Biomedical Domain, pp. 9–13. ACL, Phildadelphia (2002)
17. Cohen, A.M., Hersh, W.R.: A Survey of Current Work in Biomedical Text Mining. Briefings in Bioinformatics 6(1), 57–71 (2005)
18. Pinto, A.C.S., Matos, P.F., Perlin, C.B., Andrade, C.G., Carosia, A.E.O., Lombardi, L.O., Ciferri, R.R., Pardo, T.A.S., Ciferri, C.D.A., Vieira, M.T.P.: Technical Report Sickle Cell Anemia. Technical Report, Federal University of São Carlos (2009), http://sca.dc.ufscar.br/download/files/report.sca.pdf
19. Fleiss, J.L.: Measuring Nominal Scale Agreement among Many Raters. Psychological Bulletin 76(5), 378–382 (1971)
20. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 2nd edn. Morgan Kaufmann, San Francisco (2005)
21. Anthony, L., Lashkia, G.V.: Mover: A Machine Learning Tool to Assist in the Reading and Writing of Technical Papers. IEEE Trans. Prof. Comm. 46(3), 185–193 (2003)
22. Landis, J.R., Koch, G.G.: The Measurement of Observer Agreement for Categorical Data. Biometrics 33(1), 159–174 (1977)

Constructive Neural Networks to Predict Breast Cancer Outcome by Using Gene Expression Profiles

Daniel Urda, José Luis Subirats, Leo Franco, and José Manuel Jerez

Departament of computer science,
University of Málaga. Spain
{durda,jlsubirats,lfranco,jja}@lcc.uma.es
http://www.lcc.uma.es

Abstract. Gene expression profiling strategies have attracted considerable interest from biologist due to the potential for high throughput analysis of hundreds of thousands of gene transcripts. Methods using artifical neural networks (ANNs) were developed to identify an optimal subset of predictive gene transcripts from highly dimensional microarray data. The problematic of using a stepwise forward selection ANN method is that it needs many different parameters depending on the complexity of the problem and choosing the proper neural network architecture for a given classification problem is not a trivial problem. A novel constructive neural networks algorithm (CMantec) is applied in order to predict estrogen receptor status by using data from microarrays experiments. The obtained results show that CMantec model clearly outperforms the ANN model both in process execution time as in the final prognosis accuracy. Therefore, CMantec appears as a powerful tool to identify gene signatures that predict the ER status for a given patient.

Keywords: Artificial neural networks; Constructive neural networks; Predictive modelling; Gene expression profiles; Breast cancer.

1 Introduction

Gene expression profiling strategies have been widely used in cancer studies for prediction of disease outcome [24,22] by using data from microarrays experiments. Microarrays technology is a powerful platform successfully used for the analysis of gene expression in a wide variety of experimental studies [13]. Nevertheless, this platform provides data that are in origin vulnerable to the curse of dimensionality (each sample is characterized by several thousand gene transcripts) and the curse of data set sparsity (only few examples are available for the analysis). In this sense, the estimation of prognosis models by using gene expression profiling requires the application of an additional procedure to select the most significant gene transcripts.

Methods using artifical neural networks (ANNs) were developed to identify an optimal subset of predictive gene transcripts from highly dimensional microarray data [22,14]. Recently, Lancashire et al. [7] proposed a stepwise forward

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 317–326, 2010.

selection artificial neural network approach, in which architecture size evolves sequentially by selecting and adding input neurons to the network, in order to identify an optimum gene subset based on predictive performance. This method was applied to a gene microarray dataset to identify and validate gene signatures corresponding with estrogen receptor in breast cancer.

Nevertheless, in spite of the demonstrated efficiency of ANNs in prediction and classification problems in general, the estimation of prognosis models by using stepwise forward selection procedures has, on one hand, a high computational cost, and on the other hand, a high complexity related to the design of the optimal architecture size for the neural network models. Several approaches [16,2,8,5] were proposed in this sense to solve, or alleviate, the problem of choosing the proper neural network architecture for a given problem. However, there is no general agreement on the strategy to follow in order to select an optimal neural network architecture, and the computationally inefficient "trial and error" method is still much used in applications using ANNs.

Constructive neural networks (CNN) algorithms constitute an alternative in designing artificial neural networks models [9,3,6,1,11,21,10,12,19]. This family of algorithms generate the network topology online by adding neurons during the training phase. This work presents the application of a novel constructive algorithm (CMantec) that incorporates competition and cooperation among neurons to obtain smaller in size architectures and better generalization capabilities, and analyzes its application to the problem of predicting estrogen receptor status by using data from microarrays experiments.

2 Materials and Methods

2.1 Materials

The dataset that has been used in this work was downloaded from[1] and consists of 49 samples each with 7129 corresponding variables specifying the intensity of the probe sets targeting each transcript. These data were previously published by West et al [23]. This used microarray technology to analyse primary breast tumours in relation to estrogen receptor (ER) status. The dataset consists of 25 patients with ER+ and 24 with ER-.

2.2 Methods

For the ANN analysis, the code was written using the neural network toolbox of MATLAB and was launched on a computational cluster with 30 nodes. The CMantec algorithm was developed in Visual Studio 2008 and was launched on a personal computer. In order to compare CPU time of these algorithms, both of them were launched on the same single computer. In section 3 the reader could find time results for these algorithms.

[1] http://data.cgt.duke.edu/west.php

Model estimation procedure. The learning process of the stepwide method proposed by [7] and shown in Fig. 1 begins considering 7129 models of one input. On each model, the dataset is divided into three randomly subsets; 60% for training, 20% for validating and 20% for testing. CMantec does not need to validate the model so this 20% percent is added to the training subset and finally uses 80% for training and 20% for testing. Once we have the model of one input that best generalize our problem, the process now begins to consider models of two inputs where the first input is the best obtained on the last step. This process is repeated until we find a step with one more input where its generalization do not increase the generalization obtained on the previous step.

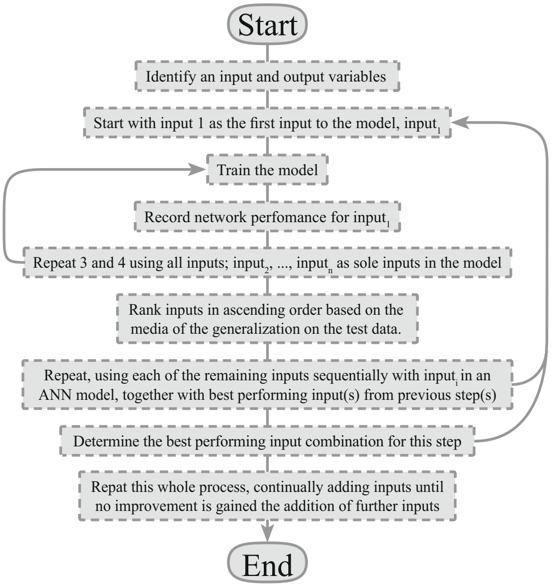

Start
Identify an input and output variables
Start with input 1 as the first input to the model, input$_1$
Train the model
Record network perfomance for input$_1$
Repeat 3 and 4 using all inputs; input$_2$, ..., input$_n$ as sole inputs in the model
Rank inputs in ascending order based on the media of the generalization on the test data.
Repeat, using each of the remaining inputs sequentially with input$_1$ in an ANN model, together with best performing input(s) from previous step(s)
Determine the best performing input combination for this step
Repat this whole process, continually adding inputs until no improvement is gained the addition of further inputs
End

Fig. 1. Summary of the stepwise algorithm modelling process

For each model, a randomly subset of the dataset is created 50 times and for each subset the model is trained 10 times. This is 7.129 x 50 x 10 = 3.564.500 simulations per step of the model.

ANN architecture. The ANN modelling used [7] is a multilayer perceptron architecture with a sigmoidal transfer function[2] where weights are updated by a back propagation algorithm incorporating supervised learning.

Data of the dataset were scaled linearly between 0 and 1 using minimun and maximum values. The architecture shown in Fig. 2 utilised two hidden perceptrons in a single hidden layer and its initial weights were randomized between 0

[2] In the tolboox of MATLAB we used 'tansig' as the transfer function.

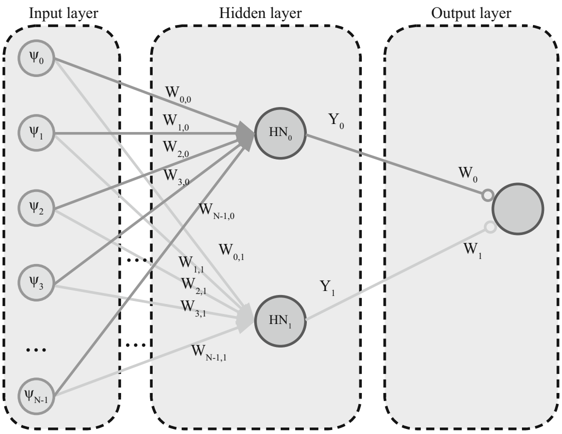


Fig. 2. Architecture of the ANN

and 1. Network training was stopped when the network error failed to improve on the test data split for 5000 epochs.

CMantec algorithm. CMantec (Competitive Majority Network Training Error Correcting) is a novel neural network constructive algorithm that combines competition between neurons using a modified perceptron learning rule to build compact architectures and cooperation between them to obtain the output of the network.

The topology of a CMantec network consists of one hidden layer that maps the information to an output neuron that uses a majority function to give us the output network (Fig. 3a). The choice of the output function as a majority gate is motivated by previous experiments in which very good computational capabilities have been observed for the majority function among the set of linearly separable functions [18]. The implementation of the majority neuron can be done establishing all the sinaptic weights to 1 and the umbral value to $M/2$ where M is the number of neurons in the hidden layer (Fig. 3b).

The results [20] show that the new algorithm generates very compact neural architectures with state-of-the-art generalization capabilities. In this paper we are applying this developed model to a real bioinformatic problem.

- Thermal Perceptron rule.
  At the single neuron level (Fig. 3c) the CMantec algorithm uses the thermal perceptron rule that ensures stability of the acquired knowledge while the architecture grows and while the neurons compete for new incoming information. Competition makes it possible that even after new units have been added to the network, existing neurons still can learn if the incoming information is similar to their stored knowledge and thus we first give some details of this algorithm. The thermal perceptron, introduced by Frean in 1992 [4] is a modification of the original perceptron learning rule [15] that

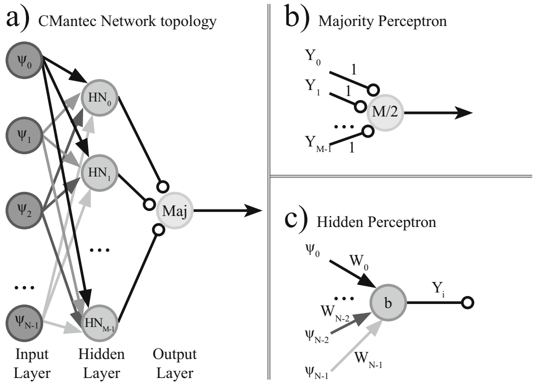


Fig. 3. Architecture of a CMantec network

aims to obtain a rule that provides a successful and stable linearly separable approximation to a non-linearly separable problem. The standard perceptron learning rule converges when the problem is linearly separable but is unstable when trying to learn non-linearly separable problems.

Consider a neuron, modelled as a threshold gate, having two response states: ON=TRUE=1 and OFF=FALSE=0, receiving input from N incoming continuous signals. The activation state (S) of the perceptron depends on the N input signals, ψ_i, and on the actual value of the N synaptic weights (w_i) and the bias (b) as follows:

$$S = \begin{cases} 1 \; (ON) & \text{if } \; \phi \geq 0 \\ 0 \; (OFF) & \text{otherwise,} \end{cases} \tag{1}$$

where ϕ is the synaptic potential of the neuron defined as:

$$\phi = \sum_{i=1}^{N} w_i \; \psi_i \; - \; b. \tag{2}$$

In the thermal perceptron rule, the modification of the synaptic weights, Δw_i, is done on-line (after the presentation of a single input pattern) according to the following equation:

$$\Delta w_i = (t - S) \;\; \psi_i \;\; T_{fac} \; , \tag{3}$$

where t is the target value of the presented input, and ψ is the value of input unit i connected to the output by weight w_i. The difference to the standard perceptron learning rule is that the thermal perceptron incorporates the factor T_{fac}. This factor, whose value is computed as shown in Eq. 4, depends on the value of the synaptic potential and on an artificially introduced temperature (T) that is decreased as the learning process advances, in a way that resembles the well known simulated annealing procedure [17].

$$T_{fac} = \frac{T}{T_0}\ \exp\{-\frac{|\phi|}{T}\}. \tag{4}$$

In Eq. 4, T_0 is the initial temperature value set at the beginning of the learning process, T is the actual temperature, and ϕ is the synaptic potential defined in Eq. 2. The value of the Temperature, T, is lowered steadily as the iterations (I) proceeds from the iteration 1 to the maximum number of iterations allowed (I_{max}) according to the value of a parameter m, following the next linear equation.

$$T\ =\ T_0 - m\, I. \tag{5}$$

In the Frean original formulation [4], the evolution of T depends on the values of T_0, m and I_{max} but in order to reduce the space of parameters and as it was observed that the algorithm was not much affected, the value of m in this work was set as to reduce T to 0 at the end of the allowed number of iterations, i.e., when $I = I_{max}$. Further, we observed that it was not necessary to use different values of T_0 when the input dimension was fixed and thus all the experiments in this work has been run with a value of T_0 equals to the number of input variables, N.

- The CMantec algorithm.
  The procedure for constructing an architecture for a given set of examples (inputs plus corresponding target values) of dimension N starts by putting N neurons in the input layer, with no processing capabilities except the task of introducing the incoming values to the network. The initial architecture includes a single neuron in the hidden layer and an output neuron that will compute the majority function of the activation of the neurons present in the hidden layer.

  The learning process starts as usual with the presentation of randomly chosen patterns from the training data set and the single present neuron in the hidden layer tries to learn these patterns using the thermal perceptron rule in Eq. 3 described in the previous section.

  The CMantec algorithm has 3 parameters to be set at the time of starting the learning procedure. The maximum number of iterations allowed for each thermal perceptron in the hidden layer per learning cycle , I_{max}. The second parameter is named the growing factor, (g_{fac}), as it determines when to stop a learning cycle and include a new neuron in the hidden layer. The third parameter is Fi_{temp} and shows how fast is going to be deleted the noisy examples from the training dataset.

  Thus, patterns are learned by selecting those thermal perceptrons from the hidden layer where its output differs from the target value and its T_{fac} is larger than the set value of g_{fac}. If more than one thermal perceptron in the hidden layer satisfy these conditions, the perceptron that has the highest

T_{fac} is the selected candidate to learn the incoming pattern. A new single neuron is added when there is no thermal perceptron that comply these conditions and a new learning cycle is created. In this case I is set to 0 for every thermal perceptron in the hidden layer and noisy examples are deleted by an active learning rule.

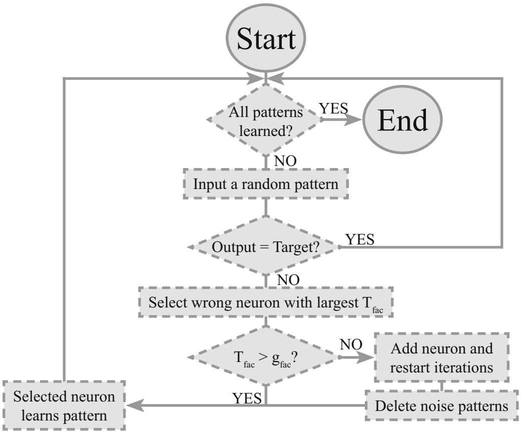

Start
All patterns learned?
YES
End
NO
Input a random pattern
Output = Target?
YES
NO
Select wrong neuron with largest T_{fac}
T_{fac} > g_{fac}?
NO
Add neuron and restart iterations
YES
Selected neuron learns pattern
Delete noise patterns

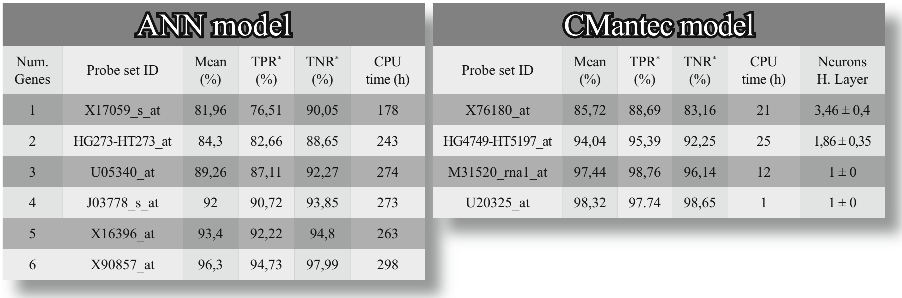
Fig. 4. Flow diagram of the CMantec algorithm

In Fig. 4 a flow diagram of the algorithm is shown. It includes a filtering stage, showed at the lower right part of the figure ("Delete noisy patterns"). This stage filters those patterns that not satisfy a filtering condition. This condition is established as:

$$nlp < mean(nap) + \phi * std(nap)$$

where nlp is the number of times that a certain pattern must be learned and nap is a vector of nlp values corresponding to the resting patterns in

Table 1. ANN and CMantec identified genes



ANN model

Num. Genes	Probe set ID	Mean (%)	TPR* (%)	TNR* (%)	CPU time (h)
1	X17059_s_at	81,96	76,51	90,05	178
2	HG273-HT273_at	84,3	82,66	88,65	243
3	U05340_at	89,26	87,11	92,27	274
4	J03778_s_at	92	90,72	93,85	273
5	X16396_at	93,4	92,22	94,8	263
6	X90857_at	96,3	94,73	97,99	298

CMantec model

Probe set ID	Mean (%)	TPR* (%)	TNR* (%)	CPU time (h)	Neurons H. Layer
X76180_at	85,72	88,69	83,16	21	3,46 ± 0,4
HG4749-HT5197_at	94,04	95,39	92,25	25	1,86 ± 0,35
M31520_rna1_at	97,44	98,76	96,14	12	1 ± 0
U20325_at	98,32	97.74	98,65	1	1 ± 0

* TPR (**T**rue **P**ositive **R**ate) - TNR(**T**rue **N**egative **R**ate)

the dataset. The parameter ϕ is initially set to 2 and every time a neuron is added it is reduced on a Fi_{temp}%.

In all the simulations that has been launched for this model, the parameters of the algorithm were initialize to: $I_{max} = 10000$, $g_{fac} = 0.2$ and $Fi_{temp} = 10$.

3 Results and Conclusion

The final results on both ANN and CMantec algorithm are shown in table 1, in which two important results could be highlighted. On one hand, execution time costs using CMantec algorithm are considerably smaller than using ANN algorithm. The column labelled as CPU time indicates an orientative execution time to estimate any model with n genes (first column in table 1). An approximation to the ratio between the execution times for both models is computed by averaging over values in columns CPU times, obtaining 254.83 and 14.75 hours for ANN and CMantec models respectively. This means that CMantec algorithm needs aproximately less than 17 times the execution time of the ANN model.

On the other hand, CMantec model needs less genes than ANN model in order to get a better generalization over the test data subset. CMantec requires only 4 genes while ANN model needs 2 more genes in order to get their respective maximum predictive performance. Moreover, CMantec outperforms the ANN model in prognosis accuracy (98.32% against 96.3%). Also, sensitivity (TPR) and especificity (TNR) were computed, and the results showed the improvement in both pronostic indexes by the CMantec model.

An analysis of variance (ANOVA) test was then applied to compare the mean values for prognosis accuracy between CMANTEC and ANN algorithms. A p-value<0.001 indicated that differences in prognosis accuracies were found statistically significant.

As conclusion, this work presents a novel contructive neural network (CMantec) that has been applied to identify the bests gene transcript signatures predictive for estrogen receptor. The CMantec model clearly outperforms the ANN model in process execution time and final prognosis accuracy. Thus, CMantec appears as a powerful tool to identify gene signatures that predict the ER status for a given patient, also reducing the number of gene transcripts selected as inputs to the system. We will continue studying on this research line testing and validating our model against other datasets and other types of cancer.

Acknowledgments

The authors acknowledge support from CICYT (Spain) through grant TIN2008-04985 (including FEDER funds) and from Junta de Andalucía through grant P08-TIC-04026. Leonardo Franco acknowledges support from the Spanish Ministry of Science and Innovation (MICIIN) through a Ramón y Cajal fellowship.

References

1. Andree, H.M.A., Barkema, G.T., Lourens, W., Taal, Vermeulen, J.C.: A comparison study of binary feedforward neural networks and digital circuits. Neural Networks 6, 785–790 (1993)
2. Baum, E.B., Haussler, D.: What size net gives valid generalization? Neural Computation 1, 151–160 (1989)
3. Frean, M.: The upstart algorithm: A method for constructing and training feedforward neural networks. Neural Computation 2, 198–209 (1990)
4. Frean, M.: Thermal perceptron learning rule. Neural Computation 4, 946–957 (1992)
5. Gómez, I., Franco, L., Jerez, J.M.: Neural Network Architecture Selection: Can function complexity help? Neural Processing Letters (in press, 2009) doi:10.1007 s11063-009-9108-2
6. Keibek, S.A.J., Barkema, G.T., Andree, H.M.A., Savenlie, M.H.F., Taal, A.: A fast partitioning algorithm and a comparison of binary feedforward neural networks. Europhys. Lett. 18, 555–559 (1992)
7. Lancashire, L.J., Rees, R.C., Ball, G.R.: Identification of gene transcript signatures predictive for estrogen receptor and lymph node status using a stepwise forward selection artifical neural network modelling approach. Artificial Intelligence in Medicine 43, 99–111 (2008)
8. Lawrence, S., Giles, C.L., Tsoi, A.: What Size Neural Network Gives Optimal Gener- alization? Convergence Properties of Backpropagation. Technical Report UMIACS-TR-96-22 and CS-TR-3617, University of Maryland (1996)
9. Mezard, M., Nadal, J.P.: Learning in feedforward layered networks: The tiling algorithm. J. Physics A 22, 2191–2204 (1989)
10. Nicoletti, M.C., Bertini, J.R.: An empirical evaluation of constructive neural network algorithms in classification tasks. International Journal of Innovative Computing and Applications 1, 2–13 (2007)
11. Parekh, R., Yang, J., Honavar, V.: Constructive Neural-Network Learning Algorithms for Pattern Classification. IEEE Transactions on Neural Networks 11, 436–451 (2000)
12. García-Pedrajas, N., Ortiz-Boyer, D.: A cooperative constructive method for neural networks for pattern recognition. Pattern Recognition 40, 80–98 (2007)
13. Pellagatti, A., Vetrie, D., Langford, C.F., Gama, S., Eagleton, H., Wainscoat, J.S., Boultwood, J.: Gene Expression Profiling in Polycythemia Vera Using cDNA Microarray Technology. Cancer Res. 63, 3940–3944 (2003)
14. Linder, R., Richards, T., Wagner, M.: Microarray data classified by artificial neural networks. Methods Mol. Biol. 382, 345–372 (2007)
15. Rosenhlatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review 65, 386–408 (1959)
16. Rumelhart, D., Hinton, G., Williams, R.: Learning internal representations by backpropagating errors. In: Rumelhart, D., Mc-Clelland, J. (eds.) Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1, pp. 318–362. MIT Press, Cambridge (1986)
17. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220, 671–680 (1983)
18. Subirats, J.L., Franco, L., Gòmez, I., Jerez, J.M.: Computational capabilities of feedforward neural networks: the role of the output function. In: Proceedings of the XII CAEPIA'07, vol. II, pp. 231–238 (2008) ISBN: 978-84-611-8848-2

19. Subirats, J.L., Jerez, J.M., Franco, L.: A New Decomposition Algorithm for Threshold Synthesis and Generalization of Boolean Functions. IEEE Transactions on Circuits and Systems I 55, 3188–3196 (2008)
20. Subirats, J.L., Franco, L., Molina, I., Jerez, J.M.: Competition and Stable Learning for Growing Compact Neural Architectures with Good Generalization Abilities: The C-Mantec Algorithm (2009) (sent for publication)
21. Utgoff, P.E., Stracuzzi, D.J.: Many-Layered Learning. Neural Computation 14, 2497–2539 (2002)
22. Wei Jun, S., Greer Braden, T., Frank, W., Steinberg Seth, M., Chang-Gue, S., et al.: Prediction of Clinical Outcome Using Gene Expression Profiling and Artificial Neural Networks for Patients with Neuroblastom. Cancer Res. 64, 6883–6891 (2004)
23. West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., et al.: Predicting the clinical status of human breast cancer by using genes expression profiles. Proc. Natl. Acad. Sci. U.S.A. 98, 11462–11467 (2001)
24. Xu, Y., Selaru, F.M., Yin, J., Zou, T.T., Shustova, V., Mori, Y., Sato, F., et al.: Prediction of Clinical Outcome Using Gene Expression Profiling and Artificial Neural Networks for Patients with Neuroblastom. Cancer Res. 62, 3493–3497 (2002)

Class Imbalance Methods for Translation Initiation Site Recognition*

Nicolás García-Pedrajas[1], Domingo Ortiz-Boyer[1], María D. García-Pedrajas[2], and Colin Fyfe[3]

[1] Department of Computing and Numerical Analysis, University of Córdoba, Spain
npedrajas@uco.es, dortiz@uco.es
http://www.cibrg.org/

[2] Experimental Station La Mayora, CSIC, Málaga, Spain
mariola@eelm.csic.es

[3] School of Computing, University of the West of Scotland, United Kingdom
colin.fyfe@uws.ac.uks

Abstract. Translation initiation sites (TIS) recognition is one of the first steps in gene structure prediction, and one of the common components in any gene recognition system. Many methods have been described in the literature to identify TIS in transcripts such as mRNA, EST and cDNA sequences. However, the recognition of TIS in DNA sequences is a far more challenging task, and the methods described so far for transcripts achieve poor results in DNA sequences. Most methods approach this problem taking into account its biological features. In this work we try a different view, considering this classification problem from a purely machine learning perspective.

From the point of view of machine learning, TIS recognition is a class imbalance problem. Thus, in this paper we approach TIS recognition from this angle, and apply the different methods that have been developed to deal with imbalance datasets.

Results show an advantage of class imbalance methods with respect to the same methods applied without considering the class imbalance nature of the problem. The applied methods are also able to improve the results obtained with the best method in the literature, which is based on looking for the next in-frame stop codon from the putative TIS that must be predicted.

1 Introduction

Translation initiation sites (TIS) recognition consists of identifying the start codon, ATG, which marks the beginning of the translation in most genes. Most previous approaches have focused on recognizing TIS in transcripts. However, recognizing TIS in genomic sequences is a different, and more difficult task. Transcripts usually contain one or zero TIS, and no introns. On the other hand,

* This work has been financed in part by the Excellence in Research Project P07-TIC-2682 of the Junta de Andalucía.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 327–336, 2010.

in a generic genetic sequence, we can find the ATG codon, and thus a putative TIS, in any place. In this work we consider the most difficult case of analyzing genomic sequences that contain junk DNA, exons, introns and untranslated terminal regions (UTRs). The different features of recognizing TIS in transcripts and genomic sequences are illustrated in the different performance of the predictors in each problem. TisMiner [11] is one of the best performing programs for TIS recognition in transcripts which is able to achieve a specificity of 98% at a sensitivity level of 80%. However, when tested in genomic sequences, its performance at the same level of sensitivity drops to a specificity of 55%.

One of the most important features of TIS prediction in genome sequences is the fact that negative instances outnumber positive instances by many times. In machine learning theory, this is called the class imbalance problem [1]. Most learning algorithms expect a somewhat balanced distribution of instances among the different classes. It has been shown that learning algorithms suffer from the skewed distribution that is associated with class imbalance. However, most of the previous papers in TIS recognition have not considered addressing this problem from the point of view of class imbalance methods.

In this paper we approach the recognition of TIS as a class imbalance problem. We test whether class imbalance methods are able to achieve the same performance of the methods designed specifically for TIS recognition from the biological point of view.

This paper is organized as follows: Section 2 summarizes the most important aspects of class imbalance problems; Section 3 shows the experimental setup and the results obtained; and finally Section 4 states the conclusions of our work and the future research lines.

2 Class Imbalance Problems

It has been repeatedly shown that most classification methods suffer from a imbalanced distribution of the training instances among the classes [3]. Most learning algorithms expect an approximately even distribution of the instances among the different classes and suffer, in different degrees, when that is not the case. Dealing with the class imbalance problem is a difficult task, but a very relevant one as many of the most interesting and challenging real-world problems have a very uneven class distribution, such as gene recognition, intrusion detection, web mining, etc.

In most cases this problem appears in two class datasets. There is a class of interest, the positive class, which is highly underrepresented in the dataset, together with a negative class which accounts for most of the instances. In highly imbalance problems the ratio between the positive and the negative class can be as high as 1:1000 or 1:10000. Many algorithms and methods have been proposed to ameliorate the effect of class imbalance on the performance of the learning algorithms. There are mainly three different approaches [16] [6]:

- Internal approaches acting on the algorithm. These approaches modify the learning algorithm to deal with the imbalance problem. They can adapt the

decision threshold to create a bias towards the minority class or introduce costs in the learning process to compensate the minority class.
- External approaches acting on the data. These algorithms act on the data instead of on the learning method. They have the advantage of being independent from the classifier used. There are two basic approaches, oversampling the minority class and undersampling the majority class.
- Combined approaches which are based on *boosting* [5] taking into account the imbalance in the training set. These methods modify the basic boosting method to account for the minority class underrepresentation in the dataset.

There are two principal advantages of sampling against cost sensitive methods. Firstly, sampling is more general as it does not depend on the possibility of adapting a certain algorithm to work with classification costs. Secondly, the learning algorithm is not modified, which can be in many cases a difficult task and also adds additional parameters to be fine tuned.

As stated, data driven algorithms can be broadly classified in two groups. Undersampling the majority class and oversampling the minority class. There are also algorithms that combine both processes. Both undersampling and oversampling can be made randomly or with a more complicated process searching for least/most useful instances. Previous works have shown that undersampling the majority class lead to better results than oversampling the minority class [3], at least, when oversampling is performed using sampling with replacement from the minority class. Furthermore, combining undersampling on the majority class with oversampling of the minority class has not yield to better results than undersampling of the majority class alone [10]. One of the possible sources of this worse performance of oversampling is the fact that there is no new information introduced in the training set, as oversampling must rely on adding new copies of the minority class instances already in the dataset.

2.1 Undersampling

The first method for balancing a dataset is undersampling the majority class until both classes have the same number of instances. We have not used oversampling methods because all previous works agree that undersampling performs better than oversampling. Additionally, as we are dealing with very large datasets, oversampling will make the datasets almost twice in size, preventing the use of some of the most interesting classifiers, such as support vectors machines.

Random undersampling consists of randomly removing instances from the majority class until a certain criterion is reached. In most works instances are removed until both classes have the same number of instances. Several studies comparing sophisticated undersampling methods with random undersampling [7] have failed to establish a clear advantage of the formers. Thus, in this work we consider first random undersampling. However, the problem with random undersampling is that many, potentially useful, samples from the majority class are ignored. Liu et al. [12] propose two ensemble methods combining undersampling and boosting to avoid that problem. This methodology is called exploratory undersampling. The two proposed methods are called `EasyEnsemble` and BalanceCascade. We describe these two methods in more detail as we have tested both experimentally. EasyEnsemble consists of applying repeatedly the standard ensemble method AdaBoost [2] to different samples of the majority class. Algorithm 1 shows EasyEnsemble method. The idea behind EasyEnsemble is generating T balanced subproblems sampling from the majority class.

```
Data   : A minority training set $\mathcal{P}$ and a majority training set $\mathcal{N}$, $|\mathcal{P}| \ll |\mathcal{N}|$,
         a number of subsets $T$ to sample from $\mathcal{N}$ and $s_i$ the number of
         iterations to train the AdaBoost ensemble $H_i$.
Result : The final ensemble: $H(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{T}\sum_{j=1}^{s_i}\alpha_{i,j}h_{i,j}(\mathbf{x}) - \sum_{i=1}^{T}\theta_i\right)$.
for $i = 1$ to $T$ do
    Randomly sample a subset $\mathcal{N}_i$ from $\mathcal{N}$, $|\mathcal{N}_i| = |\mathcal{P}|$
    Learn $H_i$ using $\mathcal{P}$ and $\mathcal{N}_i$. $H_i$ is an AdaBoost ensemble with $s_i$ weak
    classifiers $h_{i,j}$ and corresponding weights $\alpha_{i_j}$. The ensemble threshold is $\theta_i$.
    $H_i$ is given by: $H_i(\mathbf{x}) = \text{sgn}\left(\sum_{j=1}^{s_i}\alpha_{i,j}h_{i,j}(\mathbf{x}) - \theta_i\right)$.
end
```

Algorithm 1. EasyEnsemble method

EasyEnsemble is an unsupervised strategy to explore the set of negative instances, $\mathcal{N}$, as the sampling is made without using information from the classification performed by the previous members of the ensemble. BalanceCascade, on the other hand, explores $\mathcal{N}$ in a supervised manner, removing from the majority class those instances that have been correctly classified by the previous classifiers added to the ensemble. BalanceCascade is shown in Algorithm 2.

```
Data   : A minority training set $\mathcal{P}$ and a majority training set $\mathcal{N}$, $|\mathcal{P}| \ll |\mathcal{N}|$,
         a number of subsets $T$ to sample from $\mathcal{N}$ and $s_i$ the number of
         iterations to train the AdaBoost ensemble $H_i$.
Result : The final ensemble: $H(\mathbf{x}) = \text{sgn}\left(\sum_{i=1}^{T}\sum_{j=1}^{s_i}\alpha_{i,j}h_{i,j}(\mathbf{x}) - \sum_{i=1}^{T}\theta_i\right)$.
$f = \sqrt[T-1]{|\mathcal{P}|/|\mathcal{N}|}$. $f$ is the false positive rate that $H_i$ should achieve.
for $i = 1$ to $T$ do
    Randomly sample a subset $\mathcal{N}_i$ from $\mathcal{N}$, $|\mathcal{N}_i| = |\mathcal{P}|$
    Learn $H_i$ using $\mathcal{P}$ and $\mathcal{N}_i$. $H_i$ is an AdaBoost ensemble with $s_i$ weak
    classifiers $h_{i,j}$ and corresponding weights $\alpha_{i,j}$. The ensemble threshold is
    $\theta_i$. $H_i$ is given by: $H_i(\mathbf{x}) = \text{sgn}\left(\sum_{j=1}^{s_i}\alpha_{i,j}h_{i,j}(\mathbf{x}) - \theta_i\right)$.
    Adjust $\theta_i$ such that $H_i$'s false positive rate is $f$.
    Remove form $\mathcal{N}$ all examples that are correctly classified by $H_i$.
end
```

Algorithm 2. BalanceCascade method

2.2 SMOTE-N

One of the problems with oversampling is that merely making copies of the minority class samples does not add new information to the dataset, and the learning method is not able to significantly improve the classification of the minority class. To overcome this problem, Chawla et al. [3] proposed a method called SMOTE, that combines undersampling of the majority class with oversampling of the minority class. However, instead of oversampling the minority class just making copies of the minority class samples, SMOTE generates synthetic instances from actual instances of the minority class. Synthetic samples are generated in the following way: take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. In this way, we can generate new samples that share the main features of actual instances for a more dense dataset.

The original algorithm is proposed for numerical attributes. However, an algorithm, called SMOTE-N, is also proposed for nominal attributes as it is our case. The procedure for generating this subset of synthetic samples, SMOTE-N, is shown in Algorithm 3. SMOTE-N differs from standard SMOTE in using a modified version of the value difference metric (VDM), proposed by Cost and Salzberg [4], instead of the Euclidean distance.

```
Data   : The subset of misclassified instances $S_M = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, and the number of synthetic instances to
         generate for each instance $N_s$.
Result : The populated $S_M$ with the instances added.
for $t = 1$ to $N$ do
    Obtain $k$ nearest neighbors of $\mathbf{x}_t$ using modified VDM distance
    // Generate $N_s$ synthetic samples
    for $i = 1$ to $N_s$ do
        for $j = 1$ to $D$ do
            $\mathbf{x}^i_{new}$ = select randomly a value from the corresponding feature $i$
            within the set of nearest neighbors of the same class as $\mathbf{x}_t$.
        end
        Add $\mathbf{x}_{new}$ to $S_M$
    end
end
```

Algorithm 3. SMOTE-N algorithm

2.3 Evaluation Measures

Accuracy is not a useful measure for imbalanced data, specially when the minority class is very rare. If we have a ratio of 1:100, a classifier that assigns all instances to the majority class will have a 99% accuracy. Several measures [16] have been developed to take into account the imbalance nature of the problems. Given the ratio of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) we can define the following set of basic measures:

- True positive rate TP_{rate}, recall R o sensitivity. $TP_{rate} = R = \frac{TP}{TP+FN}$. This measure is relevant if we only are interested in the performance on the positive class.
- True negative rate TN_{rate} or specificity. $TN_{rate} = \frac{TN}{TN+FP}$.
- False positive rate FP_{rate}. $FP_{rate} = \frac{FP}{TN+FP}$.
- False negative rate FN_{rate}. $FN_{rate} = \frac{FN}{TP+FN}$.
- Positive predictive value PP_{value} or precision P. $PP_{value} = \frac{TP}{TP+FP}$.
- Negative predictive value NP_{value}. $NP_{value} = \frac{TN}{TN+FN}$

From these basic measures others have been proposed. F-measure was proposed [9] to join recall and precision in a measure that is a harmonic mean of both:

$$F = \frac{2RP}{R+P} = \frac{2}{1/R+1/P}. \tag{1}$$

The harmonic mean of two measures tends to be closer to the smaller one than the arithmetic mean. Thus, F measures whether recall and precision have both high values. If we are concerned about the performance on both negative and positive classes $G - mean$ measure [8] considers both:

$$G - mean = \sqrt{TP_{rate} \cdot TN_{rate}}. \tag{2}$$

$G - mean$ measures the balance performance of the learning algorithm between the two classes. In this work we will use both FP and FN rates and as an average measure the $G - mean$.

Many classifiers are subject to some kind of threshold that can be varied to achieve different values of the above measures. For that kind of classifiers receiver operating characteristic (ROC) curves can be constructed. A ROC curve, is a graphical plot of the TP_{rate} (sensitivity) against the FP_{rate} (1 - specificity) for a binary classifier system as its discrimination threshold is varied. The perfect model would achieve a true positive rate of 1 and a false positive rate of 0. A random guess will be represented by a line connecting the points $(0,0)$ and $(1,1)$. ROC curves are a good measure of the performance of the classifiers. Furthermore, from this curve a new measure, area under the curve (AUC), can be obtained which is a very good overall measure for comparing algorithms.

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 or can be retrieved from `http://cibrg.org/`.

3 Experimental Setup and Results

Among the different methods for dealing with class imbalance problems we have selected the most successful ones in the literature. We have used random undersampling, SMOTE-N, `BalanceCascade` and `EasyEnsemble`. Other methods were tried with worse results. We must take into account that the two former are single classifier methods, while the two latter, are ensemble methods. Due to their increased complexity, `BalanceCascade` and `EasyEnsemble` must achieve a significantly better performance than undersampling and SMOTE-N to be competitive. As base learners we used a C4.5 decision tree [13], a support vector machine (SVM) [15] and a k-nearest neighbors classifier.

We have performed our experiments using cross-validation for setting the values of the parameters. For each one of the classifiers used we have obtained the best parameters from a set of different values. For SVM we tried a linear kernel with $C \in \{0.1, 1, 10\}$, and a Gaussian kernel with $C \in \{0.1, 1, 10\}$ and $\gamma \in \{0.0001, 0.001, 0.01, 0.1, 1, 10\}$, testing all the 21 possible combinations. For C4.5 we tested 1 and 10 trials and softening of thresholds trying all the 4 possible combinations. For k-NN the value of k is obtained by cross-validation in the interval $[1, 100]$.

3.1 Datasets

We have used two datasets for testing the performance of the described methods. The CCDS dataset was compiled by Saeys et al. [14] from the consensus CDS database. The CCDS project is a collaborative effort of compiling and identifying a core of human genes that are accurately annotated. The annotation is a mix of manual curation and automatic annotation. CCDS contains 350,578 negative samples and 13,917 positive samples with a positive negative ratio of 1:25. The ustilago datasets is a set of coding and non-coding regions of genomic sequences from the sequencing of *Ustilago Maydis*. The sequences are first obtained from the Broad Institute[1] and then completed with the information of the Munich Information Center for Protein Sequences (MIPS)[2]. The ustilago dataset contains 607,926 negative samples and 6,510 positive samples with a ratio of 1:93. For estimating the testing error we used a 10-fold cross-validation (cv) method. Our aim was to test the proposed methods in two very different datasets to test whether the problems is more difficult depending on the organism.

Tables 1 and 2 show the results for the two datasets. These results are illustrated in Figure 1. The first conclusion we can obtain from these results is that the two datasets pose similar problems. In both cases the values of FP and FN rates are very similar, being CCDS dataset a somewhat harder problem.

The performance of the stop codon method, which proved the best in [14], is very unstable. For CCDS datasets shows a good FP rate but a very poor FN rate. On the other hand, for ustilago, the performance in terms of FP rate

[1] `http://www.broadinstitute.org/annotation/genome/ustilago_maydis/`

[2] `http://www.helmholtz-muenchen.de/en/mips/home/index.html`

Table 1. Results for CCDS dataset using the different class imbalance methods

Method	Classifier	FP rate	FN rate	G-measure
—	Stop codon	26.75	65.61	0.5081
None	C.45	0.62	87.61	0.3507
Undersampling	C4.5	24.95	40.55	0.6679
	SVM	15.68	18.69	0.8278
	k-NN	37.41	22.69	0.6955
SMOTE-N	C4.5	26.66	31.83	0.7071
	SVM	6.20	24.44	**0.8419**
	k-NN	30.03	14.74	0.7724
`EasyEnsemble`	C4.5	33.10	15.89	0.7515
`BalanceCascade`	C4.5	48.34	9.94	0.6817

Table 2. Results for ustilago dataset using the different class imbalance methods

Method	Classifier	FP rate	FN rate	G-measure
—	Stop codon	22.89	0.35	0.8766
None	C.45	0.00	100.00	0.0000
Undersampling	C4.5	26.83	28.68	0.7223
	SVM	10.23	18.05	0.8505
	k-NN	27.96	6.40	0.8211
SMOTE-N	C4.5	24.49	28.69	0.7029
	SVM	6.79	16.00	**0.8848**
	k-NN	20.39	8.82	0.8520
`EasyEnsemble`	C4.5	29.23	15.37	0.7738
`BalanceCascade`	C4.5	25.64	19.66	0.7728

is rather good, and excellent in terms of FN rate. The performance of random undersampling is acceptable if we compare the results with stop codon method. Due to the large datasets, only C4.5 can be applied without resorting to undersampling. The table shows that undersampling is useful in reducing FP rates for both datasets, with a very significant improvement. In fact, C4.5 is useless using all the instances, as it assigns almost all of them to the majority class.

With regards to SMOTE-N its behavior depends on the classifier. For C4.5 its performance is very similar to the performance of undersampling. For ustilago the results as almost the same, and for CCDS is able to improve the FN rate while keeping the FP rate in a very similar value. On the other hand, using a SVM, SMOTE-N is very efficient in reducing FP rate for both problems. For ustilago is even able to reduce FN rate. However, for CCDS it increases the FN rate from 10.69% to 24.44%. For k-NN SMOTE-N is better than random undersampling. For CCDS improves both FP and FN rates with respect to undersampling. For ustilago it improves FP rate, although FN rate is slightly worse.

Finally, we have tested the two methods `EasyEnsemble` and `BalanceCascade`. These two methods are applied using C4.5 as base learner. SVM cannot be used as the computational cost is too high. k-NN is not used as it has been shown that

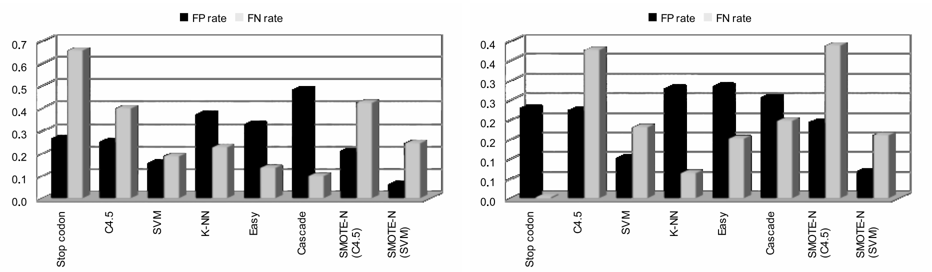


Fig. 1. FP and FN rates for CCDS for the different methods

this classifier is not efficient as a member of an ensemble. The results of both `EasyEnsemble` and `BalanceCascade` are different depending on the problem. For CCDS they are able to significantly improve FP rate, but with the side effect if significantly incrementing FN rate. However, for ustilago both methods significantly improve the FN rate without damaging the FP rate.

4 Conclusion

In this work we have shown that the methods developed to cope with class imbalance problems can be useful for approaching TIS recognition problems. We have tested four methods designed to deal with class imbalance problems. These methods are able to improve the results of stop codon method, which it is the best reported in [14].

The results show that simple random undersampling is a very competitive method when compared with more complex ones. SMOTE-N achieved a good performance improving random undersampling in several cases. On the other hand, the performance of `EasyEnsemble` and `BalanceCascade` depends on the problem, and the added complexity might not be justified by the results.

As future research we are working on testing new class imbalance methods. Also, Zien et al. [17] developed specific kernels for TIS recognition that we want to test together with class imbalance methods.

References

1. Barandela, R., Sánchez, J.L., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition 36, 849–851 (2003)
2. Bauer, E., Kohavi, R.: An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Machine Learning 36(1/2), 105–142 (1999)
3. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
4. Cost, S., Salzberg, S.: A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning 10(1), 57–78 (1993)

5. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proc. of the Thirteenth International Conference on Machine Learning, Bari, Italy, pp. 148–156 (1996)
6. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evolutionary Computation 17(3), 275–306 (2009)
7. Japkowicz, N.: The class imbalance problem: significance and strategies. In: Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning, Las Vegas, USA, vol. 1, pp. 111–117 (2000)
8. Kubat, M., Holte, R., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Machine Learning 30, 195–215 (1998)
9. Lewis, D., Gale, W.: Training text classifiers by uncertainty sampling. In: Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information, New York, USA, pp. 73–79 (1998)
10. Ling, C., Li, G.: Data mining for direct marketing problems and solutions. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98). AAAI Press, New York (1998)
11. Liu, H., Han, H., Li, J., Wong, L.: Using amino acids patterns to accurately predict translation initiation sites. Silico Biology 4, 255–269 (2004)
12. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 39(2), 539–550 (2009)
13. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)
14. Saeys, Y., Abeel, T., Degroeve, S., de Peer, Y.V.: Translation initiation site prediction on a genomic scale: beauty in simplicity. Bioinformatics 23, 418–423 (2007)
15. Schölkopf, B., Smola, A., Williamson, R., Bartlett, P.L.: New support vector algorithms. Neural Computation 12, 1207–1245 (2000)
16. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40, 3358–3378 (2007)
17. Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., Müller, K.R.: Engineering support vector machines kernels that recognize translation initiation sites. Bioinformatics 16(9), 799–807 (2000)

On-Line Unsupervised Segmentation for Multidimensional Time-Series Data and Application to Spatiotemporal Gesture data

Shogo Okada, Satoshi Ishibashi, and Toyoaki Nishida

Dept. of Intelligence Science and Technology, Graduate School of Informatics,
Kyoto University

Abstract. This paper proposes an on-line unsupervised segmentation algorithm of multidimensional time-series data. On-line unsupervised segmentation algorithm is important for discovery of frequent patterns from a large amount of multidimensional sensor data such as whole motion data and multi-modal data, because real-world problems are often dynamic and incremental:the input data may change over time and on-line segmentation methods are significantly faster than batch methods. We enhance the Sliding Window and Bottom-up (SWAB) algorithm toward the implementation of the segmentation algorithm of multidimensional data and propose MD-SWAB (Multi Dimensional SWAB). To evaluate the proposed algorithm, we use the continuous hand gesture data observed with a motion capture system. We have found that MD-SWAB outperforms a comparative algorithm in the segmentation performance and is significantly faster than the existing algorithm. In addition, we combine it with the on-line clustering method HB-SOINN to label the detected segments. The result of experiment shows that gesture patterns which are belong to same category are represented as similar label sequences.

Keywords: Unsupervised Learning, Gesture Recognition, Time-series Analysis.

1 Introduction

In the previous decade, there has been an explosion of interest in mining time series (temporal) databases. Unsupervised segmentation algorithms [4] and change point detection algorithms [2] of time-series data are proposed in the domain of data mining. These algorithms have been available for detection of segmentation points, but not available for clustering (grouping) of segmented sequence and labeling me. SAX algorithm [8] is the symbolic representation for time series that allows for dimensionality reduction and indexing with a lower-bounding distance measure in batch form. In SAX, we need to set the word length (number of symbols per subsequence) and vocabulary size (number of different symbols) parameters. Unfortunately there is no way to find the optimal values of these parameters. In addition, batch segmentation approaches produce better results,

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 337–347, 2010.

but are off-line and require scanning of the entire data set [4]. This is impractical, where large amounts of data arrive in streams. The algorithms proposed in [4], [2], [3], [8] are available for single dimensional time-series data, but are not available for multidimensional data directly. To analyze human actions in daily activities, which are observed from multi-sensor such as motion capture system, we need to segment multidimensional time-series data which is a kind of spatiotemporal data and cluster it. To detect segmentation points from multidimensional time-series data, we need to decide one segmentation point from segmentation points which are detected from each time series data.

In this paper, we enhance the Sliding Window and Bottom-up algorithm [4] of one dimensional data, toward the implementation of the segmentation algorithm of multidimensional data and propose MD-SWAB (Multi Dimensional SWAB). To apply MD-SWAB to multidimensional data, SWAB is first applied to every dimension of the time-series data. We propose the MD-SWAB algorithm for combining of resulting segmentation points from multidimensional data. Each SWAB algorithm has a role to segment each dimension of time-series data. Merged error values in segmentation points are calculated from each-dimensional time-series data. The point at which the density of merged error in the window interval is high is detected as the segmentation point. We explain the detail of this mechanism in Section 3.1. In addition, by combining it with the on-line unsupervised clustering algorithm: HB-SOINN [9], we propose a novel approach that is able to segment multidimensional time-series data and perform clustering of the segmented gesture primitives and label it in an on-line manner.

HB-SOINN is on-line clustering method of time-series data. [9] reports that HB-SOINN has markedly improved the clustering performance over that of comparative clustering methods.

To evaluate the proposed approach, we use hand gesture data sets from motion capture and compare the on-line unsupervised segmentation performance (recall, precision and F value) with a conventional algorithm. We have found that MD-SWAB outperforms the conventional algorithm. The contribution of this paper is to propose the effective segment algorithm of multidimensional time-series data (spatiotemporal motion data). The method will help people who analyze a lot of multi dimensional sensor data.

2 Related Work

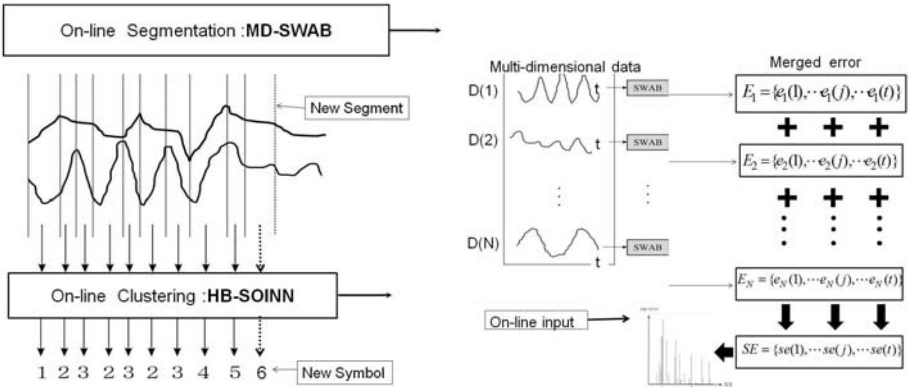
Kohlmorgen and Lemm [5] proposed an on-line unsupervised segmentation algorithm. The segmentation algorithm uses a HMM to represent the incoming data sequence, where each model state represents the probability density estimate over a window of the data. The segmentation is implemented by finding the optimum state sequence over the developed model.

Kulic and Nakamura proposed a modified version of the Kohlmorgen and Lemm algorithm [5] for unsupervised segmentation of on-line human motion data [6], and then input the extracted segments into an automated clustering and hierarchical organization algorithm [7].

The method [5] takes much time to find the optimum state sequence over the developed model and segmentation of high dimensional data. Our method which extends to SWAB is faster than this method. In this paper, we compare the on-line unsupervised segmentation performance of MD-SWAB and the Kohlmorgen and Lemm algorithm [5], and examine the effectiveness of MD-SWAB.

3 Proposed Approach

Our proposed approach is composed from two layered algorithms. Figure 1(a) shows the architecture of proposed system. In the upper layer, segmentation of the multidimensional time-series data is performed (segmentation layer). In the first step of the layer, the segmentation point in one dimensional time-series data is detected by using the SWAB algorithm [4]. In the second step, we integrate the segmentation points which are calculated from each dimensional signal, then detect the segmentation point of multidimensional time-series data.


On-line Segmentation :MD-SWAB
New Segment
On-line Clustering :HB-SOINN
1 2 3 2 3 2 3 4 5 6
New Symbol
Multi-dimensional data
D(1)
D(2)
D(N)
t
SWAB
Merged error
$E_1 = \{e_1(1), \cdots e_1(j), \cdots e_1(t)\}$
$E_2 = \{e_2(1), \cdots e_2(j), \cdots e_2(t)\}$
$E_N = \{e_N(1), \cdots e_N(j), \cdots e_N(t)\}$
$SE = \{se(1), \cdots se(j), \cdots se(t)\}$
On-line input

(a) Overview of MD-SWAB

(b) Procedure of MD-SWAB

Fig. 1. MD-SWAB (Multi-dimensional Sliding Window and Bottom-up) algorithm

In the lower layer, unsupervised clustering of segmented gesture primitives is performed. We use HB-SOINN for clustering of the segmented primitives, and for labeling of unlabeled primitives. The advantage of the proposed algorithm is that it is able to segment the continuous gesture and labels (indexes) them in an on-line manner.

3.1 On-Line Segmentation of Time-Series Data

In this section, we explain the base algorithm: SWAB algorithm and MD-SWAB which enhances SWAB.

SWAB algorithm. SWAB algorithm combines the advantages of a bottom-up segmentation algorithm which produces precise results with that of a sliding-window algorithm of on-line manner. At first, we buffer w_0 frames (data) as an initial buffer. The algorithm keeps buffering the small sequence (buffer) of the input data.

A bottom-up segmentation is applied to the data in the small buffer. After bottom-up segmentation, the segment with the oldest data is extracted from the buffer and becomes a segment. New data is incorporated into the buffer This process is performed by Sliding Window and Bottom-Up is applied again. This process of applying Bottom-Up to the buffer, reporting the leftmost segment, and reading in the next subsequence is repeated as long as data arrives potentially forever.

The bottom-up algorithm begins by creating the finest possible approximation of the time series, so that n/2 segments are used to approximate the n-length time series. Next, the cost of merging each pair of adjacent segments is calculated, and the lowest cost pair in the buffer is iteratively merged until all segments in the buffer exceed a merged error threshold Th when merged. For calculation of merged error, the least square method is used.

We define the merged error as the differences between the sum of least square error values of two segments and least square error value of the segment which is merged from two segments. When we use a too large buffer size for the SWAB algorithm, the algorithm is similar to Bottom-up algorithm (batch mode). When we use a too short buffer size for it, it is similar to the sliding window algorithm and the segmentation accuracy is decreased. To prevent this problem, we set the upper bound of the buffer size and the lower bound of it as $2w_0$ and $w_0/2$ respectively.

Details of SWAB are described elsewhere in the literature [4].

MD-SWAB algorithm. Figure 1 shows the MD-SWAB system. In MD-SWAB, segmentation process is done in an on on-line manner. The number of dimensions is defined as N. When t frames time-series data are input, the merged error sequence is defined as:

$E_i = \{e_i(1), \cdots, e_i(j), \cdots, e_i(t)\}$, where i is dimension number: $1 \leq i \leq N$. If the j th frame is reported as a segmentation point by SWAB, $e_i(j)$ equals $ME_i(j)$, where $ME_i(j)$ is the merged error which is calculated by SWAB.

If the j th frame is not the segmentation point and then $e_i(j) = 0$. Second, sum of the merged error is calculated in each frame and the sequence of it is defined as: $SE = \{se(1), \cdots, se(j), \cdots, se(t)\}$, where $se(j)$ is defined as : $se(j) = \sum_{i=1}^{N} e_i(j)$ Next, we implement a sliding window algorithm to the sequence SE. If the density of SME: $DSME$ in buffer window$w$ satisfies next conditional equations, we define $sp(k) = l$ as the segmentation point (frame), where k is index of segmentation point.

$$\frac{1}{w}\sum_{j=l}^{w+l} se(j) > C \times Th \qquad (1)$$

In Equation (1), the point where the density of $se(j)$ is high is detected as the segmentation point, where C is the proportionality factor. Figure 2(a) shows the decision method of segmentation points. In addition, we set the constraint condition that two or more segments do not exist in buffer w as $sp(k) + w < sp(k+1)$. The process of MD-SWAB is described in the following paragraph.

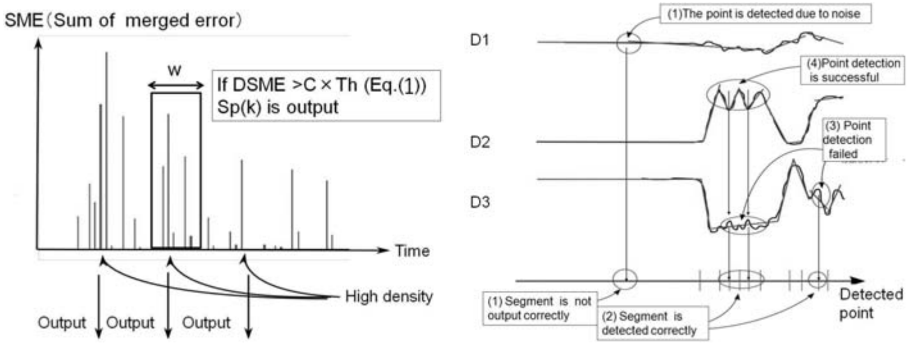
MD-SWAB algorithm

Step1. Perform SWAB to each dimensional time-series data.
Step2. Extract the merged error sequence E_i after segmentation of data of dimension i by SWAB.
Step3. Sum the merged error sequence E_i and calculate SE.
Step4. Define the point as the segmentation point by using the conditional equation Equation (1).

We need to set 4 parameters for MD-SWAB. These parameters are initial buffer size w_0, merged error threshold Th, window size W and proportionality factor C for calculating $DSME$.

Two parameters W will influence the frequency of detection of segmentation points. If we choose a large value for W, many segmentation points will be merged to one segmentation point incorrectly. If we choose a small value for W, the segmentation point that the density of $se(j)$ is high and the segmentation point will be not detected.

By controlling parameters W and C, we aim to remove the segmentation point which is detected from only one dimensional time series data and the segmentation point which is detected due to the influence of observation noise of motion capture system ((1) in Figure 2(a)). On the other hand, when the segmentation candidate points are not detected incorrectly ((3) in Figure 2(a)) and the other segmentation candidates are detected correctly ((4) in Figure 2(a)), the output of the segmentation point is successful ((2) in Figure 2(a)). An example of processing by MD-SWAB is shown in Figure 2(a).


SME (Sum of merged error)
W
If DSME >C × Th (Eq.(1))
Sp(k) is output
Time
High density
Output
Output
Output
D1
D2
D3
(1)The point is detected due to noise
(4)Point detection is successful
(3) Point detection failed
(1) Segment is not output correctly
(2) Segment is detected correctly
Detected point

(a) The decision method of segmentation points by Equation (1)

(b) Advantage of proposed segmentation method

Fig. 2. Decision method of segmentation points and it's advantage points

3.2 On-Line Clustering of Detected Segments

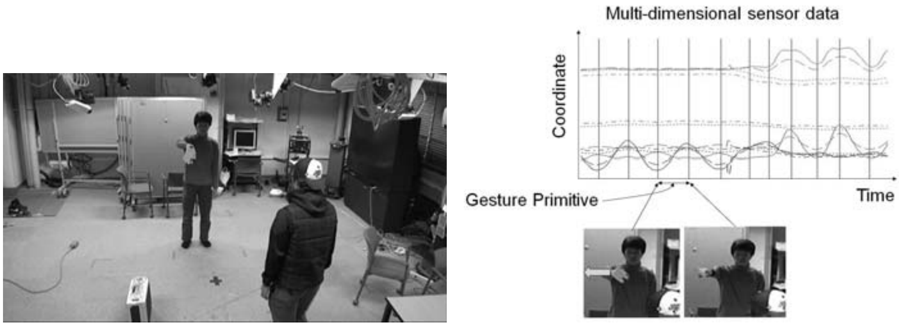
We apply a clustering algorithm of time-series data for grouping the detected segments and labeling.

Necessary reason of On-line clustering. MD-SWAB algorithm is fast segmentation algorithm because of the advantage of on-line segmentation. For keeping of segmentation speed, we need not the batch clustering algorithm but the on-line clustering algorithm. Though calculation of the on-line clustering approach is faster than that of the batch clustering approach, the clustering accuracy of on-line approaches will become worse than that of batch approaches in general.

Therefore we need an on-line clustering approach which has a high accuracy of clustering. Here, the HB-SOINN algorithm is able to perform clustering in an on-line manner, and also have the clustering accuracy of the batch algorithm. Thus we use HB-SOINN for our objective. In the next section, we describe the algorithm of HB-SOINN.

HB-SOINN algorithm. HB-SOINN is a hybrid approach which integrates a self-organizing incremental neural network (SOINN), which is a tool for incremental clustering, and HMM which is used for modeling of time-series data. In HB-SOINN, HMM is used as a pre-processor for SOINN, to map the variable-length patterns into fixed-length patterns. HMM contributes to robust feature extraction from sequence patterns, enabling similar statistical features.

A new observation sequence seg_i is modeled by using HMM λ_i, where seg_i is the segment which is output from MD-SWAB. The HMM is a left-to-right model and has N states which are based on the unit Gaussian probabilities with a diagonal-covariance matrix for each state. The estimated λ_i by Baum Welch algorithm is $\lambda_i = \{((\mu_i(1), \Sigma_i(1)), \ldots ((\mu_i(N), \Sigma_i(N))\}$ μ_i in λ_i is input data ξ_k to SOINN as: $\xi_i = \{\mu_i(1) \ldots \mu_i(N)\}$. Using HMM, the variable length sequence (segments which are detected by MD-SWAB) seg_i is mapped to a vector of fixed dimension ξ_i for input to SOINN.



(a) Appearance of experiment

(b) An example of correct segmentation positions

Fig. 3. Experimental setting

We use adjusted SOINN [11], which only adopts the first layer of SOINN as in [9]. Details of SOINN are described elsewhere [12], [9]. For HB-SOINN, we set parameters λ, a_d and the similarity threshold Th_m.

4 Experiment

For evaluating of segmentation performance of MD-SWAB, we obtain continuous gesture data by using an optical motion capture system. We used the method for comparison with the MD-SWAB method.

4.1 Experimental Setting

At first, we attached the reflection markers to body parts, and the position information (xyz coordinates) of maker was obtained from 8 cameras. The indicator performs the navigation task by using one hand gesture as in Figure 3(a). To project the coordinates of the each maker components into x, y, and z axes, we attach two makers to the palm of the right hand and two makers to the elbow (total 4 makers). Time-series data is obtained from 30 frame/second image sequence, and the length of time-series is 5909. In this experiment, the direction of the performer and his position are fixed.

4.2 Comparative Method

We use Dynamic HMM On-line Segmentation (DHOS) [5] as the comparative method with MD-SWAB. The segmentation algorithm in [5] is based on the assumption that data belonging to the same gesture primitive will have the same underlying probability distribution. An HMM with a dynamically changing number of states is used in this algorithm together with an on-line variant of the Viterbi algorithm [10] which performs an unsupervised segmentation and classification of the data.

When a new data point y_t (multidimensional vector or one dimensional value) arrives at time t, a new vector $x_t = \{y_{t-(m-1)\tau} \cdots, y_t\}$ is embedded, where τ is the delay parameter of the embedding and m is the embedding dimension. Then a probabilistic density function (pdf) $p_t(x)$ is estimated by using x_t and Multivariate Gaussian kernel is used for pdf $p_t(x)$. As a result, time-series data is transformed to a sequence of pdf $p_t(x)$. We can also readily compute the distances between the new pdf and all the previous pdfs for calculating the Viterbi path. The advantage of this algorithm is being able to apply it to multidimensional data.

We need to set 6 parameters for this algorithm. These parameters are m, τ, window size of pdf: W, segmentation threshold: θ, widths of kernel: ς, the parameter for the calculation of distance between pdfs k, respectively. We estimate ς automatically by using maximum likelihood estimation. Thus, we need to set 5 parameters: m, τ, W, θ, k for our experiment. Details of DHOS is described in the original literature [5]. Details of DHOS for segmentation of human-motion (multidimensional sensor data) is also described in the literature [6].

4.3 Evaluation of Proposed System

The start and end points of gesture primitive were defined as correct segmentation positions. We defined one stroke as gesture primitive. We watched the video and wave patterns simultaneously, and detected correct segmentation points by hand. Here, Figure 3(b) shows the example of correct segmentation positions. A result of segmentation by hand, 171 segments were extracted from the data.

4.4 Experimental Result

The parameters of MD-SWAB and HB-SOINN are set as $w_0 = 180, Th = 250000, W = 10, C = 20, \lambda = 100, a_d = 10$ and $Th_m = 10$ and the parameters of DHOS are set as $m = 4, \tau = 1, W = 20, \theta = 0.5$ and $k = 12$ by performing the preliminary experiment using a gesture data set which was not used this experiment.

We compared segmented points (SP) by MD-SWAB and correctly estimated segment points (CSP) of section 4.1. We evaluated MD-SWAB and DHOS by using recall: R, precision: P and F value: F. The following equations denote the calculation of R, P and F.

$$R = \frac{\text{Correct estimated segment points}}{\text{Total correctly segment points by human}} \ (0 \leq R \leq 1)$$

$$P = \frac{\text{Correct estimated segment points}}{\text{Total segment points by method}} (0 \leq P \leq 1)$$

$$F = \frac{2 \times R \times P}{R + P} \quad (0 \leq F \leq 1)$$

Criterion whether SP is CSP was decided such that the difference between segmented primitive and correct segmented frames is smaller than ±10, the begin frame of segmented primitive is $sp(k)$ and the end frame of it is $sp(k+1)$. Table 1 shows R, P, F of MD-SWAB and DHOS. From Table 1, R of DHOS is higher than R of MD-SWAB, however P of MD-SWAB is much higher than P of DHOS. We found that DHOS detected many segmentation points, but these segmentation points included the points which were not the correct segmentation points.

5 Discussion

Firstly, we discuss calculation time and the labeling result by HB-SOINN. Second, we describe future work for the proposed method.

5.1 Calculation Time

As a result of the processing by a PC which had equal performance (CPU is Intel Xeon 3.0GHZ), we compared the segmentation time of MD-SWAB and DHOS. MD-SWAB took 11 seconds for segmentation. In addition, the proposed method that integrated MD-SWAB and HB-SOINN took 121 seconds for segmentation and clustering.

On the other hand, DHOS took 7142 seconds. This is attributed to saving all past time-series data as pdfs. According to [5], the cut-off strategy where we discard old pdfs is efficient for calculation. We performed an additional experiment on DHOS by discarding old time-series data (pdfs). Here, we discarded $r \times t$ frames data, where t was current frame and $r = 0.4, 0.6, 0.8$. For example, when time-series data of 5000 frames is input, we discard the oldest $5000r$ frames data from the buffer. As a result of this experiment, HDOS takes $4755, 2947, 1867$ seconds for calculation in $r = 0.4, 0.6, 0.8$ respectively. However, when we discard $0.8 \times t$ frame data, segmentation accuracy (R, P, F) is worse than that of $r = 0$ (saving all past time-series data).

In summary, the segmentation time of MD-SWAB was clearly faster than that of HDOS. HDOS takes a lot of time for segmentation.

5.2 Segmentation Accuracy

Figure 4 shows some segmentation points of both methods. The lines in Figure 4 show a segmentation point (frame). Right gray lines denote SP of DHOS: dark gray lines denote SP of MD-SWAB: black lines denote CSP. The figure shows the segmentation points of DHOS, many segment points are detected incorrectly. Segment points of MD-SWAB are detected correctly. DHOS decides segmentation points based on the difference between Gaussian distribution. When width parameter σ is a large value, time-series data which has sensitive changing points is smoothed and DHOS can not detect such changing points. When width parameter σ is a small value, time-series data which has sensitive changing points is smoothed and DHOS detect even unnecessary segmentation points as Figure 4. On the other hand, MD-SWAB avoids this problem by using density of sum of error (Figure 2(a)) for the segmentation evaluation.

Table 1. Recall, Precision and F value

	Recall	*Precision*	*F* value
MD-SWAB	0.776	0.731	0.753
DHOS	0.832	0.416	0.555

Therefore total evaluation F of MD-SWAB is much higher than F of DHOS. Experimental results show the segmentation performance of MD-SWAB is clearly superior to that of DHOS.

5.3 Labeling of Segmented Gesture Primitives

In this section, we discuss the accuracy of clustering to segmented gesture primitives and labeling. We mainly investigated whether the same kind of gestures were represented by similar label sequences which were obtained after on-line

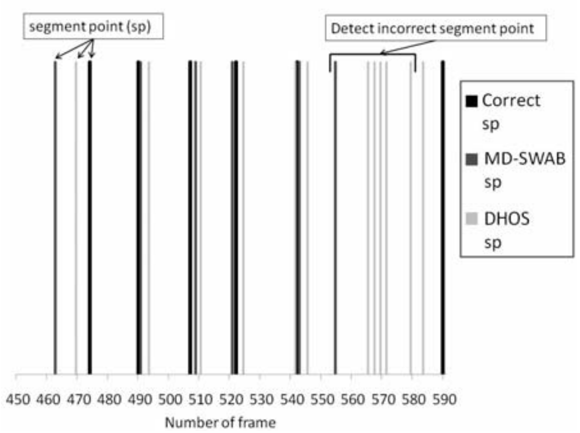


Fig. 4. Part of segmentation points

segmentation and clustering. And we picked up gestures which the indicator used more than 2 times and investigated them. As a result of on-line clustering by HB-SOINN, 28 clusters are output and the number of labels is 28.

Table 2 shows the labeling result after on-line clustering. In Table 2, gestures: "come on (A), come on (B), to right, go there" are arbitrary and include repeats. Then the length of labeling sequence is different depending on the pattern p_x. From Table 2, we found that the labeling result of "come on (B), to right, to left, go there, no motion" is good, and label sequences which are obtained from the same kind of gesture are similar. However, label sequences of "come on (A)" is different between p_1 and p_2, p_3, that of "to left" is different between all patterns, slightly, and that of "roll right" is different between p_1 and p_2.

Table 2. Labeling result

Kind	Num	Label sequences which denote gesture patterns
come on(A)	3	p_1:(2 2 5 2), p_2:(16 17 18 16 17 16 19), p_3:(27 17 16 17 16 17 16)
come on (B)	2	p_1:(7 6 7 8 6 6 7 8 7), p_2:(7 13 8 8 6)
to right	4	p_1:(4 5 4 5 4 5 4), p_2:(11 5 4 5 4), p_3:(12 5 4 5 12 5), p_4:(26 5 4 5 4 5)
to left	3	p_1:(25 16 12 4 22 12 6), p_2:(6 22 12 16), p_3:(27 12 22 16 12)
go there	5	p_1:(3 3 15 3), p_2:(3 15 3 15 3), p_3:(3 3 15 3 15 3) p_4:(3 15 3 15 3 6), p_5:(3 15 3 15 3 15 3)
roll right	2	p_1:(5 18 4 26 12 23), p_2:(4 27 23 17 4 14 9 12 5)
no motion	12	10 pattens are p_1 (1), 2 pattens are p_2 to $p_2$2 (21)

6 Conclusion

This paper proposed the fast on-line unsupervised segmentation method for multidimensional time-series data such as spatiotemporal gesture data. We enhanced the SWAB algorithm for segmentation of single dimension algorithm, and we proposed the our segmentation algorithm MD-SWAB for multidimensional data.

For evaluation of our proposed system, we used hand gesture data from motion capture and compared the unsupervised segmentation accuracy with the DHOS algorithm. We have found that MD-SWAB outperforms the DHOS algorithm in segmentation accuracy and calculation time.

In addition, the experimental result showed that the proposed system that integrated MD-SWAB and HB-SOINN was able to not only segment multidimensional time-series data in an on-line manner, but also cluster and label gesture primitives simultaneously. For detection of gesture pattern automatically, we need to add the current pattern discovery algorithm [1]. a future work is the implementation of incremental pattern discovery and acquisition algorithm.

References

1. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 493–498. ACM, New York (2003)
2. Ge, X., Smyth, P.: Segmental semi-markov models for endpoint detection in plasma etching. IEEE Transactions on Semiconductor Engineering (2001)
3. Ide, T., Inoue, K.: Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In: Proc. SIAM Intl. Conf. Data Mining, pp. 571–575 (2005)
4. Keogh, E., Chu, S., Hart, D., Pazzani, M.: An online algorithm for segmenting time series. IEEE International Conference on Data Mining 0, 289 (2001)
5. Kohlmorgen, J., Lemm, S.: A dynamic hmm for on-line segmentation of sequential data. Advances in Neural Information Processing Systems.
6. Kulic, D., Takano, W., Nakamura, Y.: Combining automated online segmentation and incremental clustering for whole body motions. In: IEEE International Conference on Robotics and Automation, pp. 2591–2598 (2008)
7. Kulić, D., Takano, W., Nakamura, Y.: Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. Int. J. Rob. Res. 27(7), 761–784 (2008)
8. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2–11 (2003)
9. Okada, S., Nishida, T.: Incremental clustering of gesture patterns based on a self organizing incremental neural network. In: IEEE International Joint Conference on Neural Networks (IJCNN 2009) (2009)
10. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. In: Proc. IEEE, pp. 257–286 (1989)
11. Shen, F., Hasegawa, O.: A fast nearest neighbor classifier based on self-organizing incremental neural network. Neural Networks 21(10), 1537–1547 (2008)
12. Shen, F., Hasegawa, O.: An incremental network for on-line unsupervised classification and topology learning. Neural Networks 19(1), 90–106 (2006)

Robust People Segmentation by Static Infrared Surveillance Camera

José Carlos Castillo, Juan Serrano-Cuerda, and Antonio Fernández-Caballero

Universidad de Castilla-La Mancha, Departamento de Sistemas Informáticos
& Instituto de Investigación en Informática de Albacete
Campus Universitario s/n, 02071-Albacete, Spain
caballer@dsi.uclm.es

Abstract. In this paper, a new approach to real-time people segmentation through processing images captured by an infrared camera is introduced. The approach starts detecting human candidate blobs processed through traditional image thresholding techniques. Afterwards, the blobs are refined with the objective of validating the content of each blob. The question to be solved is if each blob contains one single human candidate or more than one. If the blob contains more than one possible human, the blob is divided to fit each new candidate in height and width.

1 Introduction

In the surveillance field [11], [8], [9] the use of infrared cameras are being intensively studied in the last decades. Many algorithms focusing specifically on the thermal domain have been explored. The unifying assumption in most of these methods is the belief that the objects of interest are warmer than their surroundings [13]. Thermal infrared video cameras detect relative differences in the amount of thermal energy emitted/reflected from objects in the scene. As long as the thermal properties of a foreground object are slightly different (higher or lower) from the background radiation, the corresponding region in a thermal image appears at a contrast from the environment.

In [6], [2], a thresholded thermal image forms the first stage of processing after which methods for pose estimation and gait analysis are explored. In [10], a simple intensity threshold is employed and followed by a probabilistic template. A similar approach using Support Vector Machines is reported in [12]. Recently, a new background-subtraction technique to robustly extract foreground objects in thermal video under different environmental conditions has been presented [3]. A recent paper [7] presents a real-time egomotion estimation scheme that is specifically designed for measuring vehicle motion from a monocular infrared image sequence at night time. In the robotics field, a new type of infrared sensor is described [1]. It is suitable for distance estimation and map building. Another application using low-cost infrared sensors for computing the distance to an unknown planar surface and, at the same time, estimating the material of the surface has been described [5].

In this paper, we introduce our approach to real-time robust people segmentation through processing video images captured by an infrared camera.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 348–357, 2010.

2 Robust People Segmentation Algorithm

The proposed human detection algorithm is explained in detail in the following sections related to the different phases, namely, people candidate blobs detection, people candidate blobs refinement and people confirmation.

2.1 People Candidate Blobs Detection

The algorithm starts with the analysis of input image, $I(x, y)$, captured at time t by an infrared camera, as shown in Fig. 1a. Firstly, a change in scale, as shown in equation (1) is performed. The idea is to normalize all images to always work with a similar scale of values, transforming $I(x, y)$ to $I^1(x, y)$ (see Fig. 1b). The normalization assumes a factor γ, calculated as the mean gray level value of the las n input image, and uses the mean gray level value of the current image, $\overline{I}$.

$$I^1(x, y) = \frac{I(x, y) \times \gamma}{\overline{I}} \tag{1}$$

where $I^1(x, y)$ is the normalized image. Notice that $I^1(x, y) = I(x, y)$ when $\overline{I} = \gamma$.

The next step is the elimination of incandescent points - corresponding to light bulbs, fuses, and so on -, which can confuse the algorithm by showing zones with too high temperatures. As the image has been scaled, the threshold θ_i calculated to eliminate these points is related to the normalization factor γ. Indeed,

$$\theta_i = 3 \times \frac{5}{4}\gamma \tag{2}$$

$\delta = \frac{5}{4}\gamma$ introduces a tolerance value of a 25% above the mean image value. And, $3 \times \delta$ provides a value high enough to be considered an incandescent image pixel. Thus, pixels with a higher gray value are discarded and filled up with the mean gray level of the image.

$$I^1(x, y) = \begin{cases} I^1(x, y), & \text{if } I^1(x, y) \leq \theta_i \\ \overline{I^1}, & \text{otherwise} \end{cases} \tag{3}$$

The algorithm uses a threshold to perform a binarization for the aim of isolating the human candidates spots. The threshold θ_c, obtains the image areas containing moderate heat blobs, and, therefore, belonging to human candidates. Thus, warmer zones of the image are isolated where humans could be present. The threshold is calculated as:

$$\theta_c = \frac{5}{4}(\gamma + \sigma_{I^1}) \tag{4}$$

where σ_{I^1} is the standard deviation of image $I^1(x, y)$. Notice, again, that a tolerance value of a 25% above the sum of the mean image gray level value and the image gray level value standard deviation is offered.

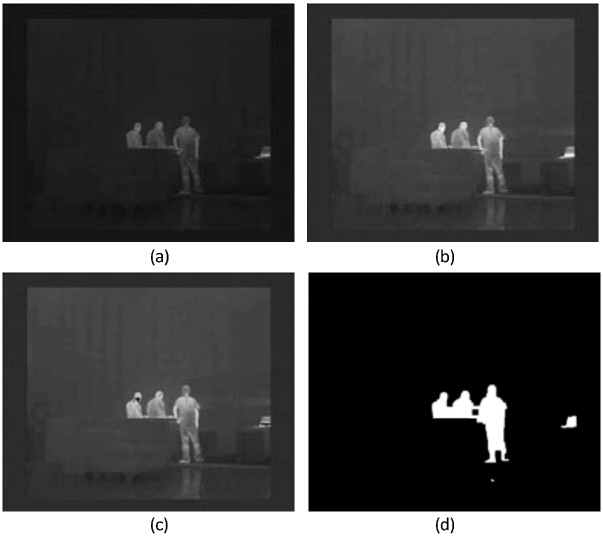


Fig. 1. (a) Input infrared image. (b) Scaled frame. (c) Incandescence elimination. (d) Thresholded frame.

Now, image $I^1(x, y)$ is binarized using the obtained threshold θ_c. Pixels above the threshold are set as maximum value $max = 255$ and pixels below are set as minimum value $min = 0$.

$$I_b^1(x, y) = \begin{cases} min, \text{ if } I^1(x, y) \leq \theta_c \\ max, \text{ otherwise} \end{cases} \tag{5}$$

Next, the algorithm performs morphological opening (equation (6)) and closing (equation (7)) operations to eliminate isolated pixels and to unite areas split during the binarization. These operations require structuring elements that in both cases are 3×3 square matrixes centered at position $(1, 1)$. These operations greatly improve the binarized shapes as shown in Fig. 1.

$$I_o^1(x, y) = I_b^1(x, y) \circ \begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} \tag{6}$$

$$I_c^1(x, y) = I_o^1(x, y) \bullet \begin{vmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix} \tag{7}$$

Afterwards, the blobs contained in the image are obtained. A minimum area, A_{min}, - function of the image size - is established for a blob to be considered to contain humans.

$$A_{min} = 0.0025 \times (r \times c) \tag{8}$$

where r and c are the number of rows and columns, respectively of input image $I(x, y)$. As a result, the list of blobs, L_B, containing people candidates in form of blobs $b_\lambda[(x_{start}, y_{start}), (x_{end}, y_{end})]$, is generated. λ stands for the number of the people candidate blob in image $I^1(x, y)$, whereas (x_{start}, y_{start}) and (x_{end}, y_{end}) are the upper left and lower right coordinates, respectively, of the minimum rectangle containing the blob. As an example, consider the resulting list of blobs related to Fig. 1 and offered in Table 1.

Table 1. People candidates blobs list

λ	x_{start}	y_{start}	x_{end}	y_{end}	area
1	297	270	482	458	35154
2	608	344	645	376	1254

2.2 People Candidate Blobs Refinement

In this part, the algorithm works with the list of blobs L_B, present in image $I^1(x, y)$, obtained at the very beginning of the previous section. At this point, there is a need to validate the content of each blob to find out if it contains one single human candidate or more than one. Therefore, the algorithm processes each detected blob separately.

Let us define a region of interest (ROI) as the minimum rectangle containing one blob of list L_B. A ROI may be defined as $R_\lambda = R_\lambda(i, j)$, when associated to blob $b_\lambda[(x_{start}, y_{start}), (x_{end}, y_{end})]$. Notice that $i \in [1..max_i = x_{end} - x_{start} + 1]$ and $j \in [1..max_j = y_{end} - y_{start} + 1]$.

People vertical delimiting. The first step consists in scanning R_λ by columns, adding the gray level value corresponding to each column pixel, as shown in equation (9).

$$H_\lambda[i] = \sum_{j=1}^{max_j} R_\lambda(i, j), \forall i \in [1..max_i] \tag{9}$$

This way, a histogram $H_\lambda[i]$ showing which zones of the ROI own greater heat concentrations is obtained. A double purpose is pursued when computing the histogram. In first place, we want to increase the certainty of the presence and situation of human heads. Secondly, as a ROI may contain several persons that are close enough to each other, the histogram helps separating human groups (if any) into single humans. This method, when looking for maximums and minimums within the histogram allows differentiating among the humans present in the particular ROI.

Now the histogram, $H_\lambda[i]$, is scanned to separate grouped humans, if any. For this purpose, local maxima and local minima are searched in the histogram to establish the different heat sources (see Fig. 2a). To assess whether a histogram column contains a local maximum or minimum, a couple of thresholds

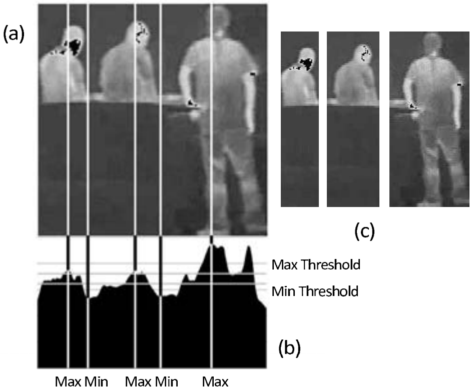


Fig. 2. (a) Input ROI. (b) Histogram. (c) Columns adjustment to obtain three human candidates.

are fixed, $\theta_{v_{max}}$ and $\theta_{v_{min}}$. Experimentally, we went to the conclusion that the best thresholds should be calculated as:

$$\theta_{v_{max}} = 2 \times \overline{R_\lambda} + \sigma_{R_\lambda} \tag{10}$$

$$\theta_{v_{min}} = 0.9 \times \overline{R_\lambda} \times max_j \tag{11}$$

Each different region that surpasses $\theta_{v_{max}}$ is supposed to contain one single human head, as heads are normally warmer than the rest of the people body covered by clothes. That is why $\theta_{v_{max}}$ has been set to the double of the sum of the average gray level plus the standard deviation of the ROI. On the other hand, $\theta_{v_{min}}$ indicates those regions of the ROI where the sum of the heat sources are really low. These regions are supposed to belong to gaps between two humans. We are looking for regions where the column summed gray level is below a 90% of the mean ROI gray level value. Fig. 2b shows the histogram for input ROI of Fig. 2a. You may observe the values for $\theta_{v_{max}}$ and $\theta_{v_{min}}$, corresponding to three peaks (three heads) and two valleys (two separation zones). Fig. 2c shows the three humans as separated by the algorithm into sub-ROIs, $sR_{\lambda,\alpha}$.

People horizontal delimiting. All humans contained in a sub-ROI, $sR_{\lambda,\alpha}$, obtained in the previous section possess the same height, namely the height of the original ROI. Now, we want to fit the height of each sub-ROI to the real height of the human contained. Rows adjustment is performed for each new sub-ROI, $sR_{\lambda,\alpha}$, generated by the previous columns adjustment, by applying a new threshold, θ_h.

The calculation is applied separately on each sub-ROI to avoid the influence of the rest of the image on the result. This threshold takes the value of the sub-ROI average gray level, $\theta_h = \overline{sR_{\lambda,\alpha}}$. Thus, sub-ROI $sR_{b,\lambda,\alpha}$ is binarized in order to delimit its upper and lower limits, obtaining $sR_{\lambda,\alpha}$, as shown in equation (12) similar to equation (5).

$$sR_{b,\lambda,\alpha}(i,j) = \begin{cases} min, \text{ if } sR_{\lambda,\alpha}(i,j) \leq \theta_h \\ max, \text{ otherwise} \end{cases} \tag{12}$$

After this, a closing is performed to unite spots isolated in the binarization, getting $sR_{c,\lambda,\alpha}$ (see Fig. 3b).

$$sR_{c,\lambda,\alpha}(i,j) = sR_{b,\lambda,\alpha}(i,j) \bullet \begin{vmatrix} 0\,1\,0 \\ 1\,1\,1 \\ 0\,1\,0 \end{vmatrix} \tag{13}$$

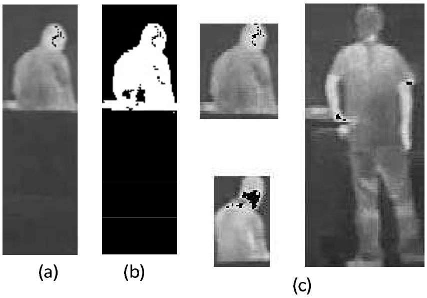


Fig. 3. (a) Input sub-ROI. (b) Binarized sub-ROI. (b) Rows adjustment to delimit three human candidates.

Next, $sR_{c,\lambda,\alpha}$ is scanned, searching pixels with values superior to min. The upper and lower rows of the human are equal to the first and last rows, respectively, containing pixels with a value set to max. The final result, assigned to new ROIs, $\Re_\kappa$, may be observed in Fig. 3c. The blobs associated to the split ROIs are enlisted into the original blobs list, L_B (see Table 2).

Table 2. Refined people candidates blobs list

κ	x_{start}	y_{start}	x_{end}	y_{end}	area
1	339	286	395	354	3808
2	396	270	481	458	15980
3	298	289	338	354	2600

3 Data and Results

We have tested our proposal with a well-known indoor infrared video benchmark, namely the "Indoor Hallway Motion" sequence included in dataset 5 "Terravic Motion IR Database" provided within the OTCBVS Benchmark Dataset Collection [4].

Firstly, in Fig. 4 the results of applying the algorithm to a sequence where a person is crossing the scene form right to left. Fig. 4a shows the scenario. Notice the presence of a hot spot in the image that could lead to confusion, but the algorithm does not make any mistake. Fig. 4b shows the first frame where the person is detected. The detection works fine - a hit of a 100% - through all the frames (see Fig. 4c) until the person reaches the hot spot region. Here (Fig. 4d), the human region also covers the hot spot. From this moment on, and until the person exits the scene, the performance of the person segmentation hit is a 98% (e.g. Fig. 4e and f).

The second example shows another person crossing from right to left, while, at the same time, a second human is slowly approaching the camera from the rear. As you may observe (Fig. 5a to c), the first human is perfectly segmented. The person approaching is wearing clothes that do not emit any heat. Thus, only the head is visible, and it is very difficult to detect such a spot as belonging to a human. Nonetheless, in a 63% of the frames where the partially occluded human is present, he is detected (see Fig. 5d and e). Unfortunately, when this person is too close to the camera, it is not detected as a human (see Fig. 5f).

The third example shows a human entering the scene from the front right and walking to the rear left (see Fig. 6. Now, this person shows an abnormal and uniform high heat level. This is a challenge for our proposal, as we try to locate the human head in relation with the body (in terms of heat and size). The performance of the algorithm in this sequence is about a 60% of hits when the person is close to the camera (see Fig. 6a to c) and grows to a 90% when the human goes far, as shown in Fig. 6e and f. Fig. 6d shows one of the three frames where a false positive is gotten in a total of 140 frames of the sequence.

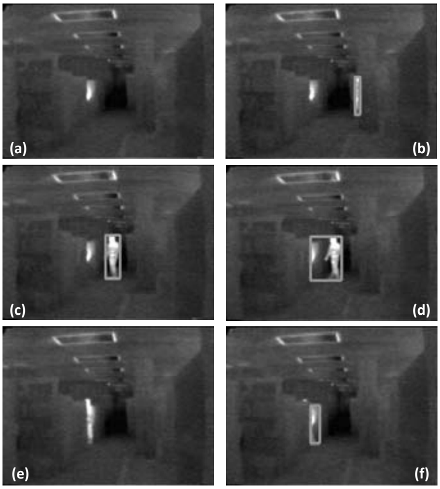


Fig. 4. Crossing from right to left. (a) Frame #220. (b) Frame #250. (c) Frame #290. (d) Frame #292. (e) Frame #324. (f) Frame #333.

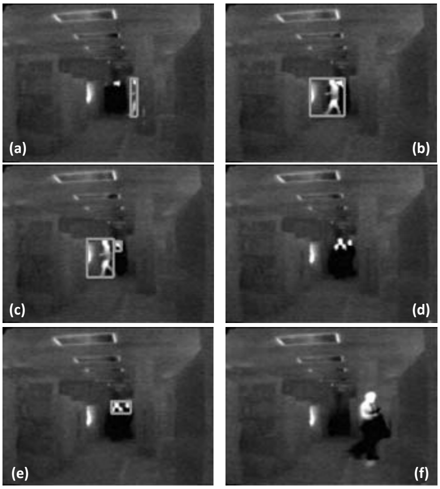


Fig. 5. Crossing from right to left, and approaching from rear to front. (a) Frame #2326. (b) Frame #2357. (c) Frame #2362. (d) Frame #2420. (e) Frame #2411. (f) Frame #2530.

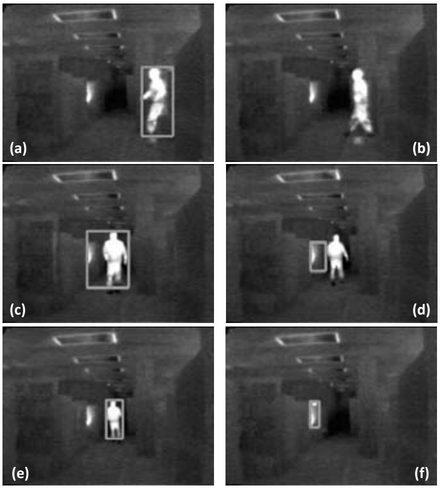


Fig. 6. Walking from front right to rear left. (a) Frame #5135. (b) Frame #5150. (c) Frame #5175. (d) Frame #5209. (e) Frame #5229. (f) Frame #5264.

Finally, we show in Fig. 7 a sequence of two people entering the scene from different sides, and walking together to the rear. As shown in Fig. 7a to c, the segmentation process is efficient. From frame #8344 on, corresponding to Fig. 7d, the humans are not correctly divided, as they are too close to the rear. As shown in Fig. 7e and f, until the humans disappear from the scene, the segmentation of the group is right for a 47%.

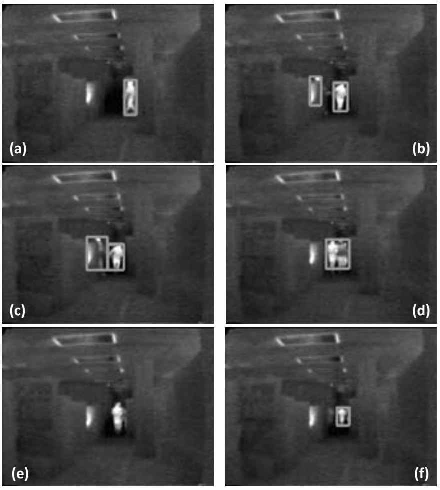


Fig. 7. Two persons walking to the rear. (a) Frame #8268. (b) Frame #8293. (c) Frame #8297. (d) Frame #8344. (e) Frame #8377. (f) Frame #8478.

4 Conclusions

A new approach to real-time people segmentation through processing images captured by an infrared camera has been extensively described. The proposed algorithm starts detecting human candidate blobs processed through traditional image thresholding techniques. Afterwards, the blobs are refined with the objective of solving the question if each blob contains one single human candidate or has to be divided into smaller blobs. If the blob contains more than one possible human, the blob is divided to fit each new candidate in height and width. The results obtained so far in indoor scenarios are promising. We have been able of testing our person segmentation algorithms on a well-known test bed.

Acknowledgements

This work was partially supported by the Spanish Ministerio de Ciencia e Innovación under project TIN2007-67586-C02-02, and by the Spanish Junta de Comunidades de Castilla-La Mancha under projects PII2I09-0069-0994 and PEII09-0054-9581.

References

1. Benet, G., Blanes, F., Simó, J.E., Pérez, P.: Using infrared sensors for distance measurement in mobile robots. Robotics and Autonomous Systems 40(4), 255–266 (2002)
2. Bhanu, B., Han, J.: Kinematic-based human motion analysis in infrared sequences. In: Proceedings of the Sixth IEEE Workshop on Applications of Computer Vision, pp. 208–212 (2002)

3. Davis, J.W., Sharma, V.: Background-subtraction in thermal imagery using contour saliency. International Journal of Computer Vision 71(2), 161–181 (2007)
4. Davis, J.W., Keck, M.A.: A two-stage template approach to person detection in thermal imagery. In: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision, vol. 1, pp. 364–369 (2005)
5. Garcia, M.A., Solanas, A.: Estimation of distance to planar surfaces and type of material with infrared sensors. In: Proceedings of the 17th International Conference on Pattern Recognition, vol. 1, pp. 745–748 (2004)
6. Iwasawa, S., Ebihara, K., Ohya, J., Morishima, S.: Realtime estimation of human body posture from monocular thermal images. In: Proceedings of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 15–20 (1997)
7. Jung, S.-H., Eledath, J., Johansson, S., Mathevon, V.: Egomotion estimation in monocular infra-red image sequence for night vision applications. In: IEEE Workshop on Applications of Computer Vision, p. 8 (2007)
8. López, M.T., Fernández-Caballero, A., Fernández, M.A., Mira, J., Delgado, A.E.: Visual surveillance by dynamic visual attention method. Pattern Recognition 39(11), 2194–2211 (2006)
9. López, M.T., Fernández-Caballero, A., Fernández, M.A., Mira, J., Delgado, A.E.: Motion features to enhance scene segmentation in active visual attention. Pattern Recognition Letters 27(5), 469–478 (2006)
10. Nanda, H., Davis, L.: Probabilistic template based pedestrian detection in infrared videos. In: Proceedings of the IEEE Intelligent Vehicle Symposium, vol. 1, pp. 15–20 (2002)
11. Pavón, J., Gómez-Sanz, J., Fernández-Caballero, A., Valencia-Jiménez, J.J.: Development of intelligent multi-sensor surveillance systems with agents. Robotics and Autonomous Systems 55(12), 892–903 (2007)
12. Xu, F., Liu, X., Fujimura, K.: Pedestrian detection and tracking with night vision. IEEE Transactions on Intelligent Transportation Systems 6(1), 63–71 (2005)
13. Yilmaz, A., Shafique, K., Shah, M.: Target tracking in airborne forward looking infrared imagery. Image and Vision Computing 21(7), 623–635 (2003)

Building Digital Ink Recognizers Using Data Mining: Distinguishing between Text and Shapes in Hand Drawn Diagrams

Rachel Blagojevic[1], Beryl Plimmer[1], John Grundy[2], and Yong Wang[1]

[1] University of Auckland, Private bag 92019, Auckland, New Zealand
[2] Swinburne University of Technology, PO Box 218, Hawthorn, Victoria, Australia 3122
rpat088@aucklanduni.ac.nz, jgrundy@swin.edu.au,
{beryl@cs.,yongwang@}auckland.ac.nz

Abstract. The low accuracy rates of text-shape dividers for digital ink diagrams are hindering their use in real world applications. While recognition of handwriting is well advanced and there have been many recognition approaches proposed for hand drawn sketches, there has been less attention on the division of text and drawing. The choice of features and algorithms is critical to the success of the recognition, yet heuristics currently form the basis of selection. We propose the use of data mining techniques to automate the process of building text-shape recognizers. This systematic approach identifies the algorithms best suited to the specific problem and generates the trained recognizer. We have generated dividers using data mining and training with diagrams from three domains. The evaluation of our new recognizer on realistic diagrams from two different domains, against two other recognizers shows it to be more successful at dividing shapes and text with 95.2% of strokes correctly classified compared with 86.9% and 83.3% for the two others.

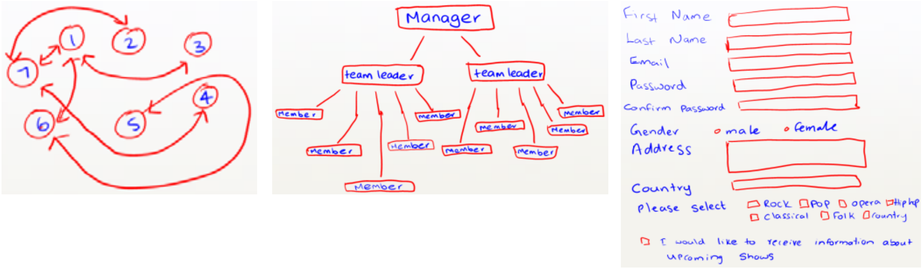
Keywords: Sketch tools, recognition algorithms, sketch recognition, pen-based interfaces.

1 Introduction

Hand drawn pen and paper sketches are commonplace for capturing early phase designs and diagrams. Pen and paper offers an unconstrained space suitable for quick construction and allow for ambiguity. With recent advances in hardware such as Tablet PC's, computer based sketch tools offer a similar pen-based interaction experience. In addition, these computer based tools can benefit from the ease of digital storage, transmission and archiving. Recognition of sketches can add even greater value to these tools. The ability to automatically identify elements in a sketch allows us to support tasks such as intelligent editing, execution, conversion and animation of the sketches.

A number of sketch tools have been developed, however they are yet to achieve general acceptance. One of the outstanding challenges is considerably more accurate recognition. Recognition rates from laboratory experiments are typically in the range

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 358–367, 2010.



a) Directed graph b) Organization chart c) User interface

Fig. 1. Example sketched diagrams for training set

of 98% to 99% and above [1-3]. However, rates achieved in less controlled conditions, where data is not limited to produce optimal performance, are usually much lower, for example accuracy rates between 84% and 93% are reported in [3-6]. Furthermore, many of these tools are limited as they are not able to distinguish between drawing elements (shapes) and text strokes in a sketch [1-3]. Most natural diagrams consist of both writing and drawing as shown in figure 1.

While recognition of handwriting is well advanced and there have been many recognition approaches proposed for hand drawn sketches, there has been less attention on the division of text and drawing. People can comprehend writing and drawing seamlessly, yet there is a clear semantic divide that suggests, from a computational perspective, it is sensible to deal with them separately. Several recognizers [7-9], commonly referred to as dividers, have been proposed for this purpose, but recognition rates in realistic situations are still unacceptable. Limited investigation of machine learning for text-shape division has been found to be effective [8]. This work extends [8] by drawing on a larger set of ink features and using a range of data mining techniques systematically selected and tuned.

2 Background

Two particular applications of dividers are freehand note-taking and hand drawn diagrams. The research on sketched diagram recognition includes dividers but has also addressed recognition of basic shapes and spatial relationships between diagram components. This project has drawn on the work from both applications of dividers.

The majority of recognizers rely on information provided by various measurements of the digital ink strokes (digital ink is represented as a vector of x, y points, each point has a time and possible pressure attribute) [1, 2, 10], as well as specific algorithms to combine and select the appropriate features.

In the area of sketched diagram recognition many systems focus only on shapes [1-3]. There have been some attempts at incorporating text-shape division in domain specific recognizers [11, 12] and domain independent diagramming tools [10, 13]. These systems are predominantly rule-based, using stroke features chosen heuristically to distinguish between text and shapes.

Research in the area of digital ink document analysis for freehand note-taking has explored text-shape division [14-19]. However as the content of documents is mainly text these methods hold some bias which make them unsuitable for sketched diagrams. In addition, as Bhat and Hammond [7] point out, some of these methods would have difficulty with text interspersed within a diagram. There has also been some work separating Japanese characters from shapes in documents [18, 20].

Three reports specifically on dividers are [7-9]. Bishop et al [8] use local stroke features and spatial and temporal context within an HMM to distinguish between text and shape strokes. They found that using local features and temporal context was successful. They report classification rates from 86.4% to 97.0% for three classifier model variations.

In our previous work [9] we developed a domain independent divider for shapes and text based on statistical analysis of 46 stroke features. A decision tree was built identifying eight features as significant for distinguishing between shapes and text. The results on a test set showed an accuracy of 78.6% for text and 57.9% for shapes. Part of the test set was composed of musical notes which had a significant effect on this low classification rate. However, when evaluated against the Microsoft and InkKit dividers, it was able to correctly classify more strokes overall for the test set.

A more recent development in this field is the use of a feature called entropy [7] to distinguish between shapes and text. First strokes are grouped into shapes and words/letters and then stroke points are re-sampled for smoothing. The angle between every point and its adjacent points in the stroke group is calculated. Each angle from the stroke group is matched to a dictionary containing a different alphabet symbol to represent a range of angles. This results in a text string representation of each stroke group. Using Shannon's entropy formula (as cited by Bhat et al [7]) they sum up the probabilities of each letter in the string to find the entropy of that group. This value of entropy is higher for text than shapes as text is more "information dense" than shapes. They report that 92.06% of data which it had training examples for were correctly classified. For data the divider had not been trained on it had an accuracy of 96.42%, however only 71.06% of data was able to be classified. We have re-implemented this algorithm for our evaluation. As our evaluation will show, this divider has been trained and tested on limited data and constrained conditions and does not perform at the reported rate of 92.06% on realistic diagrams.

The choice of features and algorithms is critical to the success of the recognition, yet heuristics currently form the basis of selection. Given that features provide such value as input to recognition algorithms, a feature set should be chosen carefully using statistical or data mining techniques. While others have used some data mining techniques [8, 15] to the best of our knowledge no one has done a comprehensive analysis of algorithms. We present below a comprehensive comparative study of features and algorithms to select the most accurate model. In particular we are looking at the problem of distinguishing between text and shapes as a first step to recognizing sketched diagrams; a fundamental problem required to preserve a non-modal user interface similar to pen and paper.

3 Our Approach

In order to use data mining techniques to build classifiers we first compiled a comprehensive feature library which is used in conjunction with our training set of diagrams to generate a training dataset. We investigated a wide range of data mining algorithms before focusing on seven that were producing the most promising results. These seven algorithms and the training dataset were used to build new dividers.

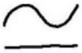
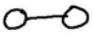
3.1 Features

Our previous feature set [9] of 46 features has been extended to a more comprehensive library of 115 stroke features for sketch recognition. It has been assembled from previous work in sketch recognition, includes some of our own additions, Entropy [7], and our previous divider [9]. Our previous divider is used for several features: pre-classification of the current stroke, pre-classification of strokes close by (for spatial context), and pre-classification of successive strokes (for temporal context).

Many researchers have developed features that measure similar attributes. In order to give the reader some sense of the types of features we have categorized the feature library into ten categories, summarized in table 1.

This feature library is available with full implementation within DataManager [21] from www.cs.auckland.ac.nz/research/hci/downloads.

Table 1. Summary of stroke feature categories

1. Curvature (e.g. the line above has a greater curvature than the line below). 	**6. Pressure** (measure the pressure applied to the screen when drawing a stroke. Pressure is dependent on the capabilities of the hardware).
2. Density (e.g. the text has larger density of points than the shape).	**7. Size**
3. Direction (this is related to the slope of the stroke).	**8. Spatial context** (with sub categories: curvature, density, divider results, intersections, location and size).
4. Divider Results (these features provide the results of text/shape divider algorithms).	**9. Temporal context** (with sub categories: curvature, density, divider results, length, location/distance and time/speed).
5. Intersections (e.g. the diagram shows intersecting strokes). 	**10. Time / speed** (includes total, average, maximum and minimum times or speed).

3.2 Dataset

For the training set we have collected and labeled sketched diagrams from 20 participants using DataManager [21]. Each participant has drawn three diagrams; a directed graph, organization chart and a user interface e.g. figure 1. There are a total of 7248 strokes in the training set, with 5616 text strokes and 1632 shape strokes.

Using this collection of diagrams we have generated a dataset of feature vectors for each stroke using DataManager. DataManager's dataset generator function is able to take the diagrams collected and calculate feature vectors based on the implementation of our feature library.

3.3 Building Classifiers

Weka (developer version 3.7) [22], an open source data mining tool, has a large number of machine learning algorithms that can be used to perform our data analysis and build, tune and test classifier models for dividers. We found that 60 of the algorithms within Weka were possibly suited to the divider problem. We began our analysis with a preliminary investigation of all these algorithms. This involved building classifier models for each algorithm using the training data. Some clearly performed better than others while some needed tuning of their specific parameters to optimize their results. Upon discussion of the preliminary investigation we were able to narrow the search down to seven algorithms that are likely to gain the best classification accuracy for a divider[1].

The chosen classifiers are: Bagging [23](with an REP tree base learner), LADTree [24](alternating decision tree using the LogitBoost strategy), LMT [25](logistic model tree), LogitBoost [26](additive logistic regression with Decision Stump or REP tree base learner), MultilayerPerceptron [22] (neural network), RandomForest [27](forest of random trees) and SMO [28](support vector machine). Using the training dataset of feature vectors generated from the diagrams collected we built dividers by training each classifier. While Weka provides sensible default parameters for most algorithms, some classifiers required tuning to optimize their results.

For Bagging [23] we tuned the algorithm by varying the number of bagging iterations that the algorithm performs. This parameter is indicative of the number of trees that can be produced. The default value for this in Weka is 10 iterations. We ran an experiment using 10-fold cross validation for Bagging with REPTree (a fast decision tree learner) at 10, 100, 500, 1000 and 5000 iterations. Paired t-tests (α=0.05) showed no significant difference in the results at each level of iterations. The highest result is shown in table 2, this was produced at 500 and 1000 iterations.

To tune the LADTree [24] we varied the number of iterations of the algorithm to 10, 100, 500 and 1000 iterations. We were unable to increase the number of iterations to greater than 1000 due to time and memory constraints. This algorithm takes a long time to train therefore we chose to run the experiment with 5-fold cross validation as opposed to 10-fold. Paired t-tests (α=0.05) showed that the LADTree with 500 and 1000 iterations were significantly more accurate than the others. There was no significant difference between the LADTrees with 500 and 1000 iterations. The highest result shown in table 2 was produced at 1000 iterations.

The default parameters in Weka for LMT [25] are sensible for this problem and did not require tuning. The result of 10-fold cross validation using our training dataset on LMT with default parameters is shown in table 2.

To begin tuning LogitBoost [26] we ran a preliminary 10-fold cross validation experiment to see if there were any significant differences in using Decision Stump or REPTree as a base classifier for LogitBoost. A paired t-test (α=0.05) showed no significant difference between the two at 120 iterations of the algorithm. Based on these results we decided to continue to investigate both trees as base classifiers.

To further tune LogitBoost we varied the number of iterations the algorithm performs and also the shrinkage parameter. Shrinkage is a parameter that can be tuned to avoid overfitting the LogitBoost model to the training dataset. When a classifier is overfitted it reduces the likelihood of the model retaining the same level of accuracy, achieved with training data, on a new test dataset. Small values for shrinkage reduce overfitting. We ran experiments using 10-fold cross validation for LogitBoost with

[1] Thanks to Eibe Frank for his advice on the selection of algorithms.

the following options: base classifier as a Decision Stump or REPTree; number of iterations at 10, 100, 500, 1000 or 5000; shrinkage at 1.0 (Weka default value) or 0.1.

Using all combinations of the above options resulted in 20 models for LogitBoost, 10 for each base classifier. For LogitBoost, the model that had the highest level of accuracy was with a Decision Stump base classifier, 5000 iterations and a shrinkage value of 0.1, the result for this model is shown in table 2. Paired t-tests (α=0.05) showed that it was significantly better than all other Decision Stump models except two that were not significantly different; they had a shrinkage value of 1.0 and number of iterations set at 1000 and 5000. When compared with the REPTree models, it was only significantly better than the REPTree model with 10 iterations at a shrinkage value of 0.1, for all others there was no significant difference.

The default parameters in Weka for MultilayerPerceptron [22] are sensible for this problem and did not require tuning. The result of 10-fold cross validation using the training dataset on MultilayerPerceptron with default parameters is shown in table 2.

To tune RandomForest [27] we varied the number of iterations of the algorithm to 10, 100, 500 and 1000 iterations. We were unable to increase the number of iterations to greater than 1000 due to memory constraints. Paired t-tests (α=0.05) showed no significant difference between any of the models. The highest result shown in table 2 was produced at 500 iterations.

Table 2. Best results obtained from selected classifiers

Classifier	% Correctly classified (10-fold cross validation)
LADTree	**97.49*** (5-fold)
LogitBoost	**96.70***
RandomForest	96.45
SMO	96.41
Bagging	95.67
MultilayerPerceptron	95.02
LMT	94.85

SMO [28] is a more complicated classifier to tune. There are two parameters that can be tuned; the complexity value of SMO and the gamma value of the RBF kernel used by SMO. To find the best model we used the GridSearch function in Weka which allows you to optimize two parameters of an algorithm by setting a maximum, minimum, base value and step value for how much a parameter can increase by for each test. One of the main advantages of GridSearch is that the parameters of interest do not have to be first level parameters, for example gamma is not a first level parameter as it is a value used by the RBF kernel, where the RBF kernel is a parameter of SMO. We found the optimal value for complexity was 100, with a gamma value of 0.1. The results of 10-fold cross validation on SMO for our training data is shown in table 2.

The best results of 10-fold cross validation (except LADTree which was 5-fold) for each classifier on our training set is shown in table 2. Paired t-tests (α=0.05) show that LogitBoost and LADTree are significantly better than the other classifiers. There is no significant difference between LogitBoost and LADTree. This is not surprising as LADTree uses the LogitBoost strategy.

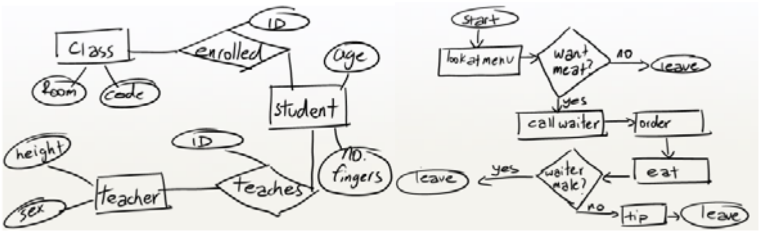
3.4 Implementation

In order to run a comparative evaluation of our two new models against other dividers we integrated our models into DataManager's Evaluator [6]. We also integrated our old divider [9] and implemented the Entropy divider [7].

The Entropy divider had to be trained as no thresholds were provided by [7]. We trained it on the same data as our new dividers using 10-fold cross validation with the decision stump algorithm from Weka [22] to find an optimal threshold. We chose the decision stump algorithm as this generates a decision tree with one node, essentially producing one decision based on the Entropy feature. The 10-fold cross validation reported that 85.76% of the training data was correctly classified; other algorithms such as OneR, a rule based method, and a J48 tree (C4.5 decision tree) showed similar results. Our divider developed from previous work [9] was not re-trained; it was implemented with the same thresholds as the original decision tree.

4 Evaluation

In order to test the accuracy of these dividers on data that they are not trained on we used a new set of diagrams from different domains to the training set. The test set was composed of ER and process diagrams (see figure 2) collected from 33 participants who drew one diagram from each domain. The participants were asked to construct the diagrams from text descriptions so that they are realistic in individual drawing. There are a total of 7062 strokes in our test set which is similar in size to our training set. There are 4817 text strokes and 2245 shape strokes. Table 3 shows the results for each divider on



a)ER diagram

b) Process diagram

Fig. 2. Example sketched diagrams for test set

Table 3. % Correctly classified for each divider

Divider	% Correct	% Text	% Shapes
LADTree	95.2	98.3	88.5
LogitBoost	95.0	98.1	88.4
Old Divider	86.9	93.1	73.5
Entropy	83.3	98.7	50.5

the test set of diagrams. LADTree is the most accurate of the four tested with 95.2% correctly classified closely followed by LogitBoost at 95.0%. The Entropy divider is the least accurate at a rate of 83.3%. It is clear that entropy has a large bias towards text as only 50.5% of the shapes in the test set are correctly classified. Our previous divider is slightly more accurate than Entropy; however its bias towards text is not as extreme. In fact the results show that all dividers classify text much more accurately than shapes.

5 Discussion

The high accuracy of the results we have obtained by using data mining techniques to build dividers demonstrates the effectiveness of this approach. We believe that other recognition problems would also benefit from a similar study of data mining techniques. However there is still room for improvement in these divider algorithms.

In terms of tuning, for all the algorithms where we varied the number of iterations we found that a high number of iterations usually resulted in significantly better results. We could tune these further by increasing the number of iterations for some algorithms however we are constrained by time and memory. Although, these constraints are for training, once the classifier is trained the memory requirements are minimal and actual classification time on instances is very fast in all cases.

We can also study the common types of failures that occur with recognition, in particular for shapes as they are the main source of misclassification. Data mining these misclassified strokes could identify features that may help correctly distinguish them. Studying error cases may also lead to the identification of new features that account for these misclassified shapes.

Feature selection strategies may also contribute to recognizer improvement. This involves using feature selection algorithms to isolate the features that perform well. When training an algorithm insignificant features can have a negative effect on the success of classification algorithms [22] therefore careful feature selection is a very important step to developing recognition techniques. We were surprised that 100+ features were employed by our top two dividers and speculate that some features are redundant or detrimental. Redundant features will only slow execution time whereas our concerns are with features that have a negative effect. Further exploration of feature selection strategies could identify features that should be excluded.

Combining different classifiers into a voting system is also worthy of investigation. Classifiers predictions can be weighted according to their performance and combined to produce one overall classification for an instance [22]. We are yet to investigate whether the different algorithms have a large number of common failures. If they all fail on the same cases then voting is not useful. For future work we plan to investigate the main cause of failures that occur for the original seven algorithms and identify what proportions are common between them.

We chose to train and test on diagrams of different domains to create a general diagram divider. Each diagram domain has its own syntax, semantics and mix of drawing shapes. Given the difference between the training 10-fold validation values and the test results (~ 2.3%), it may be worthwhile to data mine and train a divider for each diagram domain.

6 Conclusion

We have built seven new dividers using data mining techniques to distinguish between text and shapes in hand drawn diagrams. The two best dividers, LADTree and LogitBoost, are able to correctly classify 95.2% and 95.0% respectively of a test set that they have received no training for. A comparative evaluation of these dividers against two others shows that the new dividers clearly outperform the others. The success of our new dividers demonstrates the effectiveness of using data mining techniques for sketch recognition development.

Acknowledgements

Thanks to Associate Professor Eibe Frank for expert advice on using data mining techniques. This research is partly funded by Microsoft Research Asia and Royal Society of New Zealand, Marsden Fund.

References

1. Rubine, D.H.: Specifying gestures by example. In: Proceedings of Siggraph '91. ACM, New York (1991)
2. Paulson, B., Hammond, T.: PaleoSketch: Accurate Primitive Sketch Recognition and Beautification. In: Intelligent User Interfaces (IUI '08). ACM Press, New York (2008)
3. Wobbrock, J.O., Wilson, A.D., Li, Y.: Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. In: User interface software and technology. ACM, Newport (2007)
4. Plimmer, B.: Using Shared Displays to Support Group Designs; A Study of the Use of Informal User Interface Designs when Learning to Program. Computer Science (2004)
5. Young, M.: InkKit: The Back End of the Generic Design Transformation Tool. Computer Science (2005)
6. Schmieder, P., Plimmer, B., Blagojevic, R.: Automatic Evaluation of Sketch Recognition. In: Sketch Based Interfaces and Modelling, New Orleans, USA (2009)
7. Bhat, A., Hammond, T.: Using Entropy to Distinguish Shape Versus Text in Hand-Drawn Diagrams. In: International Joint Conference on Artificial Intelligence (IJCAI '09), Pasadena, California, USA (2009)
8. Bishop, C.M., Svensen, M., Hinton, G.E.: Distinguishing Text from Graphics in On-Line Handwritten Ink. In: Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition. IEEE Computer Society, Los Alamitos (2004)
9. Patel, R., Plimmer, B., et al.: Ink Features for Diagram Recognition. In: 4th Eurographics Workshop on Sketch-Based Interfaces and Modeling 2007. Eurographics, Riverside (2007)
10. Plimmer, B., Freeman, I.: A Toolkit Approach to Sketched Diagram Recognition. In: HCI 2007. eWiC, Lancaster (2007)
11. Lank, E., Thorley, J.S., Chen, S.J.-S.: An interactive system for recognizing hand drawn UML diagrams. In: Proceedings of the Centre for Advanced Studies on Collaborative research. IBM Press, Mississauga (2000)
12. Hammond, T., Davis, R.: Tahuti: A Geometrical Sketch Recognition System for UML Class Diagrams. In: 2002 AAAI Spring Symposium on Sketch Understanding (2002)

13. Zeleznik, R.C., Bragdon, A., et al.: Lineogrammer: creating diagrams by drawing. In: Proceedings of User interface software and technology. ACM, Monterey (2008)
14. Shilman, M., Viola, P.: Spatial recognition and grouping of text and graphics. In: EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2004)
15. Shilman, M., Wei, Z., et al.: Discerning structure from freeform handwritten notes. In: Document Analysis and Recognition (2003)
16. Jain, A.K., Namboodiri, A.M., Subrahmonia, J.: Structure in On-line Documents. In: Proceedings of the Sixth International Conference on Document Analysis and Recognition. IEEE Computer Society, Los Alamitos (2001)
17. Ao, X., Li, J., et al.: Structuralizing digital ink for efficient selection. In: Proceedings of the 11th international conference on Intelligent user interfaces. ACM, Sydney (2006)
18. Machii, K., Fukushima, H., Nakagawa, M.: Online text/drawings segmentation of handwritten patterns. In: Document Analysis and Recognition, Tsukuba Science City, Japan (1993)
19. Microsoft Corporation, Ink Analysis Overview (cited 2008), `http://msdn.microsoft.com/en-us/library/ms704040VS85aspx`
20. Mochida, K., Nakagawa, M.: Separating drawings, formula and text from free handwriting. In: International Graphonomics Society (IGS 2003), Scottsdale, Arizona (2003)
21. Blagojevic, R., Plimmer, B., et al.: A Data Collection Tool for Sketched Diagrams. In: Sketch Based Interfaces and Modeling. Eurographics, Annecy (2008)
22. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
23. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
24. Holmes, G., Pfahringer, B., et al.: Multiclass alternating decision trees. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) ECML 2002. LNCS (LNAI), vol. 2430, pp. 161–172. Springer, Heidelberg (2002)
25. Landwehr, N., Hall, M., Frank, E.: Logistic Model Trees. Machine Learning 95(1-2), 161–205 (2005)
26. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: a Statistical View of Boosting. Stanford University (1998)
27. Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)
28. Platt, J.: Machines using Sequential Minimal Optimization. In: Advances in Kernel Methods - Support Vector Learning (1998)

Relief Patterned-Tile Classification for Automatic Tessella Assembly

José Miguel Sanchiz, Jorge Badenas, and Francisco José Forcada

Universidad Jaume I, Castellón, Spain
{sanchiz,badenas,fforcada@uji.es}

Abstract. This paper presents the detection and classification part of an industrial machine for automated assembly of decorative tessellae over patterned tiles that have significant reliefs. The machine consists of two vision systems for detecting the tiles and the tessellae, and a robotic manipulator to make the assembly. One of the vision systems detects the tessellae and the other one detects and classifies the tiles, finding the positions and orientations on the tiles where tessellae have to be mounted. Lateral illumination is used to enhance shadows and characterize the relief pattern. The shadow pattern is used as the main feature for classifying a tile to a model. The method works with a variety of models, is quite robust, and achieves very good classification results, as required for an industrial application.

1 Introduction

This paper presents the application of computer vision techniques to the resolution of a real problem of assembling decorative crystal tessellae over ceramic tiles. The method presented has been tested on an industrial machine that consists on two conveyor belts, two computer vision systems, and a robotic manipulator. The manipulator takes tessellae that circulate on a conveyor belt and glues them onto ceramic tiles circulating on the other conveyor belt, at convenient positions and orientations. There are several models of tiles, and every model has different assembly configurations that the system has to recognize.

One vision system recognizes and tracks tessellae of different colors, sizes and shapes. The other detects and classifies the ceramic tiles, tracks them and finds the assembly positions and orientations. Assembly and tessellae positions are communicated to the manipulator when they fall in its work area.

A large amount of industrial activities have benefited from the application of computer vision technology on many manufacturing processes. Computer vision can be used to improve productivity and quality management [1]. One of the benefited activities has been robotic assembly. In the literature, different vision guided robot systems for assembling can be found [2] [3]. The application we present combines vision-guided assembly and vision-based classification of objects moving onto conveyor belts [4].

The classification method developed for this application is mainly based on shadow patterns. Shadow properties are of great value in deducing 3D structure.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 368–377, 2010.

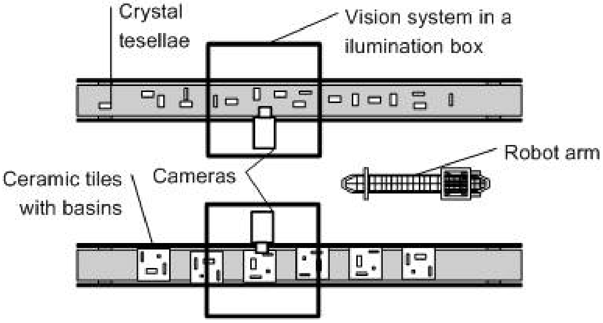


Fig. 1. System representation: a conveyor belt transports small crystal tessellae, and another belt carries ceramic tiles with basins, a robotic manipulator picks up the tessellae and glues them into the basins of the tile

Shape-from-shading [5] is a well-known technique for recovering 3D shape of an object from one or more views [6]. As an example, in [7], shadows in folds are studied for generating drapery and cloth models.

The rest of this paper is organized as follows. Section 2 presents the main elements of the system. Section 3 describes the feature extraction process. Segmentation and morphological filtering techniques are used to extract features that are used for classifying the ceramic tiles. This classification is presented in Section 4. Details on the implementation of the presented methods on an industrial machine are discussed in Section 5. Results are discussed in Section 6, and conclusions are made in Section 7.

2 System Overview

As mentioned, the complete system (an industrial machine) consists of two conveyor belts, one of them transports crystal tessellae, and the other ceramic tiles that have some basins where the tessellae have to be glued. A robotic manipulator is placed between both belts, it picks up tessellae from the first belt and glues them into the appropriate basin on the second belt, with the right orientation. A representation of the system is shown in Figure 1, and a photograph is Figure 2. Subfigure 3.a shows several tessellae that can be used with different tile models. Subfigure 3.b presents a sample tile. In this subfigure white arrows indicate the positions where tesellae have to be assembled. It can be observed that it is difficult to locate these positions. Figure 4 shows the tile of Subfigure 3.b after the robot has assembled tesellae.

The crystal tessellae are of several colors, sizes and shapes. The purpose of the first computer vision system is to identify them and track them until they are in the work area of the manipulator. So that a tessella of a certain type is available for every basin of the tiles.

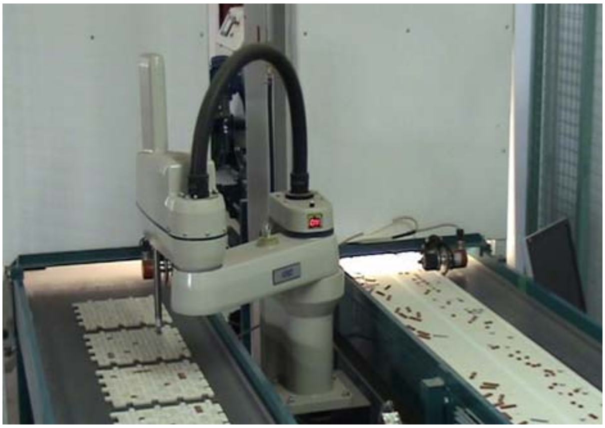


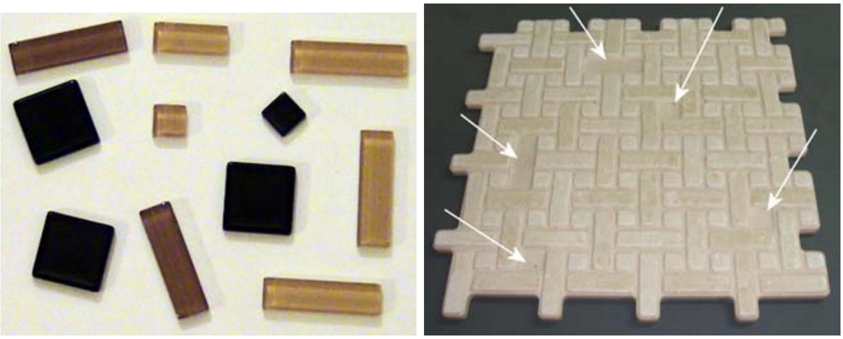
Fig. 2. A photograph of the system. The robot arm picks up tesellae from the right belt and assemble them onto a ceramic tile.

The second computer vision system has the purpose of finding all the basins in the ceramic tiles where the tessellae have to be placed. Finding a basin implies finding its central position, its orientation, and the type of tessella that assembles on it. Also the basins are tracked until they fall in the work area of the manipulator.

This paper deals mainly on the methods developed for the computer vision system that performs tile classification, which is a more complicated problem that the one that is coped with by the vision system dedicated to detecting the tessellae. For this, tiles are located and classified to the right model. The model provides the positions, orientations, and tessella type for each basin of the tile. All possible types of tiles that can be presented together to the machine in a work session form a model set. Several model sets have been prepared preprocessing images of a sample tile of every type, and precomputing features adequate for classification, such as area, shape, and the shadow pattern.

The use of shadows as the main feature is justified in this application due to the way these tiles are manufactured. All the tiles of a model set are manufactured together in the same press. The press consists of several relief patterns (four, six, eight), and manufactures a number of tiles at the same time, one with each relief pattern, and different basins for tessella assembly. Also different ground textures, tones and colors may be used in manufacturing, so textures of two tiles having the same pattern may appear quite different in the images, still the relief pattern is the same, and so are the basins.

We have found that the relief pattern can be efficiently characterized by the shadows produced with lateral illumination. We have developed a method to compute it and use it for robust classification. The shadow pattern is by far the most stable feature, independent of tile textures produced by using different materials in manufacturing. For this reason, cross-correlation of tile images and



(a) Some tesellae.

(b) A sample tile. White arrows indicate tesellae positions.

Fig. 3. Some tesellae of different colours, and shapes and a sample tile

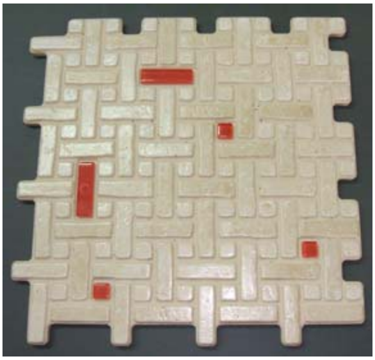


Fig. 4. The tile of Subfigure 3.b with assembled tesellae

model tile images does not work in this problem, while the shadow pattern is texture independent.

3 Feature Extraction

The objective of feature extraction is to be able to robustly classify a tile to one of the models. Usually there will be from sixteen to thirty models to decide. The models are the tiles that are packed and input together to the assembly machine. Different tiles come together because they have been manufactured at the same time in the same press. Models represent one of each tile that can appear together, in their four different orientations. Images of every model were taken and preprocessed to extract the same features as the ones used for classification. Features used are the tile area, the shape, and the shadow pattern.

The tile shape, the contour, can be the same for different models. It can be different when a tile is presented to the machine in a different orientation, so the shape can be used to discard some models, but certainly many others will have the same shape. Also the shape will be used to compute the rotation of the tile to the model once it is classified. This is needed for assembling the tessellae with the right orientation.

Since area and shape can be the same for different models, the most characteristic and stable feature that really distinguishes a model from another, in a particular orientation, is the relief pattern. As mentioned, for two tiles with the same relief pattern, texture may vary due to different materials used in manufacturing.

With these considerations, we decided to use the shadow pattern, produced with lateral illumination, as a feature for classification. For this purpose, in the illumination chamber only one white light (a fluorescent bar) has been disposed on the lower part of the imaged area. So, images appear illuminated from below, and shadows become relevant. A "positive" relief (a mesa) is then highly illuminated at its beginning, near the light source, and poorly illuminated at its end, far from the light source. While a "negative" relief (a basin) is illuminated the other way round.

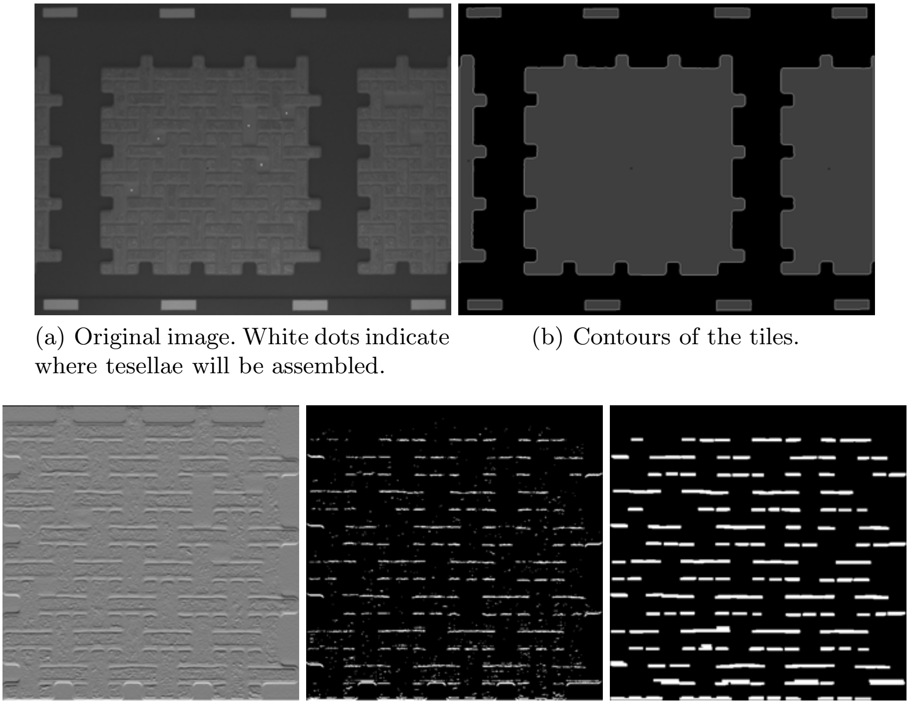
A convolution with a vertical 3×3 Sobel mask is done to estimate the vertical image gradient:

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} .$$

The image gradient is conformed to the rank [0..255] by adding 128 to every pixel. Since the mask is flipped horizontically and vertically by the convolution operation, bright transitions are in the lower values of the rank, dark transitions are in the upper values, and even parts are around the central value.

Afterwards, a thresholding is applied to the gradient image to classify pixels into three classes: evenly illuminated, bright and dark. Two thresholds are automatically computed from the histogram of the gradient image. A Gaussian distribution is fitted to the central part of the histogram, that represent the evenly illuminated parts. The mean and standard deviation are computed. We consider bright areas the pixels with a vertical gradient lower than the mean minus two times the standard deviation, and dark areas, shadows, as pixels with value bigger than the mean plus two times the standard deviation.

So the dark transitions, that are mostly thick horizontal lines, form the shadow pattern. This pattern will be matched to the shadow patterns of the models, whose images have been preprocessed in the same way. Morphological filtering is done to expand these lines to make the matching more robust. A dilation with a 3×3 square structuring element is applied to thicken the areas. Then, a *hit and miss* operation with a 15×2 structuring element (a thin line) is applied to eliminate thin and short lines, which may have arisen not from shadows but from the texture of the tile. After this, another dilation is performed. The size



(a) Original image. White dots indicate where tesellae will be assembled.

(b) Contours of the tiles.

(c) Vertical gradient.

(d) Shady pixels are remarked by segmenting vertical gradient.

(e) Shadow pattern after morphological filtering.

Fig. 5. Shadow pattern extraction process

of the structuring elements are set according to the size of the features in the images, that depends on relief height (quite stable) and image resolution (fixed).

Figure 5 shows the process of shadow pattern extraction. The central tile image is segmented, then the vertical gradient is estimated, the image gradient is segmented, and morphology is applied to form the final shadow pattern.

4 Model Classification

An image of each representative tile of a model set, in the four possible orientations, has been taken in the machine by inputing the model tiles perfectly horizontally aligned. This is important because in normal operation the rotation of a tile to the matching model has to be computed, so the tessellae can be assembled in the right positions and orientations. The machine works with one set of models at a time, although several sets have been prepared.

Several features are precomputed for every model: area, contour, and shadow pattern. The model set information is stored in a data base. In normal operation, the same features are computed for every tile present in the machine, and three tests are made to select the correct model: a test on the area, a test on the shape (the contour of the tile), and a test on the shadow pattern.

Every test discards the models that do not pass it. Some models will pass the first and the second tests, and only one model has to pass the last test. In this application we can be sure that the tile presented can be correctly classified to one of the models, but we also have to assure that it is matched to the correct model, so that it is very similar to that model, but different enough to all the others. A misclassification can result in blocking the robot manipulator that does the assembly, or even in damage to the robot arm tool.

Figure 6 shows two models of two different sets. The model in the top is the one that matches the tile in Figure 5. The model shown in the bottom belongs to another set. The image shows the extracted features, with the contour, the signature function, and the shadow pattern.

4.1 Test on Area

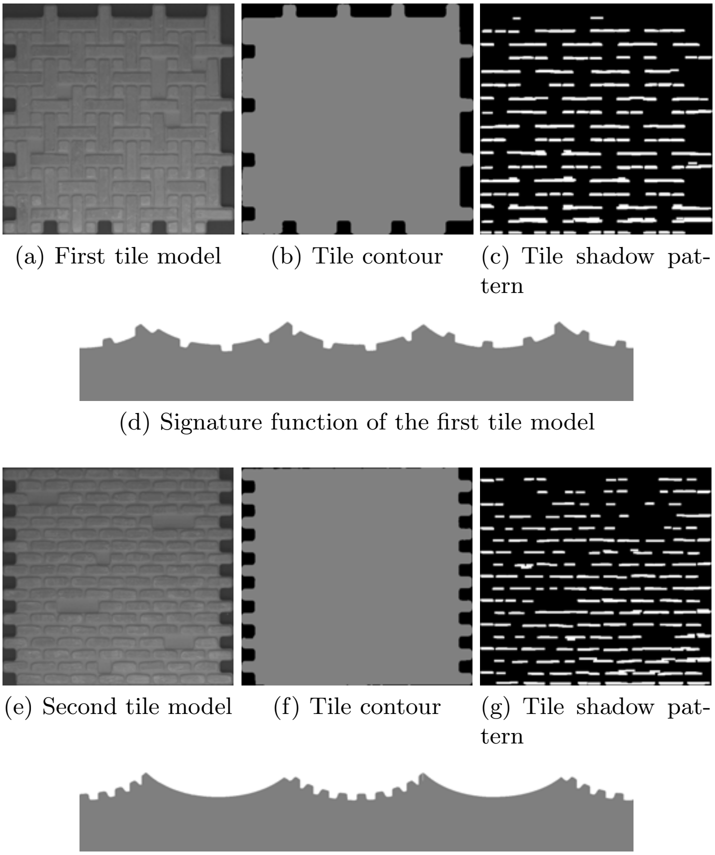
The test on area is correct if the relative area difference of the tile on the machine and the model under test is lower than a certain percentage. The percentage has been set to 10%, quite restrictive.

4.2 Test on Shape

The signature function of the contour of the tile is compared to the signature function of the contour of the test model. The signature function of a blob is a signal of distances from the center to the contour at regularly spaced angles. We have used an angular resolution of two degrees. A lower value could be used to increase resolution, but a compromise has to be taken between resolution and computational speed.

First, the rotation angle to align the contour of the tile to the contour of the model is computed by best circular correlation of the two signature functions. Then, rotating the contour, a value of dissimilarity between both functions is computed. We have used as this figure of merit the relative number of function positions where, after rotation, the relative difference of both functions is higher that a percentage (15% is used, that copes with image resolution and tile tolerances). The test finally succeeds if the figure of merit is lower than a percentage, again 15% has been used.

The test on shape not only assures that the shape of the tile is very similar to the shape of the test models that pass the test, but also provides the rotation angles to each of those models, angles that will be used to guide the robot manipulator to the correct positions and with the correct orientation for assembling the tessellae onto the tile.



(a) First tile model (b) Tile contour (c) Tile shadow pattern

(d) Signature function of the first tile model

(e) Second tile model (f) Tile contour (g) Tile shadow pattern

(h) Signature function of the second tile model

Fig. 6. Two preprocessed tile models

4.3 Test on the Shadow Pattern

So far, for every survival test model, the area and shape of the tile and of the model are very similar. The rotation angle and the centers of the tile and of the model are also known, so they can be aligned.

The test on the shadow pattern consists of overimposing the shadow pattern of the tile to the shadow pattern of the model, and evaluating a figure of merit on how similar (or dissimilar) the pattern is. The shadow patterns of all models were precomputed, and are available. To assure a misclassification will not occur, the result of this test must be only one survival model, whose shadow pattern has to be very similar to that of the tile, and quite different to the shadow pattern of all other models. The figure of merit used in this test is the number of pixels of the shadow pattern that do not overlap. The smaller this number, the more similar both patterns are. The test is correct for one model if this model has the smallest figure of merit, and the figures of merit for all the other models are bigger than a certain percentage of the smallest one. The percentage used is 20%.

For computing the figure of merit, the shadow pattern has to be aligned (translated and rotated) to the shadow pattern of the model. A difference image is computed, consisting of a binary image with the pixels that do not overlap marked. This image is filtered with a *hit and miss* morphological filter to eliminate thin and small non-overlapping lines, and to only take into account the areas where the shadow patterns are very different.

5 Implementation on a Robotics Machine

The presented classification method has been implemented on an industrial machine. The machine captures images from the two vision systems, detects the tessellae, classifies the tiles given a model set, and drives a manipulator to perform the assembly.

The classification provides the right model. A list of positions and orientation where tessellae have to be assembled onto the tile are computed by translating and rotating the ones specified in the model. The positions are tracked in the conveyor belts using rotation encoders. When a tile position and a corresponding tessella are in the manipulator work area, instructions are given to the manipulator to make the assembly.

The image area is calibrated by calibration marks, defining a real image 2D coordinate system. There is a correspondence between pixels in the image and coordinates (in mm.) in the image coordinate system. This image coordinate system is calibrated with the robotic manipulator 3D coordinate system. The conveyor belts translation vectors are also known by calibration, and the plane equation of the belts are also calibrated, both in the robot coordinate system. So positions over the conveyor belts are tracked in 3D in the robot coordinate system.

6 Results

The computer vision methods presented have been exhaustively tested, forming part of an industrial robotics assembly machine. The classification rate is higher than 99.5%. This percentage is continuously measured and reported by the system. Of course this is quite a bounded application, but industry demands this high success rates to accept the machines. Nowadays, there are 15 model sets defined in the system. A model set is usually formed by 16 or 20 models.

Problems in the machine have arisen not by misclassification, but by mechanical drifts that require adjustments of the calibration between the two vision systems, the conveyor belts, and the robot. This is done by performing a simple test periodically, and writing the calibration parameters to a configuration file.

7 Conclusions

We have presented some methods for a typical classification problem. Our main contribution is the selection of features to be used in the classification, particularly the shadow pattern. These features, together with a combination of simple tests, have proved to provide excellent results. Lateral illumination has been of great importance in enhancing the shadows, and in allowing a relatively easy feature extraction. Morphological filtering has been used, and proved to be an essential tool in this application.

The methods have been tested exhaustively in a real test bed, an industrial machine built ad-hoc for this application. Results are satisfactory, and we can affirm that the methods presented are robust.

Acknowledgements. The present work was partially supported by the projects CSD2007-00018 and DPI2008-06548-C03-01 from the Spanish Ministry of Science and Innovation, and by the project P1 07I438.01/1 from Fundació Caixa-Castelló.

References

1. Malamas, E.N., Petrakis, E.G.M., Zervakis, M., Petit, L.: A survey on industrial vision systems, application and tools. Image and Vision Computing 21, 171–188 (2003)
2. Huang, S.-J., Tsai, J.-P.: Robotic automatic assembly for random operating condition. International Journal on Advanced Manufacturing Technology 27, 334–344 (2005)
3. Bone, G.M., Capson, D.: Vision-guided fixtureless assembly of automotive components. Robotics and Computer integrated Manufacturing 19, 79–87 (2003)
4. Bozma, H.I., Yalçin, H.: Visual processing and classification of items on a moving conveyor: a selective perception approach. Robotics and Computer Integrated Manufacturing 18, 125–133 (2002)
5. Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M.: Shape from shading: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 21(8), 690–706 (1999)
6. Atkinson, G., Hancock, E.R.: Shape estimation using polarization and shading from two views. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(11), 2001–2017 (2007)
7. Han, F., Zhu, S.C.: A two-level generative model for cloth representation and shape from shading. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(7) (2007)

Using Remote Data Mining on LIDAR and Imagery Fusion Data to Develop Land Cover Maps

Jorge García-Gutiérrez[1], Francisco Martínez-Álvarez[2], and José C. Riquelme[1]

[1] Department of Computer Science, University of Seville, Spain
{jgarcia,riquelme}@lsi.us.es

[2] Area of Computer Science, Pablo de Olavide University, Spain
fmaralv@upo.es

Abstract. Remote sensing based on imagery has traditionally been the main tool used to extract land uses and land cover (LULC) maps. However, more powerful tools are needed in order to fulfill organizations requirements. Thus, this work explores the joint use of orthophotography and LIDAR with the application of intelligent techniques for rapid and efficient LULC map generation. In particular, five types of LULC have been studied for a northern area in Spain, extracting 63 features. Subsequently, a comparison of two well-known supervised learning algorithms is performed, showing that C4.5 substantially outperforms a classical remote sensing classifier (PCA combined with Naive Bayes). This fact has also been tested by means of the non-parametric Wilcoxon statistical test. Finally, the C4.5 is applied to construct a model which, with a resolution of 1 m^2, obtained precisions between 81% and 93%.

Keywords: Data mining, remote sensing, LIDAR, imagery, LULC.

1 Introduction

Remote sensing has been a very important tool to study the natural environment for long. It has been applied to lots of different tasks such as species control [1] or landscape control [2]. These techniques are of the utmost importance to reduce costs since they increase the products development speed, helping thus the experts to make decisions. Due to the remote sensing relevance, many researchers have invested much time to find how to develop new algorithms to improve the quality of the results.

One of the most important products in remote sensing is land use and land cover maps (LULC). They are used to develop policies in order to protect specially interesting areas from both environmental and economic points of view. The automatic generation is a very desirable feature since they are generated for covering large zones. To efficiently solve this problem, supervised learning is usually applied from a small quantity of data previously classified by human experts [3]. Hence, multispectral images, hyperspectral images and orthophotography have been widely used in many applications although they have their own

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 378–387, 2010.

limitations, e.g., the most useful information is usually on the visible spectrum band which is easily affected by shadows. The apparition of new sensors has caused that a new research line appears, which tries to overcome imagery problems fusing them with new technologies. LIDAR (LIght Detection And Ranging) is one of these new sensors. It is an optical technology that measures properties of scattered light in the near infrared to find range and/or other information of a distant target. The method to determine distance to an object or surface is to use laser pulses. The range to an object is determined by measuring the time delay between transmission of a pulse and detection of the reflected signal. In this way, LIDAR is able to extract the heights of objects and its fusion with imagery boosts any remote sensing technique so that it has been exploited with several purposes [4], [5].

Fusion among sensors increases data size. In this context, intelligent techniques are a must if an automatical process is required and particulary, remote data mining is a very suitable tool to deal with problems associated to big size data. With this in mind, some authors have started to use intelligent techniques like artificial neuronal networks (ANN) [6] or support vector machines(SVM) [7]. But the most used method is still based on classical statistics methods and concretely, the traditional principal component analysis (PCA) and the application of maximum likelihood principle [8], i.e., a Naive Bayes classifier.

In this work, a supervised method to obtain LULC automatically is shown and a comparison between two techniques for supervised learning in remote sensing is established with two purposes:

- Show the quality of models when intelligent techniques are applied on LIDAR and imagery fusion data.
- Show the importance of using well-known machine learning algorithms in the remote sensing context that outperforms most classical statistics procedures.

The rest of the paper is organized as follows. Section 2 provides an exhaustive description of the data used in this work. Section 3 describes the methodology used, highlighting the feature selection and model extraction processes. The results achieved are shown in Section 4 and, finally, Section 5 is devoted to summary the conclusions and to discuss future lines of work.

2 Data Description

The study area is located in the north of Galicia (Spain) as depicted in Figure 1. It is basically composed by a small residential zone and a forest zone, whose dominant species is *Eucalyptus Globulus*. In geomorphologic terms, in spite of the altitudes varying between 230 and 370 meters, the relief of the zone is quite accentuated.

The LIDAR data were acquired in November 2004 with *Optech's ALTM 2033* from a flight altitude of 1500 meters. The LIDAR sensor works with a laser wavelength of 1064 nm and the beam divergence was set to 0.3 mrad. The pulsing frequency was 33 kHz, the scan frequency 50 Hz, and the scan angle

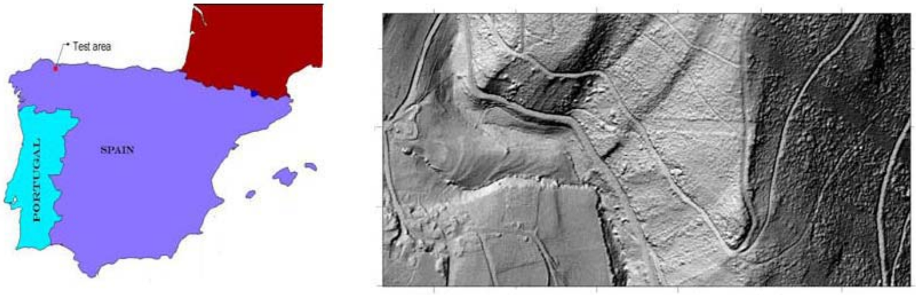


Fig. 1. Study area location and digital elevation model (DEM) used in this work

10 degrees. The first and last return pulses were registered. The complete study area was flown in 18 strips and each strip was flown three times, which gave an average measuring density of about four points per square meter.

A digital elevation model (DEM) was extracted from the LIDAR data using an adaptative morphologic filter method [9] to rectify object heights. The resulted DEM is illustrated in Figure 1. Moreover, a previous orthophoto is used to extract features from the visible spectrum band. It was taken with a resolution of 0.5 meters of the same zone and with similar atmospheric conditions to the moment of LIDAR acquisition flight.

Leaning on the orthophoto and with the help of previous knowledge about the study zone, a training base formed by 5570 pixels was selected (5% out of total, approximately) and classified into 5 different classes: road, farming land or bare earth, middle vegetation, high vegetation and buildings. In addition, in the same previous study, 317 specially interesting pixels were selected and classified to make up a hold-out test base according to the traditional testing in remote sensing.

3 Methodology

In order to classify LULC, a general method based on remote data mining techniques is applied. First, a raster matrix is built with a resolution of 1 meter. Every raster cell represents a squared meter pixel of the study area which contains several different measures extracted from LIDAR data (based on signal intensity, distribution and height) and images (visible spectrum bands). Then, a feature selection phase reduces the total number of features to build a supervised learning model. Later, an intelligent technique is used to generate a model. In this work, two different classifiers are used to make a comparison: a C4.5 decision tree [10] and a classical principal components analysis (PCA) combined with a Naive Bayes classifier which uses the maximum likelihood principle, whose combination was first used in [11]. An overall view of the whole classification process will be described in detail in the following subsections.

3.1 Feature Selection

All generated pixels have 63 different features based on LIDAR data and visible spectrum bands, which are in the geographical zone limited by the pixel itself. These features can be classified as intrapixel or interpixel. The intrapixel features are those which are calculated with data found within a pixel, whilst the interpixel features are those which are characterized as defining a relation between each pixel and its eight adjacent neighbors. With these features, characterization of the terrain is attempted, formalizing the visual differences or morphologies of the different classes. Most of them have been extracted from literature [12] and classical remote sensing applications. However, some are original of this work like SNDVI and EMP.

The Normalized Difference Vegetation Index (NDVI) is a simple numerical indicator that can be used to analyze remote sensing measurements and assesses whether the target being observed contains live green vegetation or not. In Equation 1, NDVI is calculated from the red band value (R) and the near infrared band value (NIR) which is not in the visible spectrum.

$$NDVI = \frac{NIR - R}{NIR + R} \tag{1}$$

In this work, a new attribute SNDVI (Simulated NDVI) has been generated using the intensity (I) from LIDAR as near-infrared value which approximates the real NDVI value, which cannot be calculated since no NIR information is available in LIDAR data. This parameter is calculated as follows:

$$SNDVI = \frac{I - R}{I + R} \tag{2}$$

LIDAR point density is another important characteristic to be analyzed. EMP feature counts the number of empty pixels that surrounds the current pixel in a eight-adjacent neighborhood. It helps to detect water areas because most of laser energy does not reflect on water and responses from flooded zones are not registered by the sensor.

Features from pixels in the training set are submitted to a process of selection. In this case, a correlation based feature subset selection (CFS) has been applied. CFS evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy among them. Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. The 21 final selected predictor variables after CFS execution are listed in Table 1.

3.2 Model Extraction

The next phase consists in executing the classification algorithms. Two kinds of approaches are proposed to extract the model: the C4.5 algorithm [10] and a combination between a PCA and a Naive Bayes classifier [11]. The two models

Table 1. Twenty-one candidate predictor variables after the feature selection phase

Variable	Description	Type
SNDVIMIN	Simulated NDVI minimum	Intrapixel
SNDVIMEAN	Simulated NDVI mean	Intrapixel
SNDVISTDV	Simulated NDVI standard deviation	Intrapixel
IMIN	Intensity minimum	Intrapixel
IMAX	Intensity maximum	Intrapixel
IMEAN	Intensity mean	Intrapixel
HMIN	Height minimum	Intrapixel
HMAX	Height maximum	Intrapixel
HMEAN	Height mean	Intrapixel
HSTD	Height standard deviation	Intrapixel
HCV	Height coefficient of variation	Intrapixel
PEC	Penetration coefficient	Intrapixel
PCT31	Percentage third return out of first return	Intrapixel
IRVAR	Intensity red band variance	Intrapixel
IRMEAN	Intensity red band mean	Intrapixel
IGVAR	Intensity green band variance	Intrapixel
IGMEAN	Intensity green band mean	Intrapixel
IGSKE	Intensity green band skewness	Intrapixel
IGKURT	Intensity green band kurtosis	Intrapixel
SLP	Slope	Interpixel
EMP	Empty LIDAR surrounding pixels	Interpixel

are extracted by means of the data mining environment WEKA [13]. For the C4.5 execution, the J48 implementation from Weka is selected. In both cases, the method is executed with the parameters set as default by WEKA.

Although both methods have similar computational costs, there are significant differences when each one expresses the model. On the one hand, decision trees are more intuitive for non-experts users and for this reason has been widely used in many different industrial and engineer applications and even in remote sensing. On the other hand, when the training set is more complex, generated trees are more difficult to read because branches and nodes are increased in number whilst a PCA and Naive Bayes approach maintains a more constant number of attributes and levels.

Bearing in mind, an analysis of functional aspects such as the precision on the results is necessary to know which one is more suitable for this study zone.

4 Results

In order to evaluate how much improvement can be achieved with each intelligent technique application, two different kinds of testing were carried out. First, a cross-validation on training data. Second, a hold-out testing on specially interesting pixels which is the common testing in classical remote sensing. The training base and the test set were extracted as referred in Section 2.

4.1 Comparison of Methods

After the execution of the 10-fold cross-validation tests, it can be stated that the C4.5 decision tree obtains better results. In Tables 2 and 3, the total and partial precisions as well as the kappa index of agreement (KIA) obtained for the two techniques on the 5570 training pixels are shown. The two techniques obtain high accuracy, but the application of the decision tree produces an improvement of almost fifteen percentile points. Furthermore, the potential of decision trees to indicate which features in the original set are more interesting –which allows a new automatic selection of attributes– must be highlighted.

It is well-known that just a cross-validation result or a hold-out test and a rank method among techniques is not enough to confirm than a technique is better than another or viceversa, even when many authors have used this technique in their studies. For this reason, to evaluate the statistical significance of the measured differences in algorithm ranks, a procedure suggested in several works [14] for robustly comparing classifiers across multiple datasets is used. In this work, there is only one dataset because LIDAR data has high costs to be

Table 2. Cross-validation summary of the tests on C4.5 and confusion matrix

User class \	Roads	Farming	Middle	High	Buildings
C4.5		lands	vegetation	vegetation	
Roads	219	13	13	2	0
Farming lands	9	2111	81	8	0
Middle vegetation	10	80	1389	77	14
High Vegetation	6	8	61	1130	3
Buildings	1	0	13	10	312
Producer's accuracy	0.898	0.954	0.892	0.921	0.948
User's accuracy	0.887	0.956	0.885	0.935	0.929
Total accuracy	0.927				
KIA	0.897				

Table 3. Cross-validation summary of the tests on PCA + Naive Bayes and confusion matrix

User class \	Roads	Farming	Middle	High	Buildings
PCA + NV		lands	vegetation	vegetation	
Roads	212	13	10	1	11
Farming lands	4	2073	107	25	0
Middle vegetation	10	639	696	169	56
High Vegetation	2	7	144	1034	21
Buildings	23	0	8	14	291
Producer's accuracy	0.845	0.759	0.721	0.832	0.768
User's accuracy	0.858	0.938	0.443	0.856	0.866
Total accuracy	0.773				
KIA	0.677				

obtained. So, the training set is randomly split in five subsets. Then, a 10-fold cross-validation is made for every subset. At the end, there are 50 measures for every algorithm and then, the procedure is carried out. Hence, it is possible to use average ranks that provide a fair comparison of the algorithms analyzing the 50 measures, revealing that, on average, C4.5 ranks first. Given the measured average ranks, it is possible to apply a Wilcoxon test to check whether the ranks are significantly different to consider them different populations or not (which is the expected under the null hypothesis). Leaning on a statistical package (MATLAB), p value for the Wilcoxon test has resulted on a value of $6.7266e-18$ so the null hypothesis is rejected having found that the ranks are significantly different (at α = *0.05*).

4.2 Model Precision

Once the model developed by C4.5 is extracted, it is applied to every pixel. Then, a new empirical test is done. In this way, a checking process on the test set (317 specially interesting pixels) is carried out and classes which they pertain are evaluated. In Tables 4 and 5, it is possible to observe the results of the test through the confusion matrix, producer and user's accuracies, and the kappa estimator.

Table 4. Hold-out summary on C4.5 and confusion matrix

User class \ C4.5	Roads	Farming lands	Middle vegetation	High vegetation	Buildings
Roads	14	1	0	0	0
Farming lands	2	106	9	0	0
Middle vegetation	3	5	56	15	0
High Vegetation	1	0	10	65	8
Buildings	1	0	3	1	17
Producer's accuracy	0.667	0.946	0.718	0.802	0.68
User's accuracy	0.9333	0.906	0.709	0.774	0.773
Total accuracy	0.814				
KIA	0.7457				

Since the pixels in the test set had been selected for the special difficulty shown to be classified, both methods decrease their global precision. Anyway, C4.5 improves the results by PCA and Naive Bayes in almost 8% which is still a remarkable difference. Moreover, some classes decrease their accuracy dramatically when using PCA and Naive Bayes approach which is inadmissible when dealing with so few classes.

In Figure 2, the resulted global classification next to the training base and the initial input data (orthophoto and LIDAR intensity image) for a C4.5 decision tree model are shown. As it can be appreciated, the LULC map accuracy is very high and its automatic generation is much faster than a manual creation by an expert.

Table 5. Hold out summary on PCA + Naive Bayes and confusion matrix

User class \ PCA + NV	Roads	Farming lands	Middle vegetation	High vegetation	Buildings
Roads	14	1	0	0	0
Farming lands	1	101	13	2	0
Middle vegetation	1	24	29	24	1
High Vegetation	0	1	7	74	2
Buildings	4	0	0	3	15
Producer's accuracy	0.7	0.795	0.591	0.718	0.833
User's accuracy	0.933	0.863	0.367	0.881	0.682
Total accuracy	0.735				
KIA	0.632				

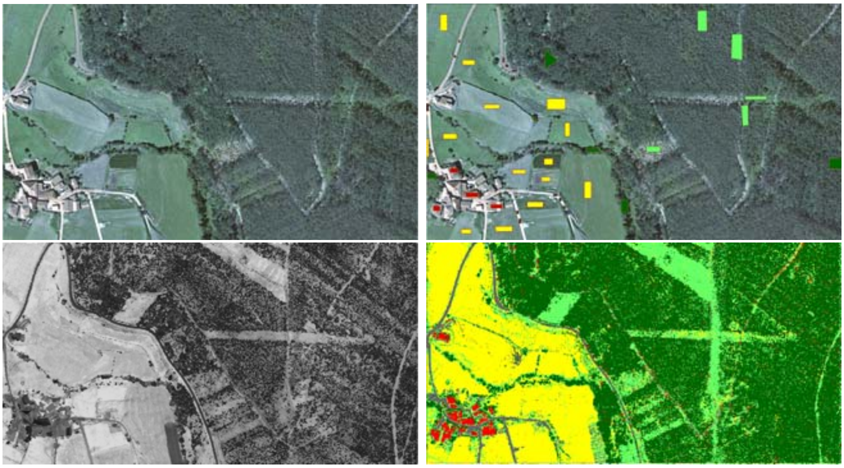


Fig. 2. From up to down and left to right: original orthophoto, training set, LIDAR intensity image and final result. Classes are colored as: urban areas in red, roads in dark grey, farming lands or naked earth in yellow, medium vegetation in light green and high vegetation in dark green.

5 Conclusions and Future Work

An approach based on LIDAR and imagery fusion data as well as the application of intelligent techniques have been tested to classify land coverage of a typical area of the north of Spain in this work. The objective was to evaluate data fusion capabilities and to establish a comparison between two different techniques to classify the fusion data: a classical statistics principal components analysis and a Naive Bayes classifier (based on the maximum likelihood principle) and C4.5, which is a well-known machine learning decision tree generator.

The developed method is based on a pixel-oriented focus which classifies raw data in five different classes. In this way, a series of features were calculated from fusion data (some of them are original in this work), which are associated with each pixel. Thereafter, an attribute selection method was applied to reduce the set of variables to consider. Lastly, the selected classifier extracted a model.

A thorough study was performed to select which algorithm was the best between the two possible ones. The tests carried out selected the algorithm C4.5, which generated a decision tree, as the model with the best fit. The results between 82% and 93% of accuracy depending on the kind of test applied showed that C4.5 outperforms its rival in between eight and fifteen percentile points. They also have demonstrated that different types of terrain can be characterized using intelligent techniques in a multi-staged process using LIDAR and image data and, moreover, robust and well-known machine learning algorithms are perfectly suitable to improve classification results in remote sensing data better than more classical methods in our study zone conditions.

In relation to future works, two problems arise. The first problem is the improvement of the classification method itself, as some problems have already been detected. This can be solved using several techniques joined in ensembles. The second one is related to dependence of results on training set. Some authors have detected problems when a classifier is not trained by well-balance or real data and its effects on the test phase. In addition, outliers (salt and pepper noise) are a great problem due to the imprecise training set definition or intrinsic problems on data, e.g., variability of LIDAR intensity depending on the number of returns per pulse. These problems are much more common than it would be desirable. There are two possible ways to explore in order to solve this problem. On the one hand, the easier option is to introduce an instance selection phase. On the other hand, migrating from a supervised learning to a semi-supervised learning can be another more interesting option in which the system helped by evolutive computation would extract its own training data just taking the number of classes from the user.

Acknowledgments

We would like to thank the Land, Underground and Biodiversity Laboratory of University of Santiago de Compostela for all the support received in the development of this work and especially, to thank Luis Gonçalves-Seco, for all the time he invested that allowed this work to be completed.

References

1. Hyde, P., Dubayah, R., Walker, W., Blair, J.B., Hofton, M., Hunsaker, C.: Mapping forest structure for wildlife habitat analysis using multi-sensor (LIDAR, SAR/INSAR, ETM+, Quickbird) synergy. Remote Sensing of Environment 102, 63–73 (2006)

2. Wang, C., Glenn, N.F.: Integrating LIDAR intensity and elevation data for terrain characterization in a forested area. IEEE Geoscience and Remote Sensing Letters 6(3), 463–466 (2009)
3. Johansen, K., Coops, N.C., Gergel, S.E., Stange, Y.: Application of high spatial resolution satellite imagery for riparian and forest ecosystem classification. Remote Sensing of Environment 110, 29–44 (2007)
4. Suárez, J.C., Ontiverosa, C., Smith, S., Snapec, S.: Use of airborne LIDAR and aerial photography in the estimation of individual tree heights in forestry. Computers & Geosciences 31, 253–262 (2005)
5. Koetz, B., Sun, G., Morsdorf, F., Ranson, K., Kneubuhler, M., Itten, K., Allgower, B.: Fusion of imaging spectrometer and LIDAR data over combined radiative transfer models for forest canopy characterization. Remote Sensing of Environment 106, 449–459 (2007)
6. Nguyen, M.Q., Atkinson, P.M., Lewis, H.G.: Superresolution mapping using a hopfield neural network with LIDAR data. IEEE Geoscience and Remote Sensing Letters 2(3), 366–370 (2005)
7. Mazzoni, D., Garay, M.J., Davies, R., Nelson, D.: An operational MISR pixel classifier using support vector machines. Remote Sensing of Environment 107, 149–158 (2007)
8. Bork, E.W., Su, J.G.: Integrating LIDAR data and multispectral imagery for enhanced classification of rangeland vegetation: A meta analysis. Remote Sensing of Environment 111, 11–24 (2007)
9. Goncalves-Seco, L., Miranda, D., Crecente, R., Farto, J.: Digital terrain model generation using airborne LIDAR in florested area of Galicia, Spain, Lisbon, Portugal. In: Proceedings of 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences, pp. 169–180 (2006)
10. Quinlan, J.R.: Improved use of continuous attributes in C4.5. Journal of Artificial Intelligence Research 4, 77–90 (1996)
11. Mutlu, M., Popescu, S.C., Stripling, C., Spencer, T.: Mapping surface fuel models using LIDAR and multispectral data fusion for fire behavior. Remote Sensing of Environment 112, 274–285 (2008)
12. Hudak, A.T., Crookston, N.L., Evans, J.S., Halls, D.E., Falkowski, M.J.: Nearest neighbor imputation of species-level, plot-scale forest structure attributes from LIDAR data. Remote Sensing of Environment 112, 2232–2245 (2008)
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1), 10–18 (2009)
14. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)

Face Recognition Using Curvelet Transform

Hana Hejazi and Mohammed Alhanjouri

Islamic university of Gaza, Gaza, Palestine
hhejazi@students.iugaza.edu.ps, mhanjouri@iugaza.edu.ps

Abstract. This paper presents a new method for the problem of human face recognition from still images. This is based on a multiresolution analysis tool called Digital Curvelet Transform. Curvelet transform has better directional and edge representation abilities than wavelets. Due to these attractive attributes of curvelets, we introduce this idea for feature extraction by applying the curvelet transform of face images twice. The curvelet coefficients create a representative feature set for classification. These coefficients set are then used to train gradient descent backpropagation neural network (NN). A comparative study with wavelet-based, curvelet-based, and traditional Principal Component Analysis (PCA) techniques is also presented. High accuracy rate of 97% and 100% achieved by the proposed method for two well-known databases indicates the potential of this curvelet based curvelet feature extraction method.

Keywords: Face Recognition, Backpropagation Neural Network, Digital Curvelet Transform.

1 Introduction

As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past few years. This evidenced by the emergence of face recognition conferences and systematic empirical evaluations of face recognition techniques. There are at least two reasons for this trend; the first is the wide range of commercial and law enforcement applications and the second is the availability of feasible technologies after more three decades of research [1].

The strong need for user-friendly systems that can secure our assets and protect our privacy without losing our identity in a sea of numbers is obvious. At present, one needs a PIN to get cash from an ATM, a password for a computer, a dozen others to access the internet, and so on. Although extremely reliable methods of biometric personal identification exist, e.g., fingerprint analysis and retinal or iris scans, these methods rely on the cooperation of the participants, whereas a personal identification system based on analysis of frontal or profile images of the face is often effective without the participant's cooperation or knowledge [1]. And face recognition among biometric identification systems are natural and does not have less negative responses in using from people, and thus much more research efforts have been pouring into this area among biometric areas [2].

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 388–396, 2010.

The major difficulty with representing faces as a set of features is that it assumes some a priori knowledge about what are the features and/or what are the relationships between them that are essential to the task at hand.

Many face recognition techniques have been developed over the past few decades. One of the most successful and well-studied techniques to face recognition is the Multiresolution analysis tools, especially wavelets, have been found useful for analyzing the information content of images; this is lead to be used in areas like image processing, pattern recognition and computer vision. Following wavelets, other multiresolution tools like contourlets, ridgelets etc. were developed. ‘Curvelet Transform’ is the recent addition to this list of multiscale transforms.

Face Recognition by Curvelet Based Feature Extraction was reported by T. Mandal, A. Majumdar, and Q.M. J. Wu [3]. They vary the gray scale resolution from 256 to 16 and 4. As the images are quantized, only the bolder curves of the face image will remain. However, they didn’t improve the recognition accuracy neither for sidewise tilted images nor by cropping images and making tilt corrections nor using other voting schemes as well, in addition they mainly depend on the voter to get good accuracy.

Later [4], they used a new feature extraction technique using PCA on curvelet domain, which applied on still face images. They have investigated the possibility of curvelet transform to be used in combination with one linear analysis tool. But they did not explore the possibility of curvelet to stand alone.

Currently, researchers are using many techniques for feature extraction as wavelet, Principal Component Analysis and more others. In our work, we developed new system to recognize the still face images; the system is divided into three main stages: preprocessing, feature extraction and classification stage that contents two phases training and testing. In feature extraction stage, the images are decomposed into its approximate and detailed components using curvelet transform. These sub-images thus obtained are called curvefaces. These curvefaces greatly reduce the dimensionality of the original image. Thereafter only the approximate components are selected to perform further computations, we have investigated the possibility of curvelet transform to be used in combination with other analysis tool, and possibility of curvelet to implement alone with two levels.

2 Background

2.1 Curvelet Transform

The Curvelet transform is a higher dimensional generalization of the Wavelet transform designed to represent images at different scales and different angles [5].

There are two implementation of Curvelet: The first digital transformation is based on unequally spaced fast Fourier transforms (USFFT) many times, while the second is based on the wrapping of specially selected Fourier samples. The two implementations essentially differ by the choice of spatial grid used to translate curvelets at each scale and angle. Both digital transformations return a table of digital curvelet coefficients indexed by a scale parameter, an orientation parameter, and a spatial location parameter. Curvelet via wrapping has been used for this work since it is the fastest curvelet transform currently available [6].

If we *have* $g[t_1,t_2]$, $t_1 \geq 0$, $t_2 < n$ as Cartesian array and $\hat{g}\ [n_1,n_2]$ to denote its 2D Discrete Fourior Transform, then the architecture of curvelets via wrapping is as follows[5]:

1. 2D FFT (Fast Fourior Transform) is applied to obtain Fourior samples $\hat{g}[n_1,n_2]$.
2. For each scale j and angle l, the product $\tilde{U}_{j,l}\ [n_1,n_2]\ \hat{g}[n_1,n_2]$ is formed, where $\tilde{U}_{j,l}\ [n_1,n_2]$ is the discrete localizing window.
3. This product is wrapped around the origin (*eight angles around origin are used in this paper*) to obtain $\breve{g}_{j,l}[n_1,n_2] = W(\tilde{U}_{j,l}\,\hat{g})\ [n_1,n_2];$ where the range for n_1,n_2 is now $0 \leq n_1 < L_{1,j}$ and $0 \leq n_2 < L_{2,j};\ L_{1,j} \approx 2^j$ and $L_{2,j} \approx 2^{j/2}$ are constants.
4. Inverse 2D FFT is applied to each $\breve{g}_{j,l}$, hence creating the discrete curvelet coefficients.

2.2 Curvelet Based Feature Extraction

Curvelet transform has been developed especially to represent objects with *'curve-punctuated smoothness'* i.e. objects which display smoothness except for discontinuity along a general curve; images with edges are good examples of this kind of objects. In a two dimensional image two adjacent regions can often have differing pixel values. Such a gray scale image will have a lot of "edges" i.e. discontinuity along a general curve and consequently curvelet transform will capture this edge information. To form an efficient feature set it is crucial to collect these interesting edge information which in turn increases the discriminatory power of a recognition system [4].

Our face recognition system is divided into three stages: preprocessing, feature extraction and classification stage. As preprocessing, the images quantized to 16 gray-scale images, and we don't use any denoising technique because the international databases that were used are cleared images. While in feature extraction stage, the images are decomposed into its approximate and detailed components using two level of curvelet transform. These sub-images thus obtained are called *curvefaces*. These *curvefaces* greatly reduce the dimensionality of the original image. Thereafter only the approximate components are selected to perform further computations, as they account for maximum variance. Thus, a representative and efficient feature set is produced. And finally the classification stage arranged the resultant features to train the classifier and later make testing for images by applying same operation on test images. Mean squared error is employed to perform the classification task that achieved by using back-propagation neural network. Figure 1 shows the curvelet coefficients of a face from ORL dataset decomposed at scale = 2 and angle = 8, we can note that the first image is the original image, the first image in the second row is the approximate coefficients and others are detailed coefficients at eight angles.

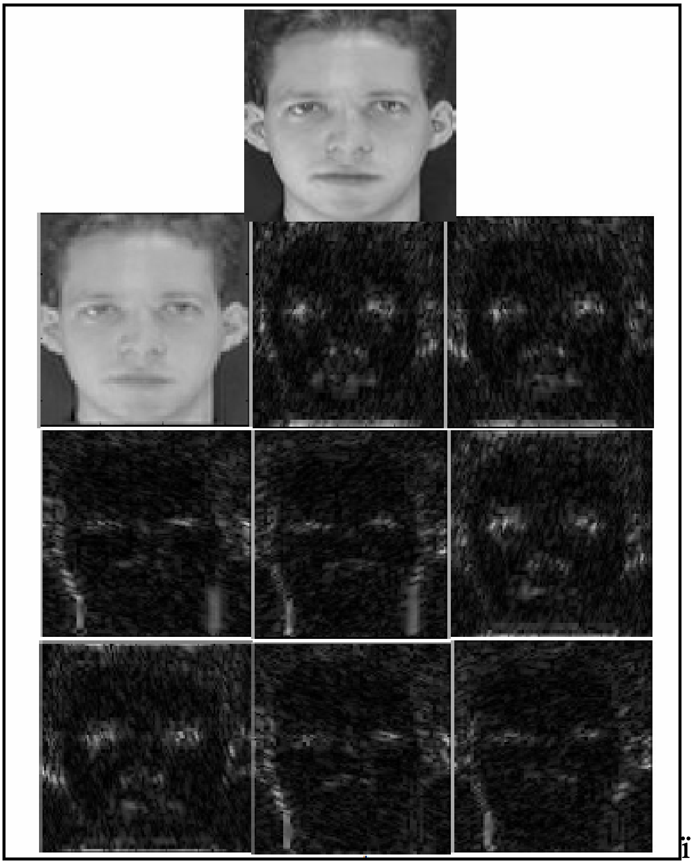


Fig. 1. Curvelet Coefficients (approximate and details)

3 Methodology

3.1 Datasets

This work uses two types of dataset from different sources: ORL (AT&T) database and Essex Grimace database, both sets are used to implement different Algorithms to recognize the human face.

ORL (AT&T) database [7] contains distinct face images sets for 40 persons with dimension of 92×112, and each set consists of 10 different images for the same person. For some persons, images were taken at different times varying the lighting, facial expression (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the faces in an upright, frontal position (with tolerance for some side movement). Sample images of this dataset are shown in figure 2.

Essex Grimace database [8] contains sequence face images for 18 persons each one have 20 images (180×200), all images taken with a fixed camera for male and female. During the sequence, the subject moves his/her head and makes grimaces which get more extreme towards the end of the sequence. Images are taken against a plain background, with very little variation in illumination. Sample images of this database are shown in figure 3.

Fig. 2. Sample images from ORL database

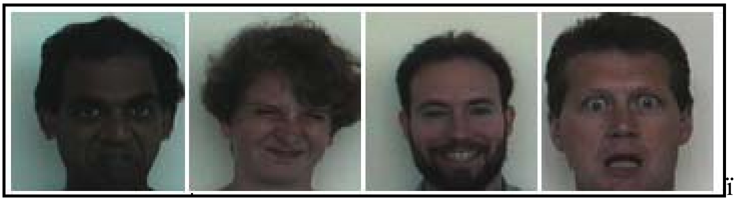
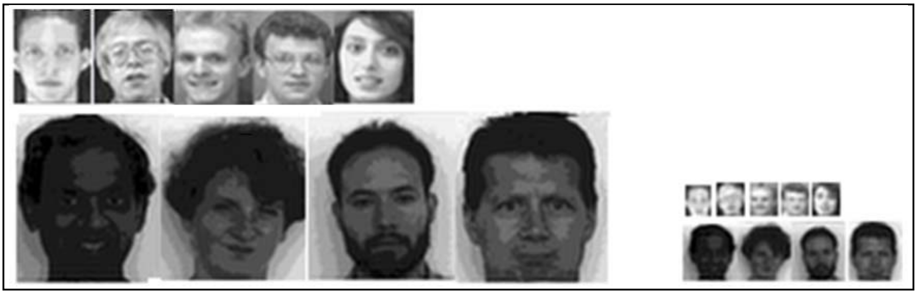


Fig. 3. Sample images from Essex Grimace database

3.2 Results

In the first part of our experimental study, 6 images per each subject for ORL and 9 images per each subject for Essex Grimace database are randomly selected as training set; the rest construct the test set. Color images of Essex Grimace database are converted to gray scale images as preprocessing stage.

The original images are decomposed using curvelet transform at scale = 3 and angle = 8. Thus 25 components are produced, including 1 approximate and 24 detailed sub-bands. The resolution of the approximate sub-band is reduced to 31x37 and 61x67 for images of ORL and Essex Grimace database respectively. To further reduce



(a) One level curvelet (b) Two level curvelet

Fig. 4. Sample images of curvelet transform

the dimensionality, curvelet transform, at scale = 3 and angle = 8, is applied once again on these approximate components only. The resolution of the approximated sub-band became 11x13 and 21x23 for images of ORL and Essex Grimace database respectively as shown in figure 4. After curvelet sub-images are produced, gradient descent back-propagation neural network with adaptive learning rate is employed to perform the identification task.

Number of hidden layer and number of nodes in each layer are varied to obtain the optimal back-propagation neural network topology for highest performance with 2-level curvelet transform. As shown in table 1, the number of hidden layer and the number of nodes in each one are important parameter to improve the recognition rate (success rate), we can note that when the NN topology has one hidden layer, the best recognition rate is achieved of 97% at 50 nodes in this layer while when second hidden layer is used, we obtained 95% recognition rate at also 50 nodes in the second hidden layer, clearly, the NN topology should be contains one hidden layer with 50 nodes to get the highest performance by using 2-level curvelet for ORL dataset. The same results are obtained for Essex Grimce dataset.

The mean squared error rates for both ORL and Essex Grimace database are illustrated in figure 5 and figure 6, respectively.

Table 1.a. Recognition rate for ORL dataset quantized16

Using one hidden layer		**Using two hidden layer**	
No. of nodes	**Success rate**	**No. of nodes in 2nd layer**	**Success rate**
80	89%	50 +80	88%
70	90%	50+70	90%
60	93%	50+60	92%
50	**97%**	50+50	95%
40	91%	50+40	90%
30	89%	50+30	87%
20	87%	50+20	85%

Table 1.b. Recognition rate for Essex Grimace dataset quantized16

Using one hidden layer		**Using two hidden layer**	
No. of nodes	**Success rate**	**No. of nodes in 2nd layer**	**Success rate**
80	94%	50 +80	91%
70	96%	50+70	93%
60	97%	50+60	95%
50	**100%**	50+50	98%
40	95%	50+40	92%
30	90%	50+30	88%
20	88%	50+20	85%

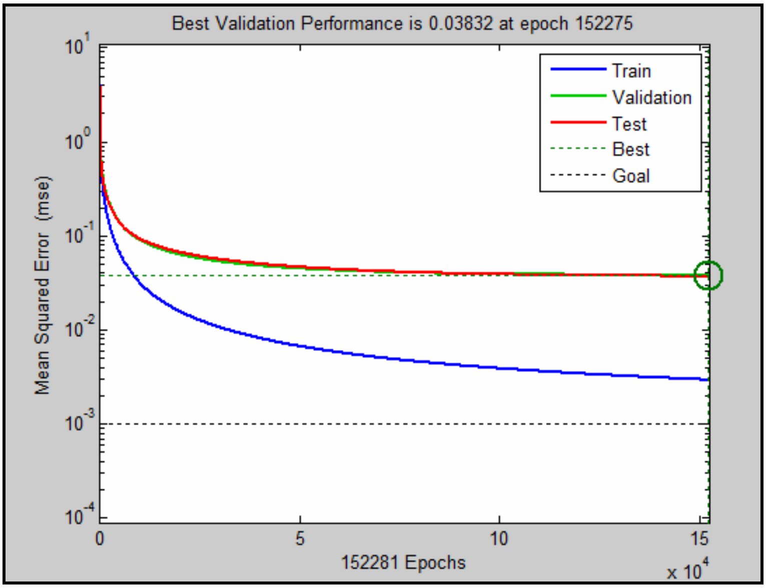


Fig. 5. Performance rate for ORL

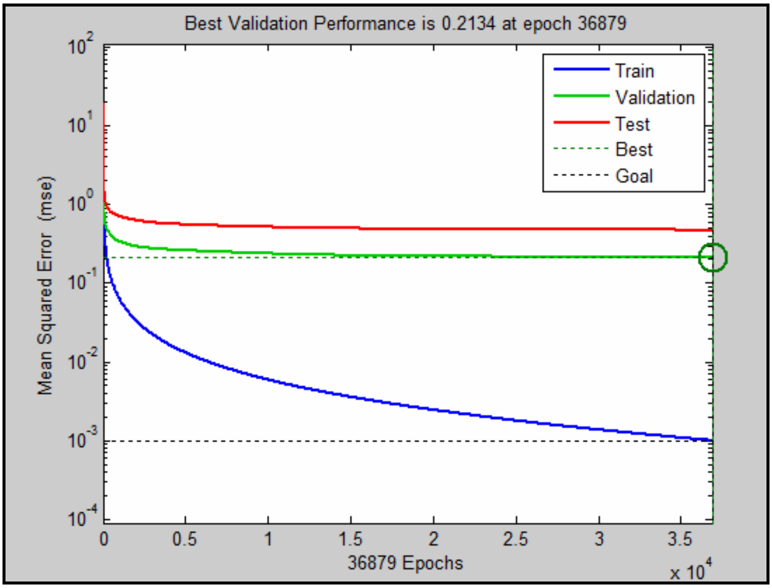


Fig. 6. Performance rate for Essex Grimace

3.3 Comparative Study

In the previous section, different results of curvelet-based curvelet technique have been presented. In order to show the capability of the proposed method we have compared it against the most popular existing techniques. A 3-level wavelet decomposition using 'Haar' wavelet was performed for wavelet based PCA technique. The best highest 50 eigenvectors were selected for both curvelet and wavelet based PCA; training images for Essex and ORL being 8 and 6 per subjects respectively. Table 2 summarizes the results of our comparative study that show the recognition rate of three different methods and compare the results with the proposal method in this paper using two level curvelet transform. The results show that the proposal method has the highest Recognition rate.

Table 2. Comparative study of four techniques for face recognition

Recognition method	**Recognition rate**	
	Essex Grimace	**ORL**
Eigenface	70%	93%
Wavelet face + PCA	98%	94%
Curvelet +PCA	100%	96%
Proposed method (Two level curvelet transform)	100%	97%

4 Conclusion

We have introduced a new feature extraction technique from still images using curvelet on curvelet domain which has been evaluated on two well-known databases. Our technique has been found to be robust against extreme expression variation as it works efficiently on Essex database. The subjects in this dataset make grimaces, which form edges in the facial images and curvelet transform captures this crucial edge information. The proposed method also seems to work well for ORL database, which shows significant variation in illumination and facial details.

From the comparative study, it is evident that curvelet based curvelet completely outperforms standard eigenface technique; it also supersedes both wavelet based PCA scheme and curvelet based PCA scheme. The promising results indicate that curvelet transform can stand alone as an effective solution to face recognition problem in future. We have investigated the possibility of curvelet transform to be used lonely.

Future work has been suggested towards exploring the merrier curvelet. Different classification techniques can also be employed.

References

[1] Zhao, W., Chellappa, R., Rosenfeld, A., Phillips, P.J.: Face Recognition: A Literature Survey. ACM Computing Surveys, 399–458 (2003)

[2] Kim, S., Chung, S.-T., Jung, S., Jeon, S., Kim, J., Cho, S.: Robust Face Recognition using AAM and Gabor Features. PWASET 21, 493–497 (2007)

[3] Mandal, T., Majumdar, A., Wu, Q.M.J.: Face Recognition by Curvelet Based Feature Extraction. In: Kamel, M.S., Campilho, A. (eds.) ICIAR 2007. LNCS, vol. 4633, pp. 806–817. Springer, Heidelberg (2007)

[4] Mandal, T., Wu, Q.M.J.: Face Recognition using Curvelet Based PCA. In: 19th international conference on Pattern Recognition, ICPR 2008, Tampa, FL, pp. 1–4 (2008)

[5] Candes, E.J., Demanet, L., Donoho, D.L., Ying, L.: Fast Discrete Curvelet Transforms, Technical Report, Cal Tech. (March 2006)

[6] Candes, E.J., Demanet, L.: Curvelet (August 24, 2007), `http://www.Curvelet.org`

[7] An archive of AT&T Laboratories Cambridge (2009), `http://www.cl.cam.ac.uk/Research/DTG/attarchive` (last visit December 12, 2009)

[8] Spacek, L.: Description of the Collection of Facial Images (June 2008), `http://cswww.essex.ac.uk/mv/allfaces/grimace.zip` (last visit December 12, 2009)

[9] Eriksson, B.: The Very Fast Curvelet Transform. SIAM, Mult. Model. Sim., 861–899 (March 2006)

[10] Safari, M., Harandi, M.T., Araabi, B.N.: A SVM-Based Method for Face Recognition Using A Wavelet PCA Representation of Faces. In: International Conference on Image Processing ICIP apos;04, October 24-27, vol. 2, pp. 853–856 (2004)

[11] Perlibakas, V.: Face Recognition Using Principal Component Analysis and Wavelet Packet Decomposition. Informatica 15(2), 243–250 (2004)

A Regulatory Model for Context-Aware Abstract Framework*

Juanita Pedraza[1], Miguel Á. Patricio[2], Agustín De Asís[1], and José M. Molina[2]

[1] Public Law Department
[2] Computer Science Department
Universidad Carlos III de Madrid
Colmenarejo, Spain
jpedraza@der-pu.uc3m.es, mpatrici@inf.uc3m.es,
aeasis@der-pu.uc3m.es, molina@ia.uc3m.es

Abstract. This paper presents a general framework to define a context aware application and analyzes social guarantees to be considered to develop this kind of applications following legal assumptions as privacy, human rights, etc. We present a review of legal issues in biometric user identification where several legal aspects have been developed in European Union regulation and a general framework to define context aware applications. As main result, paper presents a legal framework to be taken into account in any context-based application to ensure a harmonious and coherent system for the protection of fundamental rights.

Keywords: Context Applications, User Identification, Social Guarantees, Privacy and Human Rights.

1 Introduction

Nowadays it is increasingly important the development of reliable procedures that allow the secure access to new services, and the univocal identification of the user, key functionality in the scenarios of environmental intelligence and access control. The level of security given by the classic techniques based in the possession of an object (card) or an information (personal number), are surpassed by new techniques that work with measurable personal traits, both anatomic (fingerprints, iris, etc.) and behavioral (gait, key-stroking, etc.). At present, many research efforts are being made in developing new algorithms and techniques that allow to implement multi-biometric systems which combine different biometric traits to obtain a more secure and reliable identification.

Identification and personalization are essential features of context-based services. The development of efficient, non-vulnerable and non-intrusive biometric recognition

* This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, CAM CONTEXTS (S2009/TIC-1485) and DPS2008-07029-C02-02.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 397–406, 2010.

techniques is still an open issue in the biometrics field (in which, however, enormous scientific contributions have been made over the last decade). It is also a necessity to obtain contextual systems able to provide a satisfactory user experience.

The Biometric identification must be robust, efficient and quick process to be accepted in strong requirements of security in this networked society [1]. Biometrics aims to recognize a person through the physiological or behavioural attributes [2], such as iris, retina, fingerprints, DNA and so on. The cause for this increment in research fields is the security sector and the possible application in many its aspects, such as video-surveillance or access control.

The new proposals aim to approach Biometrics Recognition in an innovative manner, providing technological solutions which remove their current limitations, and integrating Biometrics Recognition in context inference and fusion activities. The contextual framework needs a biometric scheme with the following features:

- Multibiometric: which combines several sources of biometric information (traits, sensors, etc.) with the aim of mitigating the inherent limitations of each source, obtaining a more reliable and accurate system.
- Highly transparent, highly accepted, and low intrusive, using biometric traits that can be acquired even without any cooperation of the user (e.g. face, voice) and well socially accepted (like the handwritten signature)
- Able of inferring human activity and analyzing user emotions, therefore significantly focused on services customization.

These requirements affect directly to many legal aspects that should the considered before the development of industrial applications, to be used in the private sector or public sector.

In this work, authors define a set of procedures that should be contained in the context-aware applications to accomplish the legal aspect in Europe and USA related to privacy and human rights. Section 2 is center in a description of the e-passport and legal problems that surround it. In section 3, we present a general model to represent context-aware applications. A description of the legal issues to be considered in context-aware applications is enumerated in section 4. Finally, in section 5 some conclusions are included.

2 A Case Study of Legal Aspects in User Identification: E-Passports

A passport traditionally has three security requirements: (1) authenticity and integrity of the document (and its data), (2) match with the holder, and (3) authorization, depending on the situation of use. These requirements had been hardened, among other reasons, as a result of 9/11 attacks and the bombings in Madrid and London: the US, for instance, required that countries that wished to continue to participate in the visa waiver program need to provide their citizens with machine readable travel documents (MRTDs) with digital photographs.

The machine readable character of these biometric passports will be made by integrating a contactless chip into the document. The standards have been developed within the International Civil Aviation Organization (ICAO). ICAO has selected the face as biometric because it is simple to obtain, in a relatively non-intrusive manner, and is already widely used.

The EU requires an additional protection mechanism for fingerprints in accordance with the Council Regulation (EC) No 2252/2004 of 13 December 2004 on standards for security features and biometrics in passports and travel documents issued by Member States.

The principle of proportionality is a cornerstone of this legal norm: the biometric features in passports and travel documents shall only be used for verifying: (a) the authenticity of the document; (b) the identity of the holder by means of directly available comparable features when the passport or other travel documents are required to be produced by law [3].

However, this principle is insufficient: this regulation, in the light of the European data protection directive [4], has other critical subjects to consider, for instance [5][6]:

(1) Clear and legitimate objectives: the biometric data should not be processed further than for a specific and lawful purpose: if data are needed for a specific and legitimate purpose they can be used; if they are not needed for a well-defined purpose, personal data should not be used,
(2) Security: the process should be accurate and data should not be kept longer than necessary; the handling of relevant data should be done in a secure way, and the data should not be transferred to those countries that do not ensure data protection,
(3) Citizens rights: the procedures should be in accordance with data subject’s rights.

Additionally, biometric technology can be a potential threaten to dignity and other human rights; especially those that are leading us towards the ubiquitous Information Society, like context-aware applications, present fundamental challenges to notions of human rights, privacy and trust, especially in a context where governments need appropriate instruments to guarantee the security of the citizen, but they have to fully respect the citizen's fundamental rights.

3 Context Aware Applications: An Abstract Model

The more widely accepted and used definition of what context is, and it was given by Dey [7] where he defines context as: “any information that characterizes a situation related to the interaction between humans, applications, and the surrounding environment.” There are many developing systems such as platforms, frameworks and applications for offering context-aware services. The Context Toolkit proposed in [7] assist for instance developers by providing them with abstractions to build context-aware applications. The Context Fusion Networks [8] allows context-aware applications to select distributed data sources and compose them. The Context Fabric [9] is another toolkit which facilitates the development of privacy-sensitive, ubiquitous computing applications. There are previous approaches like Entree [10] which uses a knowledge base and case-based reasoning to recommend restaurant or for instance Cyberguide [11] project which provides user with context-aware information about the projects performed at GVU center in Atlanta with TV remote controllers throughout the building to detect users locations and provide them with a map that highlights the project demos available in the neighboring area of the user. A recent one is Appear which is a context-aware platform designed to provide contextual information to users in particular and well defined domains. It has a modular architecture and we have already used it in a previous work [12].

In Europe, the concept of Ambient Intelligent (AmI) includes the contextual information but expand this concept to the ambient surrounding the people. So, electronic or digital part of the ambience (devices) will often need to act intelligently on behalf of people. It is also associated to a society based on unobtrusive, often invisible interactions amongst people and computer-based services taking place in a global computing environment. Context and context-awareness are central issues to ambient intelligence [13]. AmI has also been recognized as a promising approach to tackle the problems in the domain of Assisted Living [14]. Ambient Assisted Living (AAL) born as an initiative from the European Union to emphasize the importance of addressing the needs of the ageing European population, which is growing every year as [15]. The program intends to extend the time the elderly can live in their home environment by increasing the autonomy of people and assisting them in carrying out their daily activities. Moreover, several prototypes encompass the functionalities mentioned above: Rentto et al. [16], in the Wireless Wellness Monitor project, have developed a prototype of a smart home that integrates the context information from health monitoring devices and the information from the home appliances. Becker et al. [17] describe the amiCa project which supports monitoring of daily liquid and food intakes, location tracking and fall detection. The PAUL (Personal Assistant Unit for Living) system from University of Kaiserslautern [18] collects signals from motion detectors, wall switches or body signals, and interprets them to assist the user in his daily life but also to monitor his health condition and to safeguard him. The data is interpreted using fuzzy logic, automata, pattern recognition and neural networks. It is a good example of the application of artificial intelligence to create proactive assistive environments. There are also several approaches with a distributed architecture like AMADE [19] that integrates an alert management system as well as automated identification, location and movement control systems.

All these approaches are promising applications from an engineering point off view, but, no legal aspects are considered in the development. Clearly, an important point is the necessity to identify the users of these systems. Two different approaches could be considered, one approach is based in the cooperation of the user to be identified and another one is based in the non-cooperative environment (for example in surveillance applications).

A generic framework of a Context-Aware Application consists of three layers as shown in Figure 1.

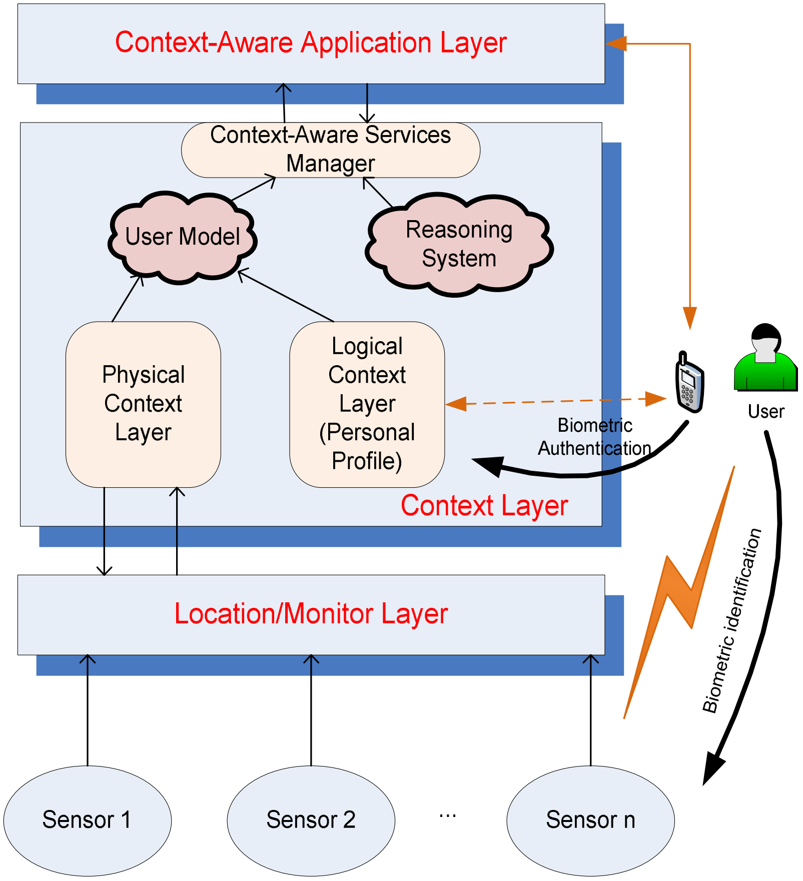


Fig. 1. Abstract Model of Context-Aware Applications

At the bottom of the Model is the Location/Monitor Layer, which is responsible for processing sensory information received by a collection of heterogeneous sensors into useful information for Context-Aware services. The set of sensors that monitor the activity of individuals are often organized in so-called sensor networks. The main role of the sensor networks in a Context-Aware environment are to provide to the Context-Aware services the answers to the following questions:

- *Who?* - Information on identifying individuals and objects of interest (animals, vehicles, etc.), i.e., the actors who are in a given scenario. One important aspect of this kind of information is the “Biometric Identification”. By automatic biometric identification systems, we mean those systems that rely on physical characteristics that are unique to each person to ascertain the identification of an individual.

- *Where and When?* - To provide a temporal framework of the action taken and its spatial location, to help to the establishment of the associations between objects and actors.
- *What?* - Information to help the recognition of activities, and the discovery of space-time relations between actors and objects.
- *Why?* - Provide information to discover associations in a particular action in a higher space-time level, in order to uncover plans and behavioral patterns.

With this sensory information, the Context Layer aims to answer the questions previously raised. Therefore, it is necessary to process and model information through the “Physical Context Manager” module. This module, in turn, is able to interact with the sensory layer in order to select certain preferences in the operation of the sensors.

A Context-Aware application should adapt its sensory information dynamically to the needs of users, taking into account a wide range of users and situations they may encounter. Through the “Logical Context Manager” module, the system is capable of adapting Context-Aware sensory information based on knowledge about their needs and characteristics, stored in what is called “Personal Profile”. This profile or logical context can be obtained directly through inputs by the user preferences, or by interaction with the environment observed from the sensory system. Obviously, in order to update the “Personal Profile”, the system must verify that the user is who he claims to be. In this sense, “Biometric Authentication” is the hardest authentication mechanism to forge.

With the merger of the logical and physical context information, the system is able to obtain the Context-Aware "User Model". The “User Model” plays a critical role in Context-Aware systems, since it embodies, on the one hand, the high-level semantic knowledge of actions of the user received from the sensory system; on the other, the user preferences, as well as its capabilities and limitations. Among these limitations, we may include information on their cognitive and sensorial level, or physical disabilities (for instance, elderly or handicapped people).

Finally, once established "User Model", the Context Layer has a "Reasoning System" module capable of inferring and accommodating the needs of services to the final users in the field of a specific Context-Aware application.

4 A Regulatory Model for the Use of Context-Aware Applications

Society as a whole needs to be aware of the obligations and rights those are applicable in relation to the use of context-aware applications. Therefore it makes sense to create a regulatory model for the use of context-aware applications [20][21][22][23].

A generic legal framework of a Context-Aware Application should be composed by principles and fundamental rules as shown in Figure 2.

This model should duly take into account:

a. Central axiological elements: The protection of human dignity, fundamental rights and in particular the protection of personal data, are the key issues of regulatory model.

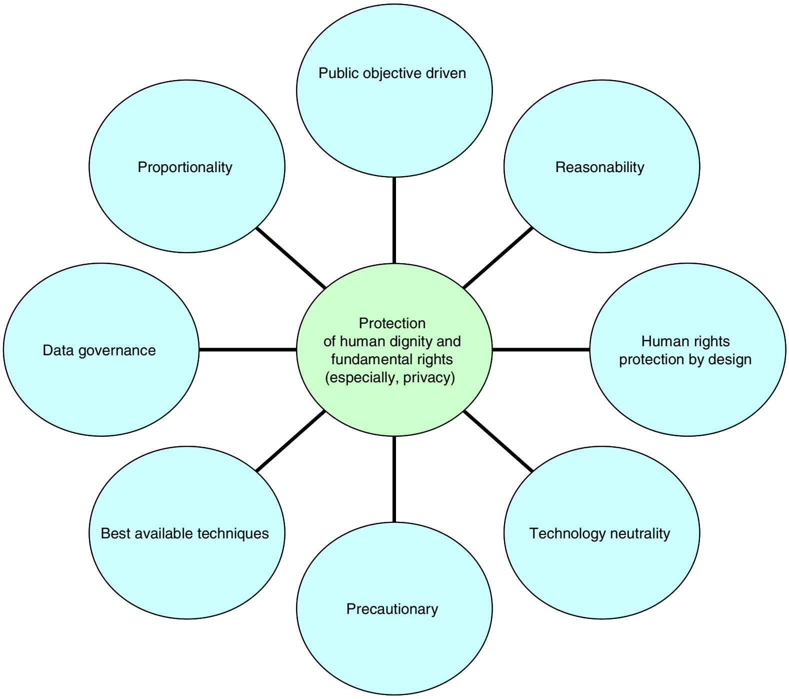

Public objective driven
Proportionality
Reasonability
Data governance
Protection of human dignity and fundamental rights (especially, privacy)
Human rights protection by design
Best available techniques
Technology neutrality
Precautionary

Fig. 2. Generic Legal Framework of a Context-Aware Application

b. Principles: This regulatory model and a range of implementing measures needs to be adopted to complete the legal framework, should duly take into account some general principles.

From our point of view, the general principles that should be taken into account could resume in the following ones:

1. Public objective driven vs. technology driven: the legal treatment for context-aware applications should not be 'technology-driven', in the sense that the almost limitless opportunities offered by new technologies should always be checked against relevant human rights protection principles and used only insofar as they comply with those principles [24].
2. Proportionality: requires that measures implemented should be appropriate for attaining the objective pursued and must not go beyond what is necessary to achieve it. The use of biometrics should not in principle be chosen if the objective can also be reached using other, less radical means.
3. Reasonability: reasonableness of a measure is therefore to be adjudged in the light of the nature and legal consequences of the relevant remedy and of the relevant rights and interests of all the persons concerned.

4. Data governance: is a useful principle that covers all legal, technical and organizational means by which organizations ensure full responsibility over the way in which data are handled, such as planning and control, use of sound technology, adequate training of staff, compliance audits, etc. [24]
5. Human rights protection by design: human rights protection requirements should be an integral part of all system development and should not just be seen as a necessary condition for the legality of a system [24].
6. Best Available Techniques: shall mean the most effective and advanced stage in the development of activities and their methods of operation which indicate the practical suitability of particular techniques for providing in principle the basis for ITS applications and systems to be compliant with Human rights protection requirements [24].
7. Precautionary: where there is scientific uncertainty as to the existence or extent of risks to human rights, the institutions may take protective measures without having to wait until the reality and seriousness of those risks become fully apparent.
8. Technology neutrality: regulatory framework must be flexible enough to cover all techniques that may be used to provide context-aware applications.

These principles should be considered in context aware applications to include legal requirements in analysis and design phases of software development, and, at the same time, national and international regulations should consider the new capacities of technology applied in these kind of systems.

5 Conclusions

In this paper, we present the necessity to consider legal aspect, related with privacy or human rights, into the development of the incipient context based services. Clearly, context based services and Ambient Intelligence (and the most promising work area in Europe that is Ambient Assisted Living, ALL) needs a great effort in research new identification procedures. These new procedures should be non-intrusive, non-cooperative, in order to the user be immersed in an Intelligent Environment that knows who is, where is and his/her preferences. These new paradigms should be development accomplished the legal issues to allow users be citizen maintaining their legal rights.

References

1. Jain, A.K., Bolle, R.M., Pankanti, S.: Biometrics: Personal Identification in a Net-worked Society. Kluwer, Norwell (1999)
2. Daugman, J.: Biometric Decision Landscape, Technique Report No. TR482, University of Cambridge Computer Laboratory (1999)
3. Schouten, B., Jacobs, B.: Biometrics and their use in e-passports. Image and Vision Computing 27, 305–312 (2009)

4. Directive 95/46/EC of the European Parliament and of the Council of October 24, 1995, on the protection of individuals with regard to the processing of personal data and on the free movement of such data OJ L 281, 23.11.1995, p. 31–50 and Directive 2002/58/EC of the European Parliament and of the Council of July 12, 2002, concerning the processing of personal data and the protection of privacy in the electronic communications sector OJ L 201, 31.7.2002, p. 37–47
5. Wrighta, D., Gutwirthb, S., Friedewaldc, M., De Hertb, P., Langheinrichd, M., Moscibrodab, A.: Privacy, trust and policy-making: Challenges and responses. Computer law & security review 25, 69–83 (2009)
6. Grijpink, J.: Biometrics and Privacy. Computer Law & Security Report 17(3) (2001)
7. Dey, A.K., Saber, D., Abowd, G.D.: A conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. Human-Computer Interaction (HCI) Journal 16, 97–166 (2001)
8. Chen, G., Kotz, D.: Context Aggregation and Dissemination in Ubiquitous Computing Systems. In: Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications, June 20-21, 2002, p. 105 (2002)
9. Hong, J.: The context fabric: An infrastructure for context-aware computing. In: Minneapolis, A.P. (ed.) Extended Abstracts of ACM Conference on Human Factors in Computing Systems (CHI 2002), pp. 554–555. ACM Press, Minneapolis (2002)
10. Burke, R., Hammond, K., Young, B.: Knowledge-based navigation of complex information spaces. In: Proceedings of the National Conference on Artificial Intelligence, pp. 462–468 (1996)
11. Abowd, G., Atkeson, C., Hong, J., Long, S., Kooper, R., Pinkerton, M.: Cyber-guide: A mobile context-aware tour guide. Wireless Networks 3(5), 421–433 (1997)
12. Sanchez-Pi, N., Fuentes, V., Carbo, J., Molina, J.: Knowledge-based system to define context in commercial applications. In: Proceedings of 8th International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing (SNPD), Qingdao, China (2007)
13. Schmidt, A.: Interactive context-aware systems interacting with ambient intelligence. IOS Press, Amsterdam (2005)
14. Emiliani, P., Stephanidis, C.: Universal access to ambient intelligence environments: Opportunities and challenges for people with disabilities. IBM Systems Journal 44(3), 605–619 (2005)
15. World population prospects: The 2006 revision and world urbanization prospects: The, revision. Technical report, Population Division of the Department of Economic and Social Affairs of the United Nations Secretariat (last access Saturday, February 28, 2009; 12:01:46 AM) (2006)
16. Rentto, K., Korhonen, I., Vaatanen, A., Pekkarinen, L., Tuomisto, T., Cluitmans, L., Lappalainen, R.: Users' preferences for ubiquitous computing applications at home. In: First European Symposium on Ambient Intelligence 2003, Veldhoven, The Netherlands (2003)
17. Becker, M., Werkman, E., Anastasopoulos, M., Kleinberger, T.: Approaching ambient intelligent home care system. In: Pervasive Health Conference and Workshops 2006, pp. 1–10 (2006)
18. Floeck, M., Litz, L.: Integration of home automation technology into an assisted living concept. Assisted Living Systems-Models, Architectures and Engineering Approaches (2007)
19. Fraile, J., Bajo, J., Corchado, J.: Amade: Developing a multi-agent architecture for home care environments. In: 7th Ibero-American Workshop in Multi-Agent Systems (2008)

20. Haas, E.: Back to the future? The use of biometrics, its impact on airport security, and how this technology should be governed. Journal of Air Law and Commerce (69), 459y ss (spring 2004)
21. Star, G.: Airport security technology: is the use of biometric identification technology valid under the Fourth Amendment? Law & Technology Journal (251) (2001-2002)
22. Luther, J.: Razonabilidad y dignidad humana. Revista de derecho constitucional europeo (7), 295–326 (2007)
23. Rodríguez de Santiago, J.M^{a}.: La ponderación de bienes e intereses en el Derecho Administrativo. Madrid (2000)
24. Opinion of the European Data Protection Supervisor on the Communication from the Commission to the European Parliament and the Council on an area of freedom, security and justice serving the citizen (2009/C 276/02) OJC 276/8 (November 17, 2009)

Ambient Intelligence Application Scenario for Collaborative e-Learning

Óscar García[1], Dante I. Tapia[2], Sara Rodríguez[2], and Juan M. Corchado[2]

[1] School of Telecommunications, university of Valladolid. Paseo de Belén 15,
47011 Valladolid, Spain
[2] Department of Computer Science and Automatic. University of Salamanca.
Plaza de la Merced, S/N, 37008, Salamanca, Spain
oscgar@tel.uva.es, {dantetapia,srg,corchado}@usal.es

Abstract. New learning methods are emerging in order to improve and facilitate the use of communication and informatics technologies both by students and professors. This paper describes an Ambient Intelligence application scenario based on Computer Supported Collaborative Learning and MANET in order to create creative and flexible e-learning environments where students can use wireless technologies for improving the leaning process.

Keywords: Ambient Intelligence, ad-hoc networking, CSCL, e-Learning, MANET.

1 Introduction

Advances in technology have surrounded people with a larger number of devices, so it is interesting to develop intuitive interfaces and systems to facilitate the people's daily activities [1], thus creating technological environments to be used in medical, public or education fields.

Ambient Intelligence (AmI) is an emerging multidisciplinary area based on ubiquitous computing, which influences the design of protocols, communications, systems, devices, etc., proposing new ways of interaction between people and technology, adapting them to the needs of individuals and their environment [3]. It offers a great potential to improve quality of life and simplify the use of technology by offering a wider range of personalized services and providing users with easier and more efficient ways to communicate and interact with other people and systems[3] [4]. Education is one of the fields where AmI techniques can benefit their users. New technologies, communication networks and devices provide new scenarios where students can learn through different, innovative and more efficient ways [5]. Within the variety of educational software we can find Computer Supported Collaborative Learning Applications (CSCL) [6]. In this kind of software participants make collaborative learning activities that require the interaction between them. The development of this kind of applications takes into account aspects of the development of Computer Supported Collaborative Work (CSCW), such as shifts, interfaces and content display problems in common working areas [7].

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 407–416, 2010.

However, the design and development process of AmI and e-learning systems is complex. This process includes the participation of technical and educational people, not only to get the requirements of applications, but also the final users which are the final ones. It is also necessary to adapt these applications to specific educational and social needs [8].

The mixture of ideas related to Ambient Intelligence and CSCL offers new possibilities in education. However it brings difficulties that have to be solved as it implies a research effort in an emerging computing science area known as Mobile Computer Supported Collaborative Learning (MCSCL). There are several topics in this area which include the study of educational characteristics of new scenarios, new architectures, technological requirements, as well as the evaluation of proposed scenarios [9]. One of these new scenarios includes the dynamic interconnection and organization of mobile nodes in arbitrary topology [10]. These kinds of networks are also known as MANET (Mobile Ad hoc NETwoks) and are a helpful communication way for students because it is not needed to have a previous infrastructure to get the interconnection of the devices.

This paper explores these areas in order to find a valid AmI scenario based on CSCL and using MANETs, emphasizing the student-student and student-professor interactions in an informal e-learning environment. In this scenario, the professor is able to know what is happening in each network formed every time. Students can receive information from wireless sensors and also collaborate each other by using mobile devices. Specifically, this work tries to reach a MCSL scenario. From a pedagogical point of view, there is a set of abilities and motivations necessary in order to reach a situation of MANET collaboration [11]. Such networks have a great advantage since they can cover groups with different sizes. However, there are requirements that must be taken into account for accomplishing the collaboration objectives: they must facilitate the communication process, social interactions and the awareness of the network. The environment in which MANETs are used will be decisive when designing the applications and the functionalities they must provide [12]. In addition, requirements such as cultural, economical, technological (user interface, functionalities, awareness, adaptability, reliability and maintenance, efficiency, connectivity, etc.) and those regarding security must be considered when creating a MANET in learning applications [13].

This paper is structured as follows: Section 2 introduces the use of MANET in CSCL environments. Section 3 presents the problem description and analyzes several considerations that help to define the proposed scenario. Section 3.1 describes the proposed scenario. Finally, Section 4 presents the results and conclusions obtained.

2 Mobile Ad-Hoc Networks in CSCL Environments

One of the most important disciplines that are part of AmI paradigm is ubiquitous computing and communication [15]. AmI is described as a model of interaction in which people are surrounded by an intelligent environment, aware of his presence, context sensitive and adaptable to your needs through embedded, transparent and non-invasive technology [16]. AmI based systems will improve the quality of life of users, offering easier and more efficient services and communication ways to interact with other users and systems.

Education is one of the areas where Ambient Intelligence is useful by providing ubiquitous communication forms that facilitate learning and collaboration among individuals. Moreover, new advances in technology have meant students and professors to be surrounded by a larger number of devices, so it exists an opportunity to develop intuitive interfaces and systems to facilitate the activities of individuals and the learning process, in a discreet and transparent way, considering students and professors the most important factor in the design and development process [14].

There is currently a strong interest in educational software, usually known as e-learning [17]. The process of creating the applications that support these kind of learning is complex as it not only requires the participation of technical staff working together to design, implement and test the software. The educational component must be present in the design process, more over because professor and students will be the users and the applications must be adapted to specific social and educational needs [18].

Nowadays there are many different mobile devices available to students. One of their main features is its ability to interconnect between them. Usually mobile devices are providing with Wi-Fi, Bluetooth, ZigBee, GSM/GPRS, GPS, etc. communication capabilities. These capabilities are useful in the education field as they give rise to the construction of ubiquitous communication environments where students can collaborate [19]. Collaborative networks creation in any place without a previous infrastructure, commonly known as ad hoc networking, makes easier the informal learning processes supported by mobile devices.

All these features offer new and desirable possibilities, but require problems to be solved. MCSCL tries to propose solutions to all these tasks. In this field there are many proposals and publications, ranging from the study of educational characteristics of the new scenarios, technological requirements and architectures proposals to the proposal and evaluation of specific scenarios [20].

The nature of the network architectures, applications, communication ways and activities proposed is very broad, with significant infrastructural and educational differences, as well as functionalities. Thus, it is usual to find applications in which resources (e.g. files, applications, forms, etc.) are shared through spontaneous networks [21]. Nevertheless, the creation of the MANET is very different in each development. However, the easiest way to create a network is to have a previous infrastructure, by means of either a wireless LAN deployed for this purpose, mobile nodes intended for providing the resources [22], or using nodes that create the network and register other nodes that are part of it [23]. On the other hand, the communication amongst devices is a key aspect in this kind of networks [22]. In order to solve network formation issues, it is usual to use master nodes, that is, network devices responsible for making up the MANET and allowing other nodes to access to it [23]. In such cases it is followed a client-server model in which the server is also used for providing contents and services to the rest of the nodes. It is important to mention that not always nodes have total freedom to move in the network. So, the mobility of the nodes and the way the teacher can supervise the work must be taken into account. If a previous network infrastructure is needed, the nodes must be under the coverage of the network, whether formed by mobile [22] or fixed nodes [21]. The most interesting cases for the scenario presented in this paper are those in which there is a mobile node acting as master. This way, teachers have at their disposal a terminal acting in this way and allowing them to supervise the connections established, who have registered in the network and even each of the interactions and the utilization of resources or applications made by each node [22] [23].

The purpose of this paper focuses on exposing a CSCL AmI scenario in which collaboration between students is conducted through unplanned ad-hoc networks Each student and educator will be able to create a network which provides resources, activities or a communication channel that foster collaboration or get students to collect information, for example, everywhere and every time, getting an ubiquitous communication scenario in a natural way that will not require to plan any of the enumerated aspects before. These characteristics imply the need of an actor that will offer the shared resources over the network created. Communication, resources access or data interchange must be organized and controlled ever we have a peer to peer communication model. This way it will exists a master role that will allow a node to act as the master device, super-peer or super-node if it is able to collect information from the other nodes. Ad-hoc communication protocols will be adapted to the applications, in order to improve neighbor discovering, connection establishment and heterogeneous devices interconnection. In this field, *superpeers* functioning will be designed and developed to reach a pure MANET infrastructure in the applications.

3 Proposed Scenario

As could be seen in the previous section, there are multiple factors that characterize the use of Ambient Intelligence, CSCL and MANETs in e-learning environments. Thus, it is convenient to limit the scope. The following six questions and answers will help to delimit the problem:

1. *Where is the learning?* It is usual to find two kinds of learning: formal (inside the classroom) or informal (outdoors). The most common situation is the first one, while situations relative to outdoors learning usually describe lessons organized out of the class [22]. The latter ones have a remote access to the application (as an adaptation of e-learning platforms) or learning examples in museums and through tourist travels [16].

2. *Where are the actors?* Many applications that use MANETs are designed to be used for facing situations. However, there are very few situations that handle remote learning. These include the classroom and usually the home of the participants. Thus, combined learning is not easy to find in the literature. This kind of learning is the most interesting for the proposed scenario because it is very interesting to forted spontaneous collaboration everywhere (e.g. classroom, home, park, etc.) [21].

3. *Which are the most usually interactions between participants?* In this case, student-student and student-professor interactions are equally considered. The most interesting scenario is that where both interactions are considered, so collaboration between students is fostered and professors can provide support in real time and know the status of interactions and communications in the specific network.

4. *What kind of architecture is adopted?* This is of vital importance in MANETs. A pure MANET should not depend on a central node that provides the connection, while a peer-to-peer infrastructure does not create an adequate interaction between all the

participants [23]. For this reason, cases that describe hybrid architectures are the most useful for our purpose [18]. These architectures include control nodes that manage communication and information exchange between the nodes, usually called *super-peers* [12].

5. *Is a previous infrastructure used?* In order to get a pure MANET, there must not be a previous infrastructure [23].

6. *What communication technologies are used?* The communication technologies employed are often associated with the used devices. For example, mobile phones are associated with GSM, GPRS, SMS, or Bluetooth communications, while PDA and laptops usually use Bluetooth, ZigBee or Wi-Fi protocols.

3.1 Scenario Description

After considering issues that affect the use of MANETs, this subsection describes an application scenario. First, requirements that must be met to the scene are listed, and then the main features, functionalities and the particular case of use are described.

The proposed scenario is focused on a university scope. Nowadays, much of the students of a faculty have mobile devices, in particular laptops, PDA or mobile phones, which presents an opportunity to use these devices in the e-learning process [19]. The main objectives set for this scenario are:

- There must be a deep collaboration between actors. Such collaboration is set in formal and informal environments. Learning will not be restricted to the classroom and the activities proposed by the teacher, but informal learning must be present too. In this way, learning can occur in other places in or out of the faculty when two students with the same subject meet.
- Professor must know the status of the activities. Support will be given to teachers to control the activities they offer. In this way they'll can know and manage the formation of the groups, as well as the activity raised to each of them.
- Professor must know the existing interactions. It is interesting from the professor's point of view, to know the interactions between different students, either in the classroom or in the coverage area. This should take into account how the networks formed without the supervision of the teacher store information of their status.
- Information must be shared, both text (files) and multimedia (audio, video), even collected from some placed where the activity will be carry on.

From these objectives, it is possible to define the following technological requirements of this scenario:

- To make learning in a formal and informal way, mobile devices must be used.
- Collaboration and informal learning will be supported by the formation of the MANET.
- To achieve an unrestricted informal learning we must use a non-centralized architecture, so there is total independence from a previous network infrastructure because of the communication is peer-to-peer.

- To achieve greater communication levels and to require a minimal network infrastructure on the user side, it must be used wireless communication between devices.
- It will be used master nodes or *superpeers* in order to provide the teachers with detailed information about the activities or interactions that have appeared in created networks. *Supernodes* must be able to collect what has been happening in all networks they have been involved.

To illustrate the previous statements, the next example is presented. Imagine a collaborative activity in the subject of urbanism at the Faculty of Architecture. The development of a partial urban plan involves an action on a specific city area without forget the status and contents on the rest of city areas, both neighboring and distant ones. There are items such as green areas, public spaces, rail or public transport infrastructures, etc. affecting equally all areas.

From the student side, the description and goals of the activities are: the class is divided into several groups of students, the same number as the zoning of the city. Each group analyses its area in depth collecting all the necessary town-planning information (topology, constructions, infrastructures, railways, public areas, private areas, etc.), storing it into data files, photos or videos. Each participant may form MANETs with peers to share information, exchange messages to organize, raise questions, complete, in a collaborative way, a concept map to reflect the organization of the area and its needs or another collaborative activity that may be raised. Furthermore, it is necessary the collaboration with colleagues from other areas since the raised intervention will be the same for the whole city, so it should be taken into account the information of the others. On the other hand, students can get information in public places (city council, land registry) which can then be spread by all participants. On the other hand, teachers can launch activities (i.e. create a new MANET) or interfere in other MANETs for observing the work status and guiding students in their interactions. Similarly, the teacher may solve the questions raised by the students. It is interesting to set some partial milestones and a final one, so that the teacher collects results, can assess the learning process and can pick up the interactions to act accordingly (reconfiguring groups, modifying the activity, making a general explanation, etc.).

Figure 1 shows the first phase of the activity schema: there is an imaginary city divided into zones where different groups are working, whose participants will form intra-group and extra-group MANETs. The figure shows intra-group in red, blue and yellow and arrows mean that each member can move from a zone to other one and form a new ad hoc network with any of the participants from other groups. The teacher can move freely through all of them observing and collecting data. In this way, the scenario presents informal learning and students are able to realize that collaborating may achieve the goals in an easier and more effective manner. This case presents certain similarities with solutions for activities in museums [16] or with activities raised for outdoor learning [22]. However, this scenario raises an unplanned network type, in which students' collaboration is totally free, with possibility of intra and extra-groups communication. Moreover, it is not only considered the student-student interaction and the data collection by teachers [23] or the teacher-student interaction to solve doubts or launch the activity and configure groups [26]. In addition, the teacher is involved in the whole process, being able to be part of any of the networks, resolving doubts and, what we consider most important, collecting information about occurred interactions and partial results to help him to reconfigure the activity if necessary. Figure 2 represents the second phase of the activity. Students can share their information in class with other group members and reach through collaboration the objective planned by the professor, that is able to get the information from each network freely formed and change the activity in order to better organize the activity to foster the expected results.

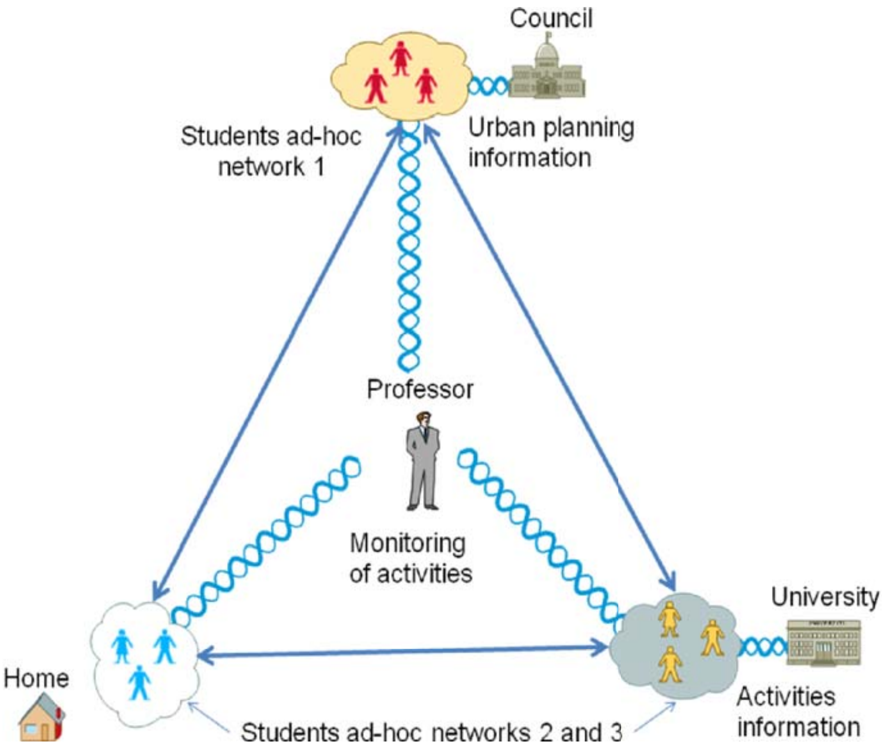

Council
Students ad-hoc network 1
Urban planning information
Professor
Monitoring of activities
University
Home
Activities information
Students ad-hoc networks 2 and 3

Fig. 1. Example of collaborative learning activity using MANETs outdoors. First phase of full activity.

The communication between devices is another aspect of this scenario. These are unplanned MANETs with communication between peer-to-peer participants, in which there are control nodes or *superpeers* to help their training, organization and data collection. Previous works take into account existing infrastructures to form these networks [9] and other uses networks without infrastructure [23], but they do not take into account interactions between participants neither the role played by the teacher. In the scenario presented collect information is considered as one of the most important parts, so professor will be able to know what is happening and take decisions dynamically improving the learning process.

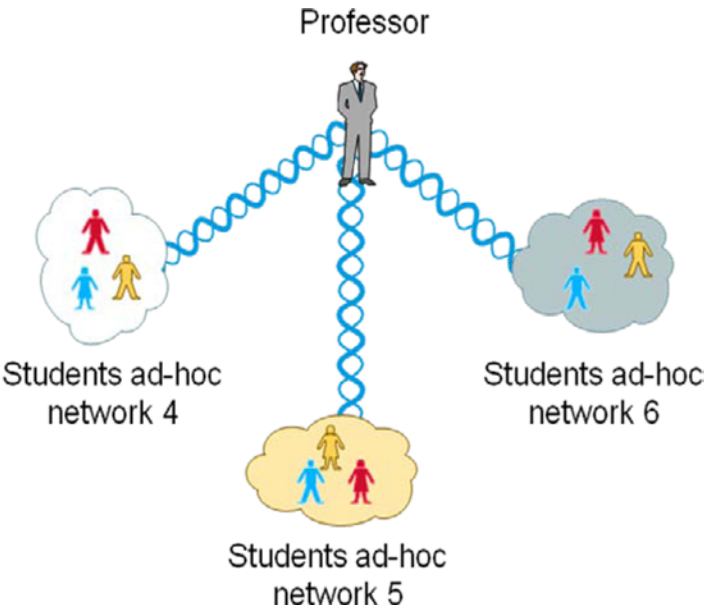

Professor
Students ad-hoc network 4
Students ad-hoc network 6
Students ad-hoc network 5

Fig. 2. Example of collaborative learning activity using MANETs in a classroom. Second phase of full activity.

4 Conclusion and Future Work

New learning methods are emerging in order to improve and the use of communication and informatics technologies both by students and professors. This paper presents an Ambient Intelligence scenario in education. In order to describe a specific workplace environment, its main characteristics are obtained from AmI, CSCL and MANET paradigms.

Features extracted are the key to get learning objectives as foster collaboration between students and between students and professors, reach an informal and blended learning scenario or have knowledge every time about the interactions and the activities raised. If MCSCL applications use effectively MANET features, it appears a new research opportunity in learning. Nowadays learning can happens anytime and anywhere, in a collaborative way using mobile devices owned by the students.

It is very important to understand what is offered by current wireless technologies and decide which of them are more representative in the devices commonly used in educational devices. It is very important to study and understand peer-to-peer communication, specially those ones that use superpeers, because this kind of nodes allow us to get pure ad hoc networks infrastructures and facilitates the collection of information of what is happening in the network or the network discovering.

The scenario presented in this paper is only the beginning of a new project. Future work implies that applications and activity design needs a deep analysis. Differences in hardware and software in the spectrum of devices we use will have influence in decisions like visualization schema, communication rules or data collection.

Acknowledgement

This project has been supported by the Spanish Ministry of Science and Technology project OVAMAH: TIN 2009-13839-C03-03.

References

1. Ducatel, K., Bogdanowicz, M., Scapolo, F., Leijten, J., Burgelman, J.C.: Scenarios for ambient intelligence in 2010. In: Information Society Technologies Advisory Group, ISTAG (2001) (Draft Final Report)
2. Schmidt, A.: Interactive context-aware systems interacting with ambient intelligence. In: Ambient Intelligence. IOS Press, Amsterdam (2005)
3. Weber, W., Rabaey, J.M., Aarts, E.: Ambient Intelligence. Springer, New York (2005)
4. Corchado, J.M., Bajo, J., De Paz, Y., Tapia, D.I.: Intelligent Environment for Monitoring Alzheimer Patients, Agent Technology for Health Care. In: Decision Support Systems. Eslevier, Amsterdam (in press, 2008)
5. European Commission. Application scenarios, next generation collaborative working viroments 2005-2010. 2nd. Report of the expert group on collaboration @ work, Bruselas (2004)
6. Koschmann, T.: CSCL: Theory and Practice of an emerging paradigm. Lawrence Eribaum Associates, Mahwah (1996)
7. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware: Some issues and experiences. Communications of the ACM 34(1), 38–58 (1991)
8. Dimitriadis, Y., Asensio-Pérez, J.I., Gómez-Sánchez, E., Martínez-Monés, A., Bote-Lorenzo, M.L., Vega-Gorgojo, G., Vaquero-González, L.M.: Middleware para cscl: Marco de componentes software y apoyo de tecnología grid. Inteligencia Artificial 7(23), 21–31 (2004)
9. Zurita, G., Nussbaum, M.: A constructivism collaborative learning environment supported by wireless interconected handhelds. Journal of Computer Assisted Learning 20(4), 235–243 (2004)
10. Ilyas, M.: The Handbook of Ad Hoc Wireless Networks. CRC Press, Florida Atlantic University, Boca Raton, Florida (2003)
11. Frohberg, D., Schwabe, G.: Skills and motivation in ad-hoc collaboration. In: CollECTeR: Collaborative Electronic Commerce Technology and Research, Basel, pp. 157–172 (2006)
12. Chang, C.Y., Sheu, J.P., Chan, T.W.: Concept and design of ad hoc and mobile classrooms. Journal of Computer Assisted Learning 19, 336–346 (2003)
13. Economides, A.A.: Requirements of mobile learning applications. International Journal of Innovation and Learning 5(5), 457–479 (2008)
14. Schmidt, A.: Interactive context-aware systems interacting with ambient intelligence. In: Ambient Intelligence. IOS Press, Amsterdam (2005)
15. Weiser, M.: The computer for the 21st century. Pervasive Computing, Scientific American, 99–100 (1991)
16. Simarro, J., Muñoz, H., Stoica, A.G., Avouris, N., Dimitriadis, Y., Fiotakis, G., Liveri, K.D.: Mystery in the museum: collaborative learning activities using handheld devices. In: Proceedings of the 7th international conference on Human-Computer interaction with mobile devices and services, pp. 315–318 (2005)

17. Vasiliou, A., Economides, A.A.: MANET-based outdoor collaborative learning. In: Auer, M.E., Al-Zoubi, A.Y. (eds.) Proceedings of the 3rd International Conference on Interactive Mobile and Computer Aided Learning (IMCL), Amman, Jordan, April 16-18 (2008)
18. Echevarría, S., Santelices, R., Nussbaum, M.: Comparative analysis of adhoc networks oriented to collaborative activities. LNCS, pp. 465–479. Springer, Heidelberg (2006)
19. Zurita, G., Nussbaum, M.: A constructivism collaborative learning environment supported by wireless interconected handhelds. Journal of Computer Assisted Learning 20(4), 235–243 (2004)
20. Naismith, L., Lonsdale, P., Vavoula, G., Sharples, M.: Literature Review in Mobile Technologies and Learning, In: NESTA (National Endowment for Science Technology and the Arts), Bristol, UK (2004)
21. Fuller, A., McFarlane, P., Saffioti, D.F.: Distributed, collaborative learning environments using ad hoc networks. In: Proceedings of the IEEE International Conference on Advanced Learning Technologies, pp. 705–707 (2004)
22. Vasiliou, A., Economides, A.A.: Mobile collaborative learning using multicast manets. International Journal of Mobile Communications 5(4), 423–444 (2007)
23. Zurita, G., Nussbaum, M.: An ad-hoc wireless network architecture for face-to-face mobile collaborative applications. In: Grass, W., Sick, B., Waldschmidt, K. (eds.) ARCS 2006. LNCS, vol. 3894, pp. 42–55. Springer, Heidelberg (2006)
24. Taylor, J., Peake, L., Phillip, D., Robertshaw, S.: Location activated nomadic discovery (land): A mobile location-sensitive information and navigation service for cumbria. In: Proceedings of the European Workshop on Mobile and Contextual Learning, pp. 21–22 (2002)
25. Cortez, C., Nussbaum, M., Santelices, R., Rodríguez, P., Zurita, G., Correa, M., Cautivo, R.: Teaching science with mobile computer supported collaborative learning (mcscl). In: Proceedings of the Seconds IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE04), pp. 67–74. IEEE Computer Society Press, Los Alamitos (2004)

Strategies for Inference Mechanism of Conditional Random Fields for Multiple-Resident Activity Recognition in a Smart Home

Kuo-Chung Hsu, Yi-Ting Chiang, Gu-Yang Lin, Ching-Hu Lu,
Jane Yung-Jen Hsu, and Li-Chen Fu

Department of Computer Science and Information Engineering
National Taiwan University
Taipei, Taiwan
b93201043@ntu.edu.tw, d94021@csie.ntu.edu.tw,
r97131@csie.ntu.edu.tw, jhluh@ieee.org,
yjhsu@csie.ntu.edu.tw, lichen@ntu.edu.tw

Abstract. Multiple-resident activity recognition is a major challenge for building a smart-home system. In this paper, conditional random fields (CRFs) are chosen as our activity recognition models for overcoming this challenge. We evaluate our proposed approach with several strategies, including conditional random field with iterative inference and the one with decomposition inference, to enhance the commonly used CRFs so that they can be applied to a multiple-resident environment. We use the multi-resident CASAS data collected at WSU (Washington State University) to validate these strategies. The results show that data association of non-obstructive sensor data is of vital importance to improve the performance of activity recognition in a multiple-resident environment. Furthermore, the study also suggests that human interaction be taken into consideration for further accuracy improvement.

Keywords: Smart home, activity recognition, multiple residents, conditional random field (CRF), data association.

1 Introduction

Activity recognition, a mechanism in ubiquitous computing to facilitate the classification of human activities, is a key enabler to many context-aware applications in a smart home. For example, in order to target maximum comfort for residents, a smart-home system should be able to recognize their activities and to provide appropriate services whenever necessary (e.g. switching on a light if a resident is studying while the environment is dim). Another important application is healthcare; activity recognition can provide instant assistance for caregivers in the event of emergencies (e.g. diagnosing an abnormality of a resident and triggering an alarm if the resident sleeps much longer than that s/he used to do). Due to these benefits, many researchers have focused on how to efficiently sense the activity of residents.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 417–426, 2010.

Numerous approaches to multiple-resident activity recognition in a house scale have been proposed in the literature. Generally speaking, these approaches include asking residents to carry sensors[1](e.g., RFIDs, infra-red receivers), installing cameras in the environment[2-3], and deploying pervasive sensors[4-5](e.g., pressure sensors, reed switches) for later activity reasoning. What seems to be lacking, however, is the human-centric concerns since a smart home system, unlike a public space, should not ignore many factors regarding ergonomic concerns. For example, the camera-based solutions may invade residents' privacy concerns, and wearable sensors usually cause inconvenience, both of which make the sensors not suitable for a smart-home environment targeting maximum comfort. Therefore, Lu *et al.* categorize deployed devices into seamless and seamful types to take residents' ergonomic concerns into account, and they suggest making use of as much seamless devices as possible for a smart-home environment [6]. Accordingly, selecting suitable devices is nontrivial and using non-obtrusive and pervasive sensors for sensing information is suggested from a smart-home environment.

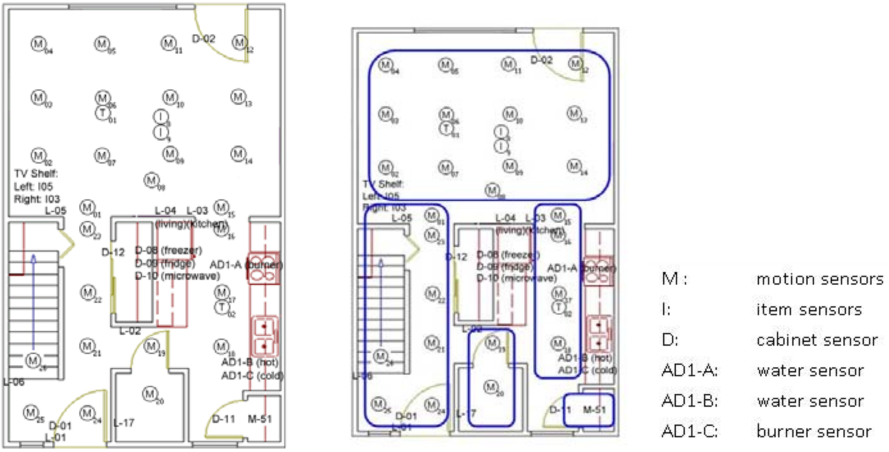
Existing studies regarding non-obtrusive activity recognition primarily concentrate on an environment involving only a single resident [7-9]. However, in a multiple-resident environment, the high complexity of tracking plural residents and the difficulty in modeling interpersonal interactions make multi-user activity recognition intractable. Among these challenges, the data association problem using non-obstructive sensors is of vital importance.

The data association problem is defined as resolving by whom each sensor data are triggered. As a result of insufficient clues, much research addresses this problem using various filters. For example, Wilson and Atkeson propose a system for simultaneous tracking and recognizing the state "active" or "inactive" of multiple people[4]. In this paper, we propose and compare several approaches to resolving the data association for recognizing 15 activities which are frequently preformed of multiple residents. The inferred activities, if reliable, can be a strong clue to many applications. The automation and healthcare may be practical under a multiple-resident environment.

The paper is organized as follows: Section 2 provides an overview of a dataset used to verify our approach. In the section 3, our problem definition and several strategies to train multiple-resident activity models are presented. Then, the experimental results and the comparison among various models are shown in Section 4. Finally, we summarize this paper in Section 5.

2 CASAS Dataset

The dataset, "WSU Apartment Testbed, ADL Multiresident Activities", is collected in the CASAS project in the WSU smart workplace [10]. There are 26 files of different participant pairs containing sensor events and their corresponding annotations whereby we utilize to validate our proposed models. There are about 400 to 800 sensor events in each file, totally 17,238 events in this dataset. The data represent two residents being asked to perform 15 activities of daily living (ADLs), which are shown in Table 1. Of these activities, some are individual activities, which are performed by one resident and which are independent to the activities performed by other residents. On the other hand, cooperative activities are performed by one or more residents, and multiple activities are dependent. The deployed sensors include motion sensors, item sensors and cabinet sensors (see Fig. 1.(a)). The output format of a sensor event and its annotation is (*Date, Time, SensorID, Value, ResidentID, TaskID*). The detail of the dataset can be found in [5].



(a) (b)

Fig. 1. The sensor deployment

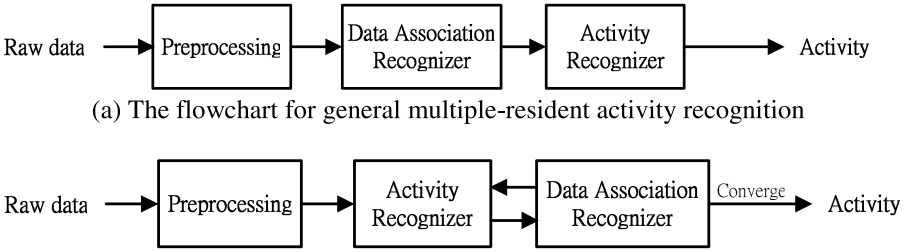
Table 1. The 15 activities asked to perform by residents in CASAS project

	individual	*Cooperative*
Activity	*Filling medication dispenser*	*Moving furniture*
	Hanging up clothes	*Playing checkers*
	Reading magazine	*Paying bills*
	Sweeping floor	*Gathering and packing picnic food*
	Setting the table	
	Watering plants	
	Preparing dinner	

3 Multiple-Resident Activity Recognition

The prior proposed approaches generally follow the following procedures for recognizing activities of multiple residents. First, sensor data are preprocessed and then transformed into feature vectors whereby the parameters of activity models are learnt. The transformed features are used to determine the data association variables and the determined variables can assist to infer activities. Fig. 2(a) shows the flowchart for the above-noted general multiple-resident activity recognition.

For describing our models, several issues should be addressed. The first issue is the problem we intend to handle and the notation definitions. The second is how to preprocess sensor data. The third is the material of conditional random field (CRF) applied to resolve multiple-resident activity recognition. The fourth is the mechanism to



(a) The flowchart for general multiple-resident activity recognition

(b) The flowchart for iterative training data association and activity recognition

Fig. 2. The flow chart of multiple-resident activity recognition

iteratively gain the data association variables and the activities. The last is the decomposition inference given the data association.

3.1 Problem Formulation

The associated notations are introduced here in recognizing activities of two residents. Given n observations $O_{1:n}$ with each observation O_t =($Date_t$, $Time_t$, $SensorID_t$, $Value_t$) at time t, let the preformed activities be $A_{1:n}$ and the data association sequence be $\theta_{1:n}$, where A_t=1,…,15 (each one is an index to an activity of interest) and $\theta_{1:n}$=1,2 which indicates that the observation O_t is triggered by the θ_t-th resident. To be unambiguous, activity A_t is defined to be performed by the θ_t-th resident at time t. The purpose of multiple-resident activity recognition is to determine $A_{1:n}$ and $\theta_{1:n}$ given $O_{1:n}$ without prior knowledge between $A_{1:n}$ and $\theta_{1:n}$.

3.2 Data Preprocessing

For each time slice, the preprocessed data are fed into the recognizer. Three types of preprocessed data are concerned: raw data, environment data and room-level data. The raw data refer to the data with the date and time removed from O_t. The environment data contain all sensed information in the house by continuously renewing the status of each sensor. For example, five sensors exist and the environment data at time t-1 is (0,1,0,0,1). At time t, a sensor event is triggered "on" by the third sensor. This event updates the environment data to (0,1,1,0,1) at time t. The room-level data are collected by manipulating the environment data and using domain knowledge to divide the whole experimental environment into several room-scale regions (see Fig. 1(b)) and then representing each room by a feature which is "on" if and only if one of the motion sensors in the room is "on".

3.3 Conditional Random Field (CRF)

In this paper, we apply conditional random fields (CRFs) [11] for multiple-resident activity recognition since they are discriminative, undirected graphical models which can be applied to sequence labeling problems. Given a sequence of observations, we can define feature functions in CRFs to model the dependency between two consecutive states in the problem domain. Let the activities we want to recognize be the states and the collected sensor data to be the observations; we can apply CRFs on activity recognition. It has been shown that CRFs perform well on activity recognition problems [7, 12-13].

Let X and Y be two set of random variables representing the observations and the states respectively. We can define an undirected graph $G(V,E)$ such that V corresponds to X and Y, and E represents the relationship (in the form of a feature function) between these vertices (random variables). If C is the set of cliques in G, in conditional random fields, we can define a condition probability $P(Y \mid X) = \exp \sum_{c \in C} \lambda_c f_c(Y_c, X_c) / Z(X)$, where Y_c and X_c are random variables in a clique c in G, $Z(X)$ is a normalization factor, f_i's are the feature functions, and λ_i's are the parameters of this CRF. For example, Fig. 3 is a typical linear-chain conditional random field (LCRF) model. $X=\{X_1,...,X_n\}$ is a observation sequence, and $Y=\{Y_1,...,Y_n\}$ is a state sequence. In this CRF model, there are two kinds of feature functions: f_t's for transition model and g_t's for observation model. The condition probability in this CRF model is

$$P(Y \mid X) = \frac{1}{Z(X)} \exp\left\{ \sum_{t=1}^{n-1} \theta_t f_t(y_t, y_{t+1}) + \sum_{t=1}^{n} \lambda_t g_t(x_t, y_t) \right\}$$

where $\Theta = \{\theta_t\}_{t=1}^{n-1}$ and $\Lambda = \{\lambda_t\}_{t=1}^{n}$ are the parameters to be learned from training data.

Optimization methods like conjugate gradient [14] and improved iterative scaling [15] can be used to train CRFs. The most commonly used method is an improved quasi-Newton method called L-BFGS[16].

3.4 Conditional Random Field with Iterative Inference

Generally, the multiple-resident activity recognition follows the procedure shown in Fig. 2(a). However, we observe that an estimated activity can be helpful and informative for data association recognizer and vice versa. Hence, a reciprocal procedure is designed to sufficiently exploit the closeness of the relationship between the two kinds of information (See Fig. 2(b)). The graphical representation of the CRF we propose is shown in Fig. 4, which associates the activities A_t with the data association

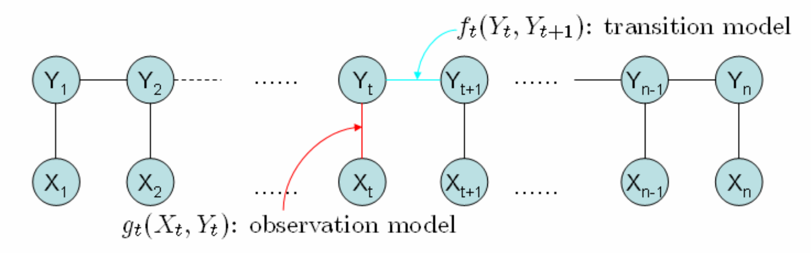

$f_t(Y_t, Y_{t+1})$: transition model
Y1
Y2
Yt
Yt+1
Yn-1
Yn
X1
X2
Xt
Xt+1
Xn-1
Xn
$g_t(X_t, Y_t)$: observation model

Fig. 3. The graphical representation of the linear-chain conditional random field

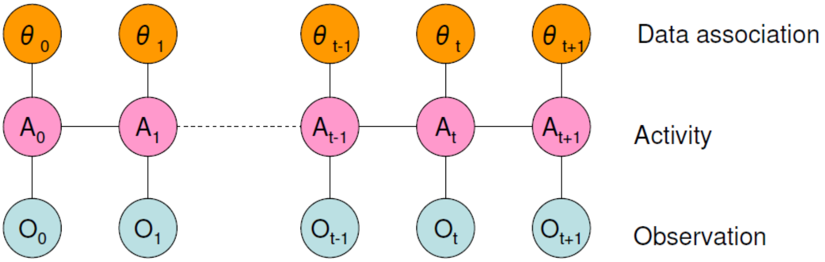


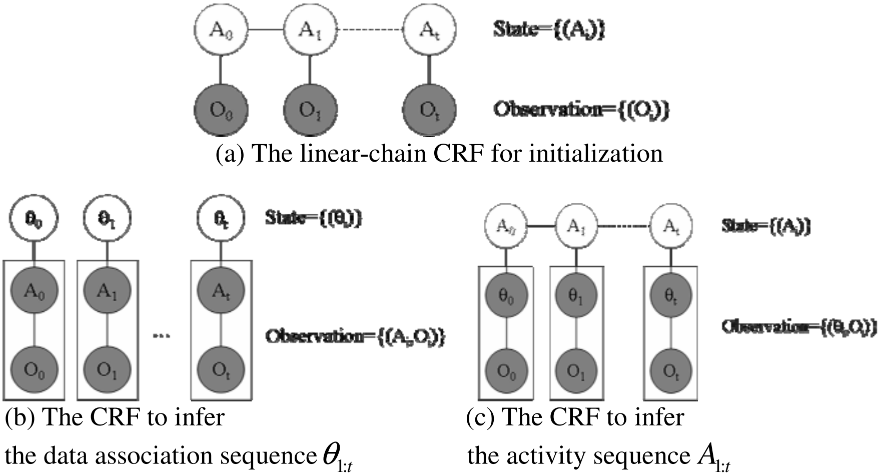
Fig. 4. The graphical representation of the CRF for multiple-resident activity recognition

variable θ_t and the observation O_t. For iteratively estimating activity and data association based on the reciprocal procedure, three variant CRFs are utilized as shown in Fig. 5. The first model is designed to initialize the activity sequence $A_{1:t}$ (Fig. 5(a)), the second is designed to infer the data association sequence $\theta_{1:t}$ given $O_{1:t}$ and $A_{1:t}$ (Fig. 5(b)), and the third is designed to infer the activity sequence $A_{1:t}$ by treating $O_{1:t}$ and $\theta_{1:t}$ as observations (Fig. 5(c)). Among these structures of CRFs, the first and the third one are in the form of linear-chain CRFs (LCRFs).

After the models are learned, the preprocessed data are used to generate $A_{1:t}^{(0)}$, and then iteratively infer the data association variables and the activities. The inference algorithm is shown in Algorithm 1.

3.5 Conditional Random Field with Decomposition Inference

The previous model describes the activities of both residents using a single model, which implies that two adjacent activities will be relevant even if they are performed



(a) The linear-chain CRF for initialization

(b) The CRF to infer the data association sequence $\theta_{1:t}$

(c) The CRF to infer the activity sequence $A_{1:t}$

Fig. 5. Three CRFs used in conditional random field with iterative training

Algorithm 1. The inference algorithm in the CRF with iterative inference

```
Input observations $O_{1:t}$
Initialize $A_{1:t}^{(0)}$ given $O_{1:t}$ by the CRF in Fig. 5(a)
While $A_{1:t}^{(i)} \neq A_{1:t}^{(i+1)}$ or $\theta_{1:t}^{(i)} \neq \theta_{1:t}^{(i+1)}$
    Infer $\theta_{1:t}^{(i+1)}$ by $O_{1:t}$ and $A_{1:t}^{(i)}$ by the CRF in Fig. 5(b).
    Infer $A_{1:t}^{(i+1)}$ by $O_{1:t}$ and $\theta_{1:t}^{(i+1)}$ by the CRF in Fig. 5(c).
```

by different residents, which is not a reasonable implication. Moreover, it seems unreasonable to implement a single chain to model multiple residents.

In decomposition inference, we assume that the influence of interaction between residents is negligible; as a result, given the data association, it is practicable to use single-resident activity sequences to infer the activities of each person, then we can decompose a multiple-resident problem into sub-problems using single-resident models. In this paper, we tried to use two approaches, transition table and conditional random field with iterative inference, to determine the data association variables, which are addressed in the following.

Transition Table

If we know that a sensor event s_{t} at time t is generated by a resident, we may use this information to infer who generate the next event s_{t+1} by the relationship between s_t and s_{t+1}. The easiest way to do this is to compute the transition probability between any two sensor events s_i and s_j. Define the transition probability of two sensor events $P(s_i \rightarrow s_j)$ to be a value proportional to the count of adjacently triggered events (s_i and s_j) of each person where s_i, s_j are two sensor events. At each time slice, we assign the one with greater transition probability to the association.

Conditional Random Field with Iterative Inference

As aforementioned, a CRF with iterative inference generates both association and activity. This generated association is used to divide the whole problem into two single-resident activity sequences. The procedure can be seen in Algorithm 2.

Algorithm 2. The inference algorithm in the CRF with decomposition training

```
Input observations $O_{1:t}$
Infer $\theta_{1:t}$ given $O_{1:t}$ by Transition Table or CRF with Iterative Inference
Decompose $O_{1:t}$ into two groups according to $\theta_{1:t}$ .
Apply Linear-Chain CRF to infer $A_{1:t}$ according to the decomposed observation.
```

4 Experimental Results and Discussion

In this section we present the experimental results from the research. Each experiment is evaluated using the leave-one-out cross-validation of 26 files provided by the CASAS project.

The first experiment compares the performance of several solutions for multiple-resident activity recognition with respect to the three types of preprocessing, which includes raw data, environment data and room-level data, to validate the accuracy and investigate the tendency of data preprocessing. The second experiment shows the correlation between association and activities. Based on the CRFs with decomposition training, we couple them with different types of data association mechanisms and analyze the accuracy.

4.1 Experiment 1: Tendency of Data Preprocessing

Table 2 shows the averaged accuracy for the three models we are interested in, of which the best one is greater than the accuracy in [5] (0.6060), but still not satisfactorily accurate. Moreover, the CRFs decomposition method outperforms others, which shows that if we can properly separate the dataset, we can apply single-resident model to the multiple-resident problem. However, the insufficient accuracy may be due to the nature of the human activity pattern under the multiple-people condition. The CRF with iterative training is intuitively competent in those files with more interactions than the one with decomposition training. On the other hand, inferring the activities of two residents separately performs well in the case of less relation between the residents. However, the dataset is a mixture of individual and cooperative activities, thus causing insufficient accuracy.

The accuracy for models coupled with distinct preprocessed data are also illustrated in Table 2. Surprisingly, the highest accuracy occurs while the model coupled with the raw data. This decline in accuracy for more manipulation is probably attributed to the sensitivity to noises. That is, taking all information into consideration may result in overfitting to the noises. Furthermore, the lower accuracy of room level data would seem to stem from little dependency between room-level features and their associated activities.

Table 2. Average and variance pairs for distinct models and data preprocessing

average /variance	CRF with iterative training	CRF with decomposition (transition)	CRF with decomposition (iterative)
raw data	0.5067 / 0.0139	0.5953 / 0.1070	0.6416 / 0.0128
environment data	0.3323 / 0.0950	0.3026 / 0.0083	0.3425 / 0.0118
room level data	0.2449 / 0.0065	0.2333 / 0.0131	0.2655 / 0.0090

4.2 Experiment 2: Correlation between Data Association and Accuracy

In this experiment, we compare the existing mechanisms and the CRF with decomposition given data association. As shown in Table 3, the accuracy of data association and the accuracy of activity are positively correlated. In addition, the low accuracy of the model coupled with transition table may be due to three factors: noise, smoothing and relevance of data association and activity. The transition accuracy may be highly sensitive to the noise. Moreover, if a transition does not appear in the training phase, data smoothing is required. Finally, according to the dataset, it suggests that some activities are performed by a specific resident.

Table 3. The accuracies for distinct data association mechanisms

	accuracy of data association	accuracy/variance of activity
given association	1.0000	0.8137 / 0.0248
transition table	0.5099	0.5067 / 0.0139
iterative training	0.7287	0.6416 / 0.0128

5 Conclusion

Multiple-resident activity recognition is an important yet challenging problem and we address this problem by applying CRFs with different strategies, which include different ways to train a CRF and to preprocess the dataset. We also investigate the importance of data association in multiple-resident activity recognition.

In the first experiment, the CRFs decomposition method outperforms other classifiers we proposed, which shows that if we can properly decompose the dataset for each individual, we can apply single-resident model to a multiple-resident problem to some degree. However, the insignificant accuracy of the CRF with decomposition training indicates that totally ignoring the interactions among residents does compromise the recognition accuracy. Moreover, in this experiment, preprocessing the dataset results in low accuracies, which is contrary to our expectation. This result suggests that in preprocessing a dataset, we have to take into consideration some factors such as the noise in the dataset and the prior knowledge of the environment. In the second experiment, we show that data association in a multiple-resident environment is a vital issue. If we have a good algorithm for data association problem, the activity recognition accuracy may be improved efficiently.

This study shows that data association is an inevitable issue in dealing with multiple-resident activity recognition. It also reveals that modeling human interactions is also necessary in making the system more reliable. We therefore encourage researchers to develop more appropriate solutions for these two critical issues so that the methods proven effective for single-resident activity recognition can be extended and applied to a multiple-resident environment.

References

1. Wang, L., Gu, T., Tao, X., Lu, J.: Sensor-Based Human Activity Recognition in a Multi-user Scenario. In: Ambient Intelligence, pp. 78–87 (2009)
2. McCowan, L., Gatica-Perez, D., Bengio, S., Lathoud, G., Barnard, M., Zhang, D.: Automatic analysis of multimodal group actions in meetings. IEEE Transactions on Pattern Analysis and Machine Intelligence 27, 305–317 (2005)
3. Nguyen, N., Venkatesh, S., Bui, H.: Recognising Behaviours of Multiple People with Hierarchical Probabilistic Model and Statistical Data Association. Presented at the British Machine Vision Conference (2005)
4. Wilson, D.H., Atkeson, C.G.: Simultaneous Tracking and Activity Recognition (STAR) Using Many Anonymous, Binary Sensors. Presented at the Pervasive (2005)
5. Singla, G., Cook, D., Schmitter-Edgecombe, M.: Recognizing independent and joint activities among multiple residents in smart environments. Ambient Intelligence and Humanized Computing Journal (2009)
6. Lu, C.-H., Wu, C.-L., Fu, L.-C.: Hide and Not Easy to Seek: A Hybrid Weaving Strategy for Context-aware Service Provision in a Smart Home. Presented at the IEEE Asia-Pacific Services Computing Conference, Yilan, Taiwan (2008)
7. Kasteren, T.v., Noulas, A., Englebienne, G., Krose, B.: Accurate activity recognition in a home setting. Presented at the Proceedings of the 10th international conference on Ubiquitous computing, Seoul, Korea (2008)
8. Ye, J., Clear, A.K., Coyle, L., Dobson, S.: On using temporal features to create more accurate human-activity classifiers. Presented at the 20th Conference on Artificial Intelligence and Cognitive Science, UCD Dublin, Ireland (2009)
9. Chen, Y.-H., Lu, C.-H., Hsu, K.-C., Fu, L.-C.: Preference Model Assisted Activity Recognition Learning in a Smart Home Environment. Presented at the International Conference on Intelligent Robots and Systems, St. Louis, MO, USA (2009)
10. W.S. University: CASAS Smart Home Project, `http://ailab.eecs.wsu.edu/casas/`
11. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Presented at the ICML (2001)
12. Vail, D.L., Veloso, M.M., Lafferty, J.D.: Conditional random fields for activity recognition. In: International Conference on Autonomous Agents and Multi-agent Systems (2007)
13. Lian, C.-C., Hsu, J.Y.-J.: Probabilistic Models for Concurrent Chatting Activity Recognition. Presented at the Proceedings of IJCAI-2009, Pasadena (2009)
14. Shewchuk, J.R.: An Introduction to the Conjugate Gradient Method Without the Agonizing Pain (1994)
15. Darroch, J.N., Ratcliff, D.: Generalized iterative scaling for log-linear models. Presented at the Annals of Mathematical Statistics (1972)
16. Nocedal, J.: Updating Quasi-Newton Matrices with Limited Storage. Mathematics of Computation 35, 773–782 (1980)
17. Kudo, T.: CRF++: Yet Another CRF toolkit (2007), `http://crfpp.sourceforge.net/`

Methodology to Achieve Accurate Non Cooperative Target Identification Using High Resolution Radar and a Synthetic Database

Antonio Jurado-Lucena, Borja Errasti-Alcalá, David Escot-Bocanegra,
Raúl Fernández-Recio, David Poyatos-Martínez, and Ignacio Montiel Sánchez

Detectability and Electronic Warfare Laboratory.
Instituto Nacional de Tecnica Aereoespacial – INTA
Ctra. de Ajalvir, Km. 4, Torrejón de Ardoz, 28850, Madrid – Spain
{juradola,errastiab,escotbd,fernandezrr,poyatosmd,
montielsi}@inta.es

Abstract. In the last few years, there is a great interest in developing an identification system capable to make a reliable classification of aircrafts into different groups (friendly, hostile or neutral). Depending on the context in which these systems are deployed, incorrect identification may lead to serious problems, such as fratricide or engagement of civilian aircrafts. Different techniques have been researched to face this problem, but non-cooperative ones have awakened more interest because they do not require aircraft collaboration. Non Cooperative Target Identification (NCTI) using radar is a complex task, mainly due to the fact that a database of possible targets is needed. To populate this database, Radar Cross Section (RCS) predictions produced by computer simulation seem to be the most feasible way to perform this task, since measurements alone cannot cover the vast range of targets, configurations and required aspect angles. These predictions are typically performed in the frequency domain and a specific processing must be done to obtain both High Resolution Range Profiles (HRRPs) and 2D Inverse Synthetic Aperture Radar (2D-ISAR) images. This paper shows a methodology to face the NCTI task, which use both synthetic HRRPs and 2D-ISAR to achieve an accurate identification.

Keywords: NCTI, NCTR, HRRP, ISAR, Radar, Synthetic Database.

1 Introduction

One of the main concerns in the Defence Area is to have a reliable identification system that minimizes fratricide between allied forces. Radar systems are the most suitable military sensors for facing the need for rapid and reliable identification of (air-) targets as it offers, compared with optical and IR sensors, long range day/night performance, penetration of weather, smoke, dust and vegetation [1].

The different techniques that have been researched to solve this problem may roughly be divided into two classes [2]: cooperative and non-cooperative techniques.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 427–436, 2010.

Cooperative techniques (often referred to as IFF - Identification Friend or Foe - techniques) are already operational in the radar domain, e.g. in airborne radars. Most of fighter aircrafts are equipped with transponder systems answering to authorized interrogations by transmitting a predetermined coded signal. By this, friendly aircrafts may be identified (if the IFF is working properly) but positive identification of hostile or neutral aircrafts is not possible. This could in principle be achieved by the so-called Non-Cooperative Target Identification (NCTI) techniques which rely on a comparison between the measured target signatures and a reference database.

NCTI can be mainly accomplished by Jet Engine Modulation (JEM), High Resolution Range Profiles (HRRPs) or 2D Inverse Synthetic Aperture Radar (2D-ISAR) images [3]. JEM is a relatively wide operational technique but it has the drawback of not all-aspect identification. On the other hand, HRRPs and 2D-ISAR images give an all-aspect capability.

For these two techniques, the database tends to be populated with predictions obtained by electromagnetic software tools. These predictions form the so called “synthetic database”. Other options, such as measurements, on scale or real size, imply high cost due to the need of expensive facilities and difficulties for obtaining all the possible targets, aspect angles and configurations.

Other important issue to take into account is the processing of raw radar data to obtain HRRPs and ISAR images. This processing is very complex and time consuming, mainly for ISAR images, so the classification scheme should be fitted to it. As it will be shown, this is one of the main advantages of the proposed classification method.

NCTI systems can be divided into two groups: on the one hand, NCTI systems capable of distinguishing groups of targets with similar features, such as for example, a civilian airplane, a military aircraft or a Unmanned Aerial Vehicle (UAV). This identification is called class identification (class ID). On the other hand, NCTI systems that are able to differentiate among targets that belong to the same class (military aircraft: F-18, F-22, MIG-31, etc). This other identification is called type identification (type ID).

This paper shows a methodology to face the NCTI task by means of a synthetic database. This methodology is divided into two stages. First of all, actual HRRPs are used as inputs on an Artificial Neural Network (ANN) already trained with synthetic profiles. This allows class ID. After that, and using this first classification, actual 2D-ISAR images are compared by means of the 1-Nearest-Neighbour (1NN) algorithm, with Euclidean distance or correlation as the distance metric, to achieve type ID.

2 Data Processing

Target information can be obtained using short-pulse radar systems, but the use of short pulses is limited by the system’s bandwidth capabilities or the transmitted power needs, so other alternatives such as pulse compression techniques (chirp modulation, binary phase coding, Barker coding…) or stepped frequency waveforms are preferably used. For pulse-compression radars operating at the same bandwidth and range of frequencies, the matched-filter output signal is approximately the same as that for a short-pulse radar.

Stepped-frequency radars can achieve high resolution with a simple hardware architecture and do not require a wide instantaneous bandwidth. They transmit a continuous series of short monotone pulses with the same amplitude, each of them at a different frequency. Each train of pulses is called *burst*. Therefore, a stepped-frequency signal consists of *M* bursts of *N* pulses whose Pulse Repetition Frequency (PRF) is *1/T*. Each frequency component in the burst is $f_n = f_0 + n\Delta f$, where f_0 is the initial frequency and Δf is the frequency step in the burst, yielding the total bandwidth of $(N - 1)\Delta f$ (Fig. 1).

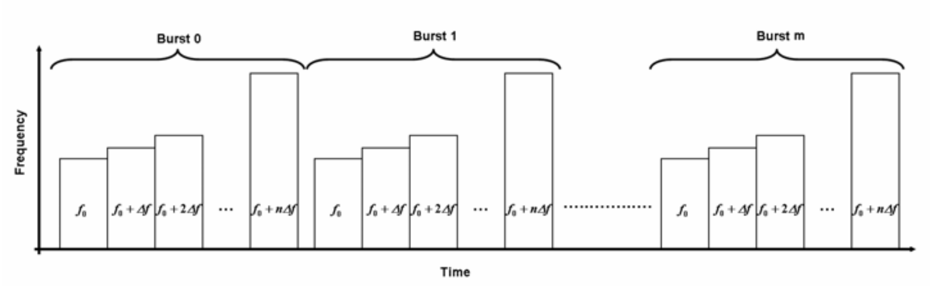


Fig. 1. Stepped Frequency schema

Radars usually measure a sequence of consecutive bursts over a period of time. The motion of the aircraft during this period causes the aircraft to appear at different positions in different bursts. This effect is known as Translation Range Migration (TRM). Aircraft motion also influences the pose of the aircraft with respect to the radar (or aspect angle). Occlusion of scatterers, Rotational Range Migration (RRM) and speckle are some other effects caused by aircraft rotations which greatly affect the data.

When comparing simulated to measured data for classification purposes, discrepancies between them must be assumed and understood [4]. Simulated data have less information than measured data because certain effects such as high order reflections or cavity resonances, present in actual measurements, are still hard to simulate with electromagnetic software tools. However, some scatters are still common in both simulated and measured data, and they can be used for classification [5].

Therefore, both simulated and measured data must be obtained to optimize the likeness between them, so they need a post-processing to adequate them all and achieve an accurate identification.

2.1 High Resolution Range Profiles (HRRPs)

Range profiles can be seen as a 1D image of an aircraft, where the parts of the aircraft that reflect the radar radiation, called scatterers, project their reflection onto the Line Of Sight (LOS). This information, different for each target, is suitable for automatic aircraft classification since it depends on the geometry of the target (see Fig. 2).

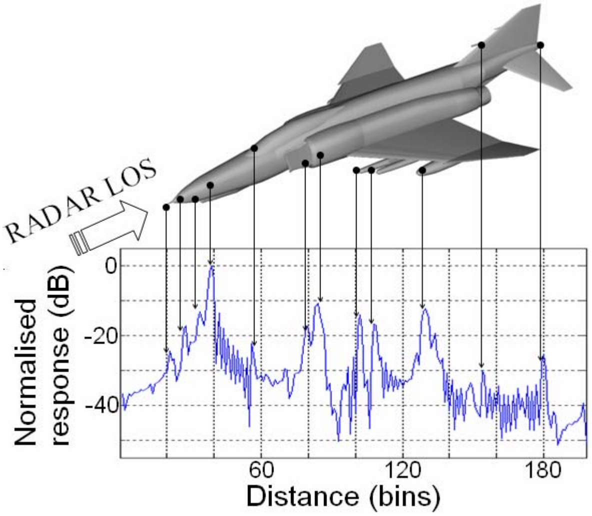


Fig. 2. HRRP example

Range profiles following the stepped-frequency scheme are produced by emitting a bandwidth *β* using *N* pulses with linearly increased frequencies. The resolution of a profile can be described in terms of its capability to resolve targets that are separated in range. The fundamental relationship for the inherent range resolution *ΔR* associated with radar bandwidth *β* is (c is the speed of light):

$$\Delta R = \frac{c}{2\beta} \; . \tag{1}$$

The coherent responses (*N* complex numbers) are Fourier transformed (by means of inverse Fast Fourier Transform - *iFFT*) and the phase information of the result is discarded, being only the absolute value considered [6].

$$x[\,n\,] = iFFT\big(X[\,k\,]\big) \Rightarrow x[\,n\,] = \frac{1}{N}\sum_{k=0}^{N-1} X[\,k\,]e^{f\frac{2\pi}{N}kn} \; . \tag{2}$$

Usually, windowing functions are applied before Fourier transforming to reduce spectral leakage, although they cause a reduction in resolution. To diminish this negative effect on the resolution, the combination of two windowing functions, one with a strong effect in the spectral leakage (e.g. Hamming windowing function) and other with less (e.g. rectangular windowing function) is an option.

Besides, both optional zero-padding on the Fourier transformation or interpolation can be applied and return HRRPs containing as many points as needed. It is worth mentioning that this processing does not increase the amount of information but adapts the data format to the classification methods. It must be taken into account that, if both a combination of two windowing functions and zero-padding are to be used, peak splitting effect could appear which is not beneficial for the identification process. Finally, HRRPs must be normalized, either in energy or maximum amplitude.

In this work, a combination of Hamming and rectangular windowing functions are applied to obtain HRRPs. After that, interpolation is used to achieve HRRPs with 512 points. Then, HRRPs are normalized so its energy is equal to 1. Finally, thresholding is considered to discriminate scatters from HRRPs which are not important for the classification, i.e., scatters produced by noise or even actual ones but with low level with respect to the energy of the whole profile.

2.2 Inverse Synthetic Aperture Radar Image (2D-ISAR)

A 2D-ISAR image is a two-dimensional representation of the distribution of the target reflectivity. ISAR images are produced by the rotation of the target and the processing of the resulting Doppler histories of the scattering centres [6]. If the target rotates in azimuth at a constant rate through a small angle, scatters will approach to the radar at a rate depending only on the cross range position (the direction perpendicular to the radar line of sight with the origin at the target axis of rotation). The rotation will result in the generation of cross range dependent Doppler frequencies which can be sorted by a Fourier transform. For small angles, an ISAR image is the two-dimensional Fourier transform of the received signal as a function of frequency and target aspect angle [7–8].

ISAR is used to obtain information on both the down-range and cross-range. The down-range resolution of an ISAR image is directly related to the bandwidth of the transmitted radar signal, and the cross-range resolution is determined by the effective rotation of the target.

The received stepped-frequency echo signal can be expressed as follows:

$$s(m,n)=\sum_{k=1}^{K} a_k e^{-j\frac{4\pi f_n}{c}\vec{r}_k(m,n)} . \tag{3}$$

Where:

$$\vec{r}_k(m,n)=\vec{r}_{k0}(m)+\vec{v}_r(m)t_n+\frac{1}{2}\vec{a}_r(m)t_n^2 . \tag{4}$$

– K is the number of scattering centers.
– a_k is the magnitude of the k^{th} scattering center.
– $\vec{r}_k$ *(m,n)* is the radial distance to k^{th} scattering center projected onto radar LOS.
– $\vec{r}_{k0}$*(m)* is the initial radial distance of k^{th} scattering center at the start of the m^{th} burst.
– $\vec{v}_r$ and $\vec{a}_r$ respectively represent the velocity and acceleration along radar LOS.

In (3), a change of $\vec{r}_k$ between pulses causes the image to be blurred. Therefore this blurring should be eliminated using proper estimation of $\vec{v}_r$ and $\vec{a}_r$ for each burst. For this reason, range profiles should be aligned using proper range alignment algorithms [9].

In this work, HRRPs obtained above are also used to generate ISAR images. These HRRPs are aligned by means of regression on the correlation between adjacent profiles and also, dominant peak alignment is used. Therefore, and although the estimation of $\vec{v}_r$ and $\vec{a}_r$ is not performed specifically, acceptable ISAR images are achieved (see Fig. 3). Every generated ISAR image requires an additional process that includes windowing, normalization and thresholding to obtain a new ISAR image with only the target signal. This new image is normalized to specific number of points of resolution per axis, where each axis has a length which is fixed previously.

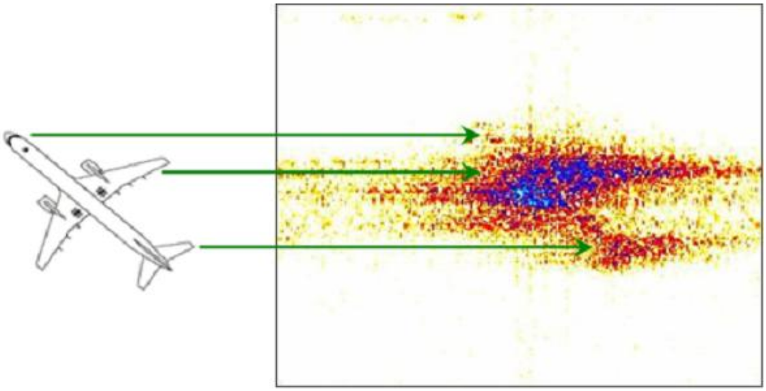


Fig. 3. Boeing 757 ISAR

3 Methodology

In this manuscript a methodology for NCTI is presented, where HRRPs are used as inputs to an Artificial Neural Network (ANN), providing class ID. After this first pre-classification and to achieve type ID, 2D-ISAR images are compared by means of the 1-Nearest-Neighbour algorithm, using Euclidean distance analysis or correlation as metric (Fig. 4).

This methodology can be applied if an estimation of the aspect angle is available, since the idea is to have an array of ANN, where each of them is trained with HRRPs of different targets, whose aspect angles are similar. Each ANN has as many outputs as classes of targets. In this case, three possible outputs have been considered: UAV, military aircraft and civilian airplane. This class ID allows to start the type ID process using ISAR images that belong to the class selected by the ANN.

As it has been shown, a set of HRRPs is needed to compute an ISAR image. Thus, the methodology fits properly with the processing scheme followed to obtain ISAR images from a High Resolution Radar. First of all, HRRPs are obtained and used to get the class ID, then those HRRPs are used to create the ISAR image and perform the Type ID taking into account the class of the target previously obtained.

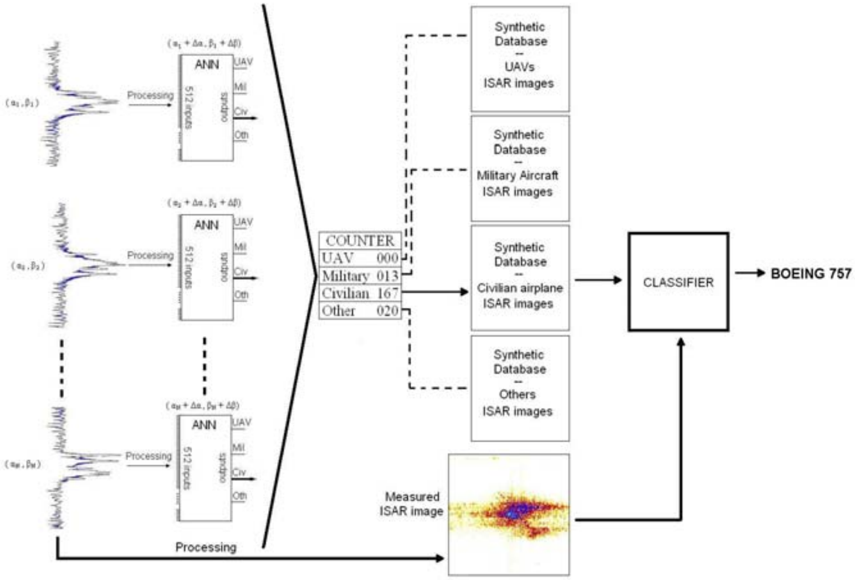


Fig. 4. Schema of the Methodology

3.1 Class ID through HRRPs and ANNs

As noted above, it is generally assumed that a range profile is a sum of the contribution of a set of discrete scatterers. The number, position and other properties of these scatterers are, in general, different for each type of aircraft. Range profiles are therefore a suitable measurement data for aircraft classification [10–13].

If enough samples of the aircraft class and the corresponding profile are available, different pattern recognition techniques can be used to construct a classifier. Note, however, that range profiles are strongly dependent on the pose of the aircraft at the time of measurement. Constructing a reliable classifier therefore requires a large training set of range profiles at different aircraft poses.

A specific processing has been done to achieve suitable HRRPs to be used as inputs in an ANN, where common scatters in synthetic and measured HRRPs are used as useful features for classification.

In this communication, the large training set consists of simulated HRRPs, which once the post-processing is over, are ready to be introduced in the ANN for training, validation and test purposes. The ANN applied is a Perceptron Multilayer with feedback propagation, which has one input layer (512 inputs, one input for each resolution cell), one hidden layer and one output layer (one output for each target class). At present, the number of neurons in the hidden layer, different networks parameters, and also other types of ANN, are being tested to obtain the optimum architecture for an accurate class ID.

Figure 5 shows a training set, which consists of 6300 HRRPs (each column is one profile) belonging to 5 different types of aircrafts. The simulated aspect angles are from -4.6409º in azimuth and 1.6164º in elevation to 9.6046º in azimuth and -1.6295º in elevation.

In this picture, it can be seen that the profiles that belong to the same type have scatters in similar positions. Also, the last scatter in each profile provides the target length information. Therefore it is expected that the ANN can distinguish among them.

Once the ANN is trained, it is ready to accept inputs from actual data. The ANN almost instantly returns an output with the selected class. This process is repeated until a certainly threshold is exceed, and then, the measured ISAR image is generated and begins the second ID stage, where this image is compared with the synthetic database corresponding with the selected class.

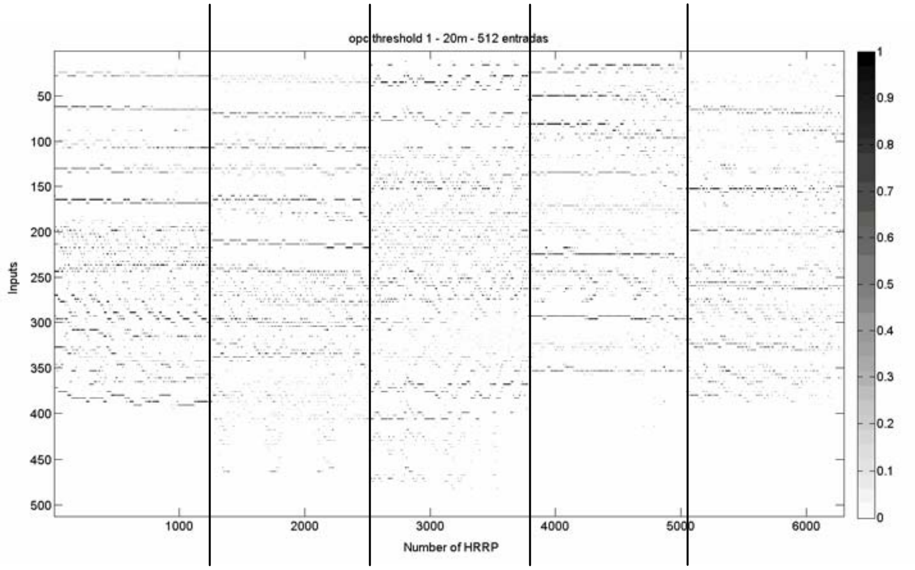


Fig. 5. Training set made up of five different types

3.2 Type ID Classification through ISAR Images and the 1-NN Algorithm

As it is said above, an ISAR image is formed by coherently processing a returned signal from a moving target at a different aspect angle relative to the radar. The change of the aspect angle is usually due to the relative motion between the radar and the target. When the target is moving smoothly, conventional motion compensation algorithms can be applied to generate a clear image of the target; however, a smoothly motion is not usual, so further investigations are being carried out to develop robust automatic motion compensation algorithms.

Once the class ID has finished and the ANN gets an accurate output about the target class, the NCTI system begins to compare the measured ISAR image with the simulated ISAR images database. Two simple methods have been used at this stage, and it has been shown that they are robust and reliable [14–15].

The first method is based on the minimum Euclidean distance estimation, where the measured ISAR is compared, by subtraction, pixel by pixel with each synthetic image on the database. The second one is based on comparison between the actual

ISAR image and the synthetic ones by means of the correlation function. Both methods suffer from a high computational cost although they have shown a good behaviour.

4 Results and Conclusions

A methodology for NCTI has been presented, which consists of two stages. First, HRRPs are used as inputs in an ANN, which provides a class ID of the target. Then, those HRRPs are used to generate an ISAR image, which is compared to synthetic ISAR images by means of the 1-NN algorithm. This last stage provides a type ID of the target.

Note that if the ANN is trained properly, it is able to distinguish between classes of targets, so this conclusion can be used as the first classification stage in this methodology. At this moment, different ANN architectures are being tested to determinate which one is more adequate for the class ID task.

Secondly the 1-NN method is used to compare ISAR images. The distance metrics tested are computationally expensive, however, good results on type ID have been obtained.

Our proposal is to continue with this methodology, going further into different ANNs and their architectures, until achieving the best setup, and improving the 1-NN algorithm to reduce the computation time.

References

1. Cohen, M.N.: An Overview of Radar-Based Automatic Non Cooperative Target Recognition Techniques. In: IEEE International Conference on Systems Engineering, pp. 29–34 (1991)
2. Schiler, J., Cranos, R.L.: Non-Cooperative Air Target Identification Using Radar. In: RTO MP-06 (1998)
3. Rosenbach, R., Schiller, J.: Non Co-Operative Air Target Identification Using Radar Imagery: Identification rate as a function of Signal Bandwidth. In: IEEE International Radar Conference, pp. 305–309 (2000)
4. Escot-Bocanegra, D., Poyatos-Martínez, D., Fernández-Recio, R., Jurado-Lucena, A., Montiel-Sánchez, I.: New benchmark radar targets for scattering analysis and electromagnetic software validation. Progress In Electromagnetics Research, PIER 88, 39–52 (2008)
5. Zwart, J.P.: Aircraft Recognition from Features Extracted from Measured and Simulated Radar Range Profiles. PhD Thesis, University of Amsterdam, The Netherlands (2003)
6. Wehner, R.D.: High Resolution Radar. Ed. Artech House, Norwood (1995)
7. Chen, V.C., Lipps, R.: ISAR Imaging of Small Craft with Roll, Pitch and Yaw Analysis. In: IEEE International Radar Conference, pp. 493–498 (2000)
8. Chen, V.C., Micelli, W.J.: Simulation of ISAR Imaging of Moving Target. IEEE Proceedings Radar, Sonar and Navigation, 160–166 (2001)
9. Jeong, H., Kim, H., Kim, K.: Application of Subarray Averaging and Entropy Minimization Algorithm to Stepped Frequency ISAR Autofocus. IEEE Transactions on Antennas and Propagation, 1144–1154 (2008)

10. Nelson, D.E., Starzyk, J.A.: Advanced Feature Selection Methodology for Automatic Target Recognition. In: Proceedings of the Twenty-Ninth South-eastern Symposium on System Theory, pp. 24–28 (1997)
11. Mitchell, R.A., Westerkamp, J.J.: Robust Statistical Feature Based Aircraft Identification. IEEE Transactions on Aerospace and Electronic System, 1077–1094 (1999)
12. Jiang, N., Wu, R., Li, J.: Super Resolution Feature Extraction of Moving Targets. IEEE Transactions on Aerospace and Electronic Systems, 781–793 (2001)
13. Li, H., Yang, S.: Using Range Profiles as Feature Vectors to Identify Aerospace Objects. IEEE Transaction on Antennas and Propagation, 261–268 (1993)
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Ed. Wiley, Chichester (2000)
15. Heiden, R.: Aircraft Recognition with Radar Range Profiles. PhD. Thesis, University of Amsterdam, The Netherlands (1998)

Modeling Spatial-Temporal Context Information in Virtual Worlds

Ángel Arroyo, Francisco Serradilla, and Óscar Calvo

Dpto. Sistemas Inteligentes Aplicados, Universidad Politecnica de Madrid, Spain
aarroyo@eui.upm.es, fserra@eui.upm.es, oscar.calvo@upm.es

Abstract. Currently, we use many definitions with diffuse boundaries: Web 2.0 (Social Networks), Web 3.0 (Semantic Web), Web 3D (Metaverses, Virtual Worlds, Mirror Worlds), Recommendation Systems, Augmented Reality, Geo-location... In this paper we explore the possibilities of the combined use of these concepts, we introduce the concept of VARD (Virtual Augmented Reality Device) and show interoperability between recommendation systems and Virtual Worlds. We have developed a Recommendation System which have two ways of interaction with the virtual world of Second Life in connection with context spatial-temporal information: an active recommendation system, called TESLAR, that interacts with avatars by a 2D HUD VARD object, and a passive and automatic recommendation system, called MarvinBot, that interacts with avatars by a Metabot VARD.

Keywords: Virtual Worlds, Augmented Reality, Metabots, Intelligent Agents, Recommendation Systems, Context Information, Multiagent Systems, Second Life, Mirror Worlds, Metaverses.

1 Introduction

The semantics of human behavior is interpreted in a persistent 3D space which is the real space-time. In this paper we postulate that the semantic representation of the behavior of avatars, their statements and actions in a persistent 3D virtual world, as is Second Life (SL), extends the margins of man-machine communication and brings us closer to a kind of sixth sense as defined by Pattie Maes [Maes & Mistry, 2009]. Augmented Reality opens the door to this sixth sense setting in each moment the relevant information to the current context and adding the necessary tools for handling such relevant information. We define the term VARD to refer to Virtual Augmented Reality Devices in metaverses.

In this paper we present the results of our experiments about combining recommendation systems, virtual life, persistent virtual worlds and the semantics of these spaces to enhance the user experience through VARDs in SL.

We have developed a semantic model for the representation and manipulation of spatial-temporal information of the avatars in SL, a recommendation system for "places of interest" in SL, called Teslar, using a 2D VARD and a second recommendation system using a Metabot VARD (MarvinBot).

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 437–447, 2010.

The paper is organized as follows. In Section 2, we overview some related work in the field of semantic representation of behaviors in 3D persistent virtual worlds, providing an introduction to knowledge representation and reasoning with ontologies and describing the spatial and temporal knowledge used in our work. In section 3, we show related works in the field of recommendation systems and the methods and the tools employed in this work. In section 4, we describe the concept of VARD. Next, in sections 5 and 6, we detail the recommendation systems developed (Teslar and MarvinBot). Finally, the paper concludes with a brief discussion on the results and plans for future research work.

2 Context Spatial-Temporal Information

Virtual spaces share spatial-temporal semantic with the real world despite they have important differences. This allows us porting the tools and techniques employed in the real world for knowledge representation and reasoning with spatial and temporal information to virtual spaces.

With the impulse of the Semantic Web there has been a significant progress in developing the tools necessary to knowledge representation and reasoning with ontologies. These developments are widely used for context representation in many semantic domains [Assal et al, 2009]. UML (Unified Modeling Language) provides a standard notation for modeling systems and context using object-oriented concepts. OWL (Ontology Web Language) provides a standard for defining ontologies and Protégé (open and extensible software based on Java) allows us to edit these ontologies. RDF (Resource Description Framework) is used to representing the context in many applications; for example, Clear, Dobson, and Nixon construct RDF triples from sensor's data using an ontological description of context and they have defined a query service interface for reasoning with this ontology on complex systems in ubiquitous computing [Clear et al, 2007].

2.1 In Real Life

Systems that perform tasks identification and annotation of the activities carried out by humans in the real world (lifelog) require sensors to capture these activities. Digital cameras get information from a given real environment, GPS systems provide location in real physical space, compasses indicate the orientation and the accelerometers convert certain gestures into an electrical signal which may be interpreted by the system. Mobile devices (mobile phones, PDAs...) have more and more types of sensors (GPS, compass, camera, accelerometer...) and these Context Sensitive Devices (CSDs) have certain awareness of their activity (position, orientation...) dependent on the equipment that they have. Cooperative mobile applications can be implemented on this historical information.

With digital cameras, for example, tracking objects and persons recognized in visual scenes, required a spatial and temporal model. Sánchez el al, show how to use OWL DL to the creation of an epistemological, functional, and structural ontology-based framework for Computer Vision that combines contextual and sensorial data to support scene recognition and enhance object tracking [Sánchez et al, 2009] [Gómez-Romero et al, 2009a] [Gómez-Romero et al, 2009b]. Snidaro, Belluz, and Foresti [Snidaro et al, 2007] show how to develop an OWL ontology enhanced with rules to represent and reason with objects and actors in the context of a surveillance application.

On the other hand, experience storage and retrieval leads to experience sharing [Kim et al, 2008] and the semantic representation of these experiences is the core in many of the future developments in Mirror Worlds, Augmented Reality and Semantic Metaverses. Kim and colleagues employ XML for time-based annotations of the Activity of Daily Living (ADL).

2.2 In Second Life

In the virtual world, the avatar that represents the user is acting as an active entity that has information about itself (profile), including membership of groups and an inventory of objects that can be used in the virtual space. As actor in a 3D space, the activity will be similar to that made by humans on the physical world and, therefore, need a semantics similar to that used by humans to describe their activities.

As an example of the use of Semantic Web standards in metaverses, Freese [Freese, 2007] describes a methodology for the conversion of RDF triples to AIML (Artificial Intelligence Markup Language) topics and categories which can then be used within an AIML-based bot. The combination of these two technologies allows to access to the knowledge represented within the RDF using natural language.

Similarly, in previous works [Arroyo et al, 2009], we have worked in the definition of metabot concept in the context of conversational agents. In this paper, we approach the problem of the spatial and temporal knowledge required for interact with the recommendation systems in virtual worlds and, in future jobs, we will experience in a greater semantic metaverse's content.

Virtual worlds do not require physical sensor devices to store any information on the activity of an object or avatar. This is an important advantage that allows us to develop sensors to store logical avatar behavior over time. In our case we store the information of position in space-time virtual view by the interpretation used in Second Life in which the world is divided into: Grid, State (Mainland or Private State), Region and Parcel. We consider the changes in location of the avatars in time, ie the connection or disconnection, and the parcel on which is connected and we will store the total time spent for avatars in visited places or parcels.

3 Recommendation Systems

Recommendation systems (RS) are software systems capable of providing its users with reviews and suggestions on objects within a domain of knowledge, in which might be interested. They can therefore be applied to any cultural object, movies, books, music and more. But also to other items, such as restaurants, vacation spots and in this case, to places within Second Life (SL).

The two fundamental approaches to address the problem of RS are: content-based methods, based on describing the object with its properties and carry out a matching process between objects, and collaborative methods, based on the search of other users of the system with similar preferences to the user asking for recommendations [Adomavicius & Tuzhilin, 2005].

In the case of SL is not easy or practical to describe each parcel through properties, but it is easy to give to the user the option to mark their favorite places. In fact he already does in some way when he/she add a site to their bookmarks. From this information it's possible to find other similar users using their favorite places, and it will be possible to recommend good places for their "soul mates".

One of the most widely used algorithm in collaborative RSs is the k-Nearest Neighbors (kNN), a common algorithm in the field of case-based reasoning [Shakhnarovish et al, 2005].

Metrics are a key element in this process. Basically there are mathematical functions that return a real number, ideally in the interval [0,1], calculated from vectors with ratings of two users u1 and u2. The most wide used algorithms are the correlation of the values of vectors and the cosine of the angle between vectors [Adomavicius & Tuzhilin, 2005].

A major problem of RS is sparsity [Bobadilla & Serradilla, 2009a], because the values for most pairs (user, item) are not defined, therefore the computing of metric only use values that are simultaneously defined for the two users, ignoring the rest.

4 VARDs

This new concept arises as a result from the study of new Metaverses and virtual environments. A Virtual Augmented Reality Device is an object which exists only into a given Metaverse but can connect this to any other reality, even the real one, increasing the perception capacities of its users.

This new concept is the result of merging two well known technology assets: Virtual Reality and Reality Augmented Devices.

But this is not a new idea: SCI-FI is always one step ahead. And we can find some examples of what a VARD can be at the present, even years before. For example, in the film TRON (1982), the character called "BIT" could be a first approach to the VARD concept. In this film, BIT is somewhat of a poliedron that answers yes or no to any question. Also in the well known MATRIX (1999) there are an easy example in the cell phone that Morpheus and company use to talk to Operator. Another example, in this case despite of its owner, is the locator-worm that is extracted from Neo's belly and used by "the computers" to mark and control humans within Matrix.

4.1 Augmentating Reality When Reality Is Virtual

A common augmented reality device allows humans to improve experimentation of "real" environments with the known limitations that exist when moving computer generated data to the human sensorium (large VR helmets, heavy data gloves, etc.). But even with these limitations this technology brings a new dimension to human perception of our world.

But, what happens when the reality is a virtual one? It is much easier to provide tons of information to a human into a Metaverse since the user is full-connected, full-located and full-virtualized by itself, or at least its avatar. Even more, when immersed into a virtual environment or Metaverse every single little simple object can contain all the computing power of a building filled with mainframes. That means there's no direct relationship between "size" and "power". So our avatar can wear a universal-translator-earring or even an invisible chess-assistant-pin.

The limits of this new brand of software products are the lack of realistic and massive Metaverses by today, but we all know those bridges will be crossed on next years, so the potential of VARD is enormous.

There's an important difference between Virtual-ARD's and Real-ARD's: commonly real-life Augmented Reality devices works only to increase the user's sense of vision. This is because we're new to this technology and vision is the most human bandwidth consumer. But in the original idea of Augmented Reality there's place for all forms of knowing, from simple, direct video enhancing to a complex near-subconscious sense of proximity to georeferenced special places transmitted with low frequency sounds.

4.2 Different Kinds of VARDs

Perhaps all VARDs must allow the user to "know" or "perceive" data from that virtual reality (Metaverse) or any other virtual reality (another Metaverse or real-world). There are, at least, four different kinds of VARD based on its morphology:

- **HUDs:** 2D objects that don't really belong to the Metaverse but to the Avatar controlling console. These HUDs often appear on a corner of the screen and uses 2D controls like buttons, text displays and gauges. Since are very similar to common 2D interfaces, this kind of VARDs are very easy to use for newby users and can be very helpful on the first contact with Metaverses. That's a common error to believe that those are the only possible ARD's, virtual or not.
- **Living Accessories:** Those that can be seen as real 3D objects into the virtual environment: clothes, accessories or any other object that all others users into that Metaverse can see and identify as an object that belongs to that reality. This "Living Accessories" will be attached in some way to the avatar's body and can't get away of it, that means its existence depends on the avatar's body existence.

- **Servants:** This kind of VARD is also a real 3D object and all users can see and interact with it. The difference with "Living accessories" is that "Servants" can move by itself and are independent from the avatar's body. This feature allows "Servants" to work as a presence extension. For example can be used as a "remote listener" from within Metaverse or, since its existence don't rely on your Avatar, can be "living" forever on metaverse and page or email you when something interesting happens and you're not logged on.
- **Metabots:** Metabot is a recently defined concept [Arroyo et al, 2009] but can be summarized as a "robot in the metaverse". The difference between "servants" and "metabots" is that Servants are a piece of the metaverse, since metabots express into a metaverse as a human would do, through an avatar. There's not an easy way to differentiate between a human-avatar and a metabot-avatar. Metabots are full versatile and can act as VARD as well as many other things, but in this paper we will see an example of how a metabot can perform lifelogging and augmented reality tasks.

4.3 VARDs Today, Second Life

In Second Life there's already a massive use of HUDs though rarely uses the concept of VARD. Usually HUDs work only as an extension of the control panel to perform actions on the avatar, wasting the potential of programming in LSL. Exceptionally, the use of automatic translators built by HUDs is being wide spread. This functionality fits comfortably with the concept of augmented reality and is a perfect example of augmented reality not oriented to the vision.

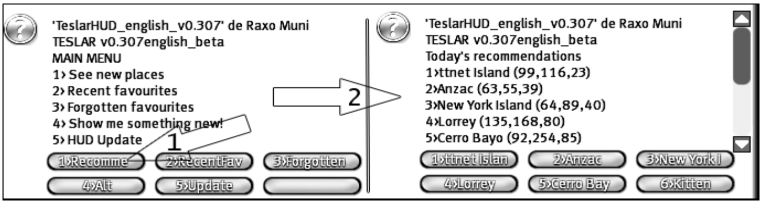
5 TESLAR. a RS with a 2D VARD

TESLAR is the name for an automatic recommendation system developed by AICU research group for running on Second Life, acronym for "Tier Engine for Second Life Augmented Reality". This system is able to recommend "places" in Second Life as long as all users shares automatically their place ratings. The system will learn of your likes and dislikes and comparing them with another users to recommend you the best valued places to visit into Second Life.

In this case, the VARD is of the "2D HUD" type, attached to the console main screen and showing the user data of the region in which he is in that moment. It also shows to the user a friendly buttons interface to rate regions and get new recommendations.

All calculations and operations are performed automatically, without human intervention and with maximum respect for user privacy.

When into Second Life and wearing TESLAR VARD any user can easily get a list of recommended places that the system has precalculated. Just press the button "See new places" and a list with the 12 best places will appear.



'TeslarHUD_english_v0.307' de Raxo Muni
TESLAR v0.307english_beta
MAIN MENU
1> See new places
2> Recent favourites
3> Forgotten favourites
4> Show me something new!
5> HUD Update

'TeslarHUD_english_v0.307' de Raxo Muni
TESLAR v0.307english_beta
Today's recommendations
1>ttnet Island (99,116,23)
2>Anzac (63,55,39)
3>New York Island (64,89,40)
4>Lorrey (135,168,80)
5>Cerro Bayo (92,254,85)

Fig. 1. TESLAR HUD main menu and recommendations

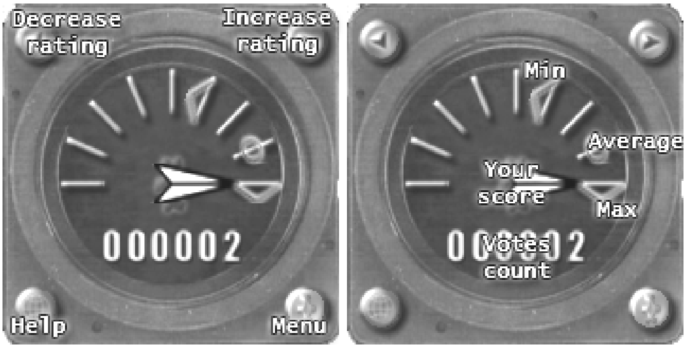


Fig. 2. TESLAR controls description

Additionally TESLAR system is able to suggest other places to visit:

- The places best rated by the user, its favorites.
- Good rated places long time ago.
- Places that you never visited but are very good to other people.

The mechanism for using TESLAR is very simple: just get the TESLAR VARD (of course totally free) from any TESLAR-TOTEM you find. There is one on the island of TESIS, headquarter of the AICU research group.

By touching the TESLAR-TOTEM you will buy a bunch of objects for 0L$. One of those objects is the TESLAR VARD. Just have to wear this new object and you will be participating the TESLAR project.

This object is designed to be attached to HUD in the upper left side. Once attached, this VARD type HUD will be showing the maximum, minimum and average score for the region and parcel the avatar is.

The HUD indicates your own score by a white marlin. The minimum and maximum value are shown with red and green angles respectively, and the average rating is displayed with the blue circle. The number of times this place has been rated appear on the ”odometer” of the HUD.

- To give your opinion simply use one of the top two buttons, red or green. According to changing your opinion go watch the needle as it turns yellow.
- To view the main menu click on the blue button (or anywhere in the panel than a button). When menu is displayed the version information is shown.

This is an active recommendation system since the user must actively press buttons in order to interact with the system. As we will see on the next chapter, that's a low processing VARD behavior and there are other much more proactive ways to do the same work.

But the TESLAR system is much more than a simple 2D HUD VARD object: this object by itself would not be able to match multiple users scores. This is only a front-end user interface that interacts with a much more complex external server program based on Java and MySQL database engine.

6 MarvinBot, a RS with a Metabot VARD

In previous studies [Arroyo et al, 2009] we have explored the possibilities of using Metabots. In this paper we will use these results to implement a recommender system that interacts with human-driven avatars through another avatar (in our case, Marvin) using the semantic model previously described.

Any RS has two main tasks: to feed the model (our variable parcels) with the location information of the avatars friends of Marvin and to provide useful recommendations to users according to their profile.

6.1 Metabots

The recommendation system based on MarvinBot, in contrast to TESLAR, is a passive system. This means that it is NOT necessary that the user actively scores each region, since Marvin is able to take note through Parcels vector of all the activity of all users involved in the system. All this information is stored in a database that is under constant scan for recognizable patterns that can be grouped and associated.

6.2 Privacy

As always happens with recommendation systems, the user feels suspicious of the idea that someone is controlling their every move, both in real life and in virtual environments. This distrust is logical and human, but is a major hurdle in developing useful RS. With the traditional RS, in which the user initiates the process of recommendation, this sense of lack of privacy is mitigated by the voluntary nature of the process. The system looks safer to users since they are who control what is known and what remains secret of themselves.

With MarvinBot RS score is the result of a continuous monitoring to the subject, absolutely controlling every move in Second Life. To avoid this lack of privacy we have developed two mechanisms. The first one: to enroll the project, the user should actively let MarvinBot to know its location. Second one, this permission may be revoked temporarily or permanently at any time. With these two tools we ensure the privacy of users. In any case we must remember that privacy is very relative, because in virtual environments there is no such thing: from the moment we enter the system each and every one of our activities is to be performed by a simulation system. If all these actions are recorded or not is only known by the metaverse provider. By definition the metaverses are authenticated thus, there is no anonymity possible.

6.3 Acquisition of Implicit Recommendations

As previously mentioned, in a passive RS the user is not actively performing any evaluation. We must investigate the evidence we have to guess evaluations. Based on passive collaborative filtering [Adomavicius & Tuzhilin, 2005] we have designed a set of values and activities that we used as indicators for the performance of ratings. The variables which we use are: region + parcel and time.

Following this empirically developed table we are able to generate automatic region scores:

Table 1. Score time table

Time in minutes	<1	1.2	2.3	3.5	5-15	15-30	30-60	60-120	120>
Score	1	2	3	4	5	6	7	8	9

Once the score is obtained it is sent to TESLAR RS, which calculates recommendations using the algorithm of k-neighbors with cosine metric.

As an improvement we have designed a control algorithm that stops the time counter after two minutes if no activity is detected on the user. With this modification we avoid false high ratings due to "idle console" issue.

6.4 Submission of Recommendations

In this first stage of the project we're using TESLAR RS to receive the recommendations. So, what we really have in this first phase is an active-passive hybrid system by which the implicit passive recommendations feeds the TESLAR database. That means MarvinBot collects information from the user's location and reports the time in each region and sends it to TESLAR, whereby the time table above described, time is transformed in score.

This architecture is described by the following graph:

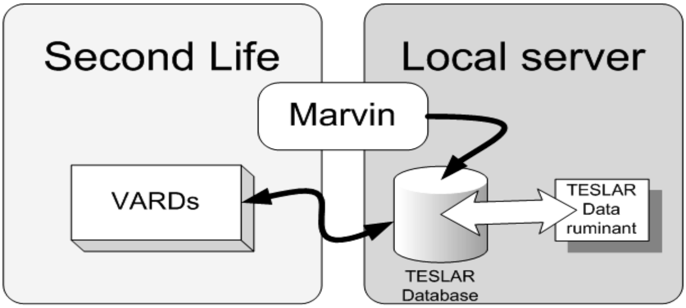


Fig. 3. TESLAR and MarvinBot common architecture

Therefore, as discussed above, we have an active-passive hybrid system by which recommendations are generated both implicitly and actively, and also the recommendations are received by request.

In future stages of this project it is planned to avoid HUD style VARDs and use direct semantic communication with Marvin taking advantage of his humanoid shape. Even more, the recommendations will be received by the users without having to wait until the user asks for it. Marvin will know when you need a recommendation and will pass it to you even before you know you need one.

7 Conclusions and Future Works

The work performed and presented in this article demonstrate the validity of our initial postulates at least considering the semantic constraints used. The reduced spatial-temporal semantics used in our experiments allowed us to develop a recommendation system that enhances the virtual experience of users in SL.

Future research will expand our view of these postulates adding greater semantic content to our models and virtual spaces. We will extend the semantic content increasing the spatial and temporal information stored to modeling behaviors in SL.

On the other hand, we intend to extend our experiments on conversational agents incorporating this spatial-temporal semantic in their interactions with the human-driven avatars and interconnecting both worlds, our facilities in the real world and a virtual isomorphic virtual space in SL, with the aim of broadening the spectrum of experience in space-time semantics shared with machines by means of knowledge representation and reasoning with spatial and temporal ontologies.

References

[Adomavicius & Tuzhilin, 2005] Adomavicius, G., Tuzhilin, A.: Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6) (June 2005)

[Arroyo et al, 2009] Arroyo, A., Serradilla, F., Calvo, O.: Multimodal agents in Second Life and the new agents of virtual 3D environments. LNCS, vol. 5601, pp. 506–516. Springer, Heidelberg (2009)

[Assal et al, 2009] Assal, H., Pohl, K., Pohl, J.: The Representation of Context in Computer Software. In: Focus Symposium, InterSymp-2009, Baden-Baden, Germany, August 3-7 (2009)

[Bobadilla & Serradilla, 2009a] Bobadilla, J., Serradilla, F.: The Incidence of Sparsity on Collaborative Filtering Metrics. In: Australian DataBase Conference (ADC 2009), Wellington, NZ, vol. 92, pp. 9–18 (2009)

[Bobadilla & Serradilla, 2009b] Bobadilla, J., Serradilla, F.: Reducing Recommender Systems Data Base Sizes and Improving their accuracy. In: International Conference on Internet Computing (ICOMP 2009), Las Vegas, pp. 196–202 (2009) (ISBN:1-60132-11-4)

[Clear et al, 2007] Clear, A.K., Dobson, S., Nixon, P.: An approach to dealing with uncertainty in context-aware pervasive systems. In: Proceedings of the UK/IE IEEE SMC Cybernetic Systems Conference 2007. IEEE Press, Los Alamitos (2007)

[Freese, 2007] Freese, E.: Enhancing AIML Bots using Semantic Web Technologies. In: Proceedings of Extreme Markup Languages, Montreal-Quebec (Agosto 7-10, 2007)

[Gómez-Romero et al, 2009a] Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Context-based reasoning using ontologies to adapt visual tracking in surveillance. In: 6th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS 2009), Genoa, Italy, September 2009, pp. 226–231 (2009)

[Gómez-Romero et al, 2009b] Gómez-Romero, J., Patricio, M.A., García, J., Molina, J.M.: Ontological representation of context knowledge for visual data fusion. In: 12th International Conference on Information Fusion (Fusion 2009), Seattle, WA, USA, July 2009, pp. 2136–2143 (2009)

[Kim et al, 2008] Kim, I.J., Ahn, S.C., Kim, H.G.: Experience Sharing in Tangible Web based on Lifelog. In: Asiagraph 2008 Proceeding (2008)

[Maes & Mistry, 2009] Maes, P., Mistry, P.: Unveiling the "Sixth Sense", game-changing wearable tech. In: TED 2009, Long Beach, CA, USA (2009)

[Sánchez el al, 2009] Sánchez, A.M., Patricio, M.A., García, J., Molina, J.M.: A context model and reasoning system to improve object tracking in complex scenarios. Expert Systems with Applications 36(8), 10995–11005 (2009)

[Shakhnarovish et al, 2005] Shakhnarovish, Darrell, Indyk (eds.): Nearest-Neighbor Methods in Learning and Vision. The MIT Press, Cambridge (2005)

[Snidaro et al, 2007] Snidaro, L., Belluz, M., Foresti, G.L.: Domain knowledge for surveillance applications. In: 10th International Conference on Information Fusion, Quebec, Canada, pp. 1–6 (2007)

Recognition and Interpretation on Talking Agents

José M. Fernández de Alba and Juan Pavón

Universidad Complutense de Madrid, Facultad de Informática,
28040 Madrid, Spain
jmfernandezdalba@gmail.com,
jpavon@fdi.ucm.es

Abstract. Speech is not a common way to interact with an artwork, but recent advances in Artificial Intelligence imply challenges and opportunities to explore new kinds of experiences in the confrontation of the spectator to the artwork. In order to have flexibility in experimentation by performing different art-installations, we have developed and implemented the concept of Talking Agent, a reusable software component with the ability to recognize human speech and synthesize a spoken response, and to make use of other sensors and actuators. The paper describes how the speech recognition and interpretation process of the Talking Agents have been studied in order to develop alternatives to obtain acceptable results in the desired scenarios.

1 Introduction

Most digital art consists on producing visual effects or the movement of mechanical devices as a result of spectator interaction, following simple algorithms or patterns. There is not yet too much application of advances in Artificial Intelligent (AI) techniques, like those in area of ambient intelligence and natural language processing, which could support interactions at a higher cognitive level, to conceive new ways of confrontation of the spectator to the art work. Notably, we are considering the use of speech processing techniques as the basis for the interaction between the art work and the spectator, which happens in a social context that can be reinforced with social abilities of multi-agent systems (MAS) and making use of other environmental information. In principle, this opens a broad diversity of exploratory paths in conceptual art, which involve implementing different art installations. To get this flexibility in the use and configuration of the various elements that take part of each art installation, the Talking Agents [5] are used as main reusable block. The Talking Agents are autonomous and social entities with the ability to interact with humans through speech and other sensors and actuators, and with other agents in a distributed computing environment. An art installation will consist of several Talking Agents, which are organized as a society of agents that may cooperate and interfere among them. The Talking Agent allows exploring conceptual issues, from the art perspective, such as the Turing test or emergent social behavior in the digital society, where

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 448–457, 2010.

humans and artificial entities live together: Among the interactions in a new hybrid and pervasive society, how to recognize who is behind, a human or a machine?

For the concrete considered scenario, in which spectators make arbitrary consultations to the agents, and maintain a brief conversation, the main objective is to determine the topics of an that consultation or later discourse, in order to prepare a related response. Since the conversation may be too abstract and cryptic, it is only needed a minimum feedback to the spectator that makes him realize that he is being understood somehow.

As far as possible, the development of Talking Agents relies on existing software components, such as for speech recognition, which has been studied in order to obtain a better performance for the previously mentioned objective. Section 2 presents an architecture overview of a single Talking Agents, focusing at the perception and interpretation phases of the information flow. Section 3 presents other natural language understanding systems and its strategies and explains why they were not been used in this work. Section 4 shows a study of the recognition and interpretation performance in the Talking Agent, explaining the process and considering the alternatives, and showing several test results. Finally, section 5 draws some conclusions and future work.

2 Architecture Overview

A single Talking Agent works like a conventional dialog system, with the only difference that it can achieve information from other agents. The information flow is like the one defined in [4], but introducing some context management, mainly to maintain data that may be interesting along the progress of conversation, like the current and previous topics of the conversation and other linguistic references; and coordination with other agents. Figure 1 represents the abstract information processing made by the agent.

The actual control model made by the agent is defined by a state machine following ICARO-T [11] XML schema for this purpose. The state machine describes a cycle, which starts in a waiting for visitors state. When the incoming of new visitors is detected (by sensors), the agent ask the other agents for their previously achieved information (in this case, the context of the conversation) in order to use it later. Then, it decides what to say and generates the text to say that. Finally, it speaks the generated text and waits to visitor's response. The interpretation for the response is made by performing different processes on the recognized text as will be explained later. When the interaction is finished, the agent goes to a waiting state in which it can share its information with the other agents or wait for another visitor.

3 Other Natural Language Understanding Systems

The kind of language processing necessary for the Talking Agents in this scenario is rather similar to that made by the well-known chatbots, since they have

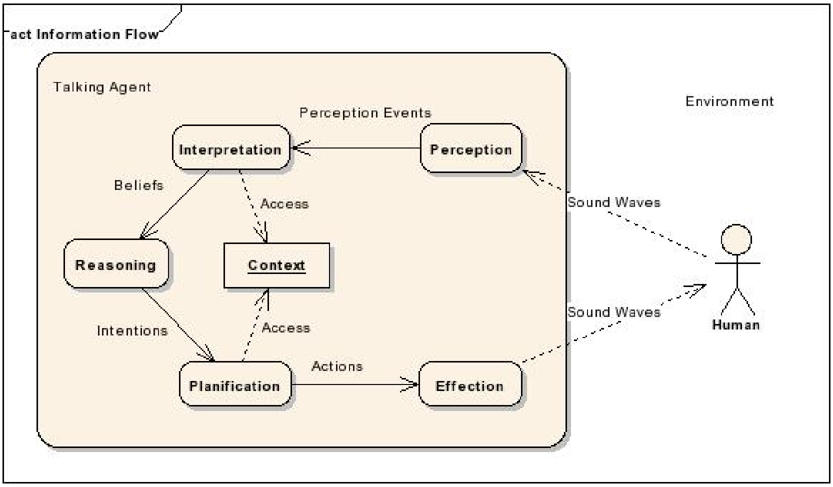


Fig. 1. Talking Agent information flow

to deal with completely arbitrary consultations, as most chatbots do. There are also chatbots oriented to certain domain or tasks, like assist users in e-commerce or e-learning [10], but their implementation use to be very similar to the more generalistic ones. The behaviour of this kind of systems often consists on identifying patterns or *keyphrases* in the introduced text in order to extract certain knowledge or initiate some sequence. These patterns may be specified using a domain-specific language, like AIML [13], which is the basis of one of the most famous chatbot systems, A.L.I.C.E. [14].

There are three main situations in which knowledge can be extracted using this pattern matching:

- **Pattern without variables:** a fixed fragment of text is identified in the input discourse, so the knowledge extracted is static, i.e. it is the same regardless the context of the phrase. For example: “hello”.
- **Pattern with variables without associated information:** a fragment of text matches the pattern, but the extracted variables do not have any associated information, so they can only be used literally. For example: “I like *”. The words identified this way could only be referenced in another generic phrase, like “I like * too”.
- **Pattern with variables with associated information:** a fragment of text matches the pattern, and the extracted variables have some extra information: it may be contained within an ontology structure, so the system could extract all its associated knowledge (properties, relationships, etc).

The knowledge extracted this way can be used the following ways:

- To generate the next phrase
- To store it as future reference
- To trigger and control the flow of certain sequence of questions

There exist also systems that are prepared to learn responses of their users, generating new patterns in order to use them in future conversations, like [9] and [1], and some of them make use of case-based reasoning, like [12], which is presented as a parameterized system, so the patterns can be specified by users to create the final systems. These systems are exposed to the general public so that they can learn faster (assuming that the users will contribute honestly). It is possible to consider also the context to improve the responses, as in [6][7][8].

There are also other concepts involved in the common chatbot systems, but this concept of pattern matching has been remarked because is the main bottleneck that prevents its use in the Talking Agents working scenario. Due to the multiple noise sources that could corrupt the user input (recognition errors, user syntactic errors, user abbreviations, external noise, etc) the pattern matching process become very inaccurate. For this reason, we consider a simpler approach, which offer much less impressive results, but is able to cope with the requirements.

4 Studying Spoken Natural Language Understanding Quality of Talking Agents

After identifying the multiple sources of error that distort the understanding in the Talking Agents scenario, in the following lines, it is described the processes adopted in order to improve the quality of this understanding, and the associated test results.

4.1 Effect of Recognition Quality and Alternatives

Due to the generalness of the used recognition component's [2] language model, it is common to obtain recognition results that are phonetically similar to the actual speech performed, but not exactly the same. Since this language model cannot be tuned, because the words a spectator can say are unpredictable, it is possible to slightly customize the results obtained by using some characteristics of the recognition component.

The first and simplest option is to extend the words dictionary by including those that are significant to perform certain interpretations. However, this customization hardly improves the overall quality of recognitions, since the new words are rarely identified, due to the lack of statistical information associated, compared with the default words contained in the language model.

Another common option is to make return more than one possible result, ordered by confidence level. Sometimes, the actual correct result is not given as first choice, but as second or third. And, even more often, there are no correct result among the choices, but all the choices together contain all the relevant words pronounced, mixed with other incorrect guesses.

Since the procedure to determine the topic of the conversation is to look for certain keywords within the set of those pronounced by the user, it may be interesting to catch as more words as possible, in order to increment the probability of finding one of the keywords. Nevertheless, this practice can also introduce some undesirable words, or noise, into the recognition results, increasing the probability of obtain a false interpretation.

For studying the effect of this procedure, the concept of completeness, correctness and quality has been used. The first tells about the percentage of words in the original speech that are contained in the results, the second tells about the percentage of words in the results that are contained in the original speech, and the third is a combination of both (the product). A perfect match between original and recognized speeches means a quality of 1.0.

In order to consider only the meaningful words, the tests have been performed after eliminating the stop words (the meaningless words) from the original and recognized speeches.

$$\begin{aligned}
\mathrm{M} &= \text{max number of alternatives} \\
j &\in \{1..\mathrm{M}\} \\
alt_j \in P(String) &= \text{j-th alternative given by the} \\
&\quad \text{recognition component as a set of words} \\
complete, correct, quality &: \{1..\mathrm{M}\} \times P(String) \rightarrow [0,1] \\
complete(i, original) &= \frac{|\bigcup_{j=1}^{i} alt_j \cap original|}{|original|} \\
correct(i, original) &= \frac{|\bigcup_{j=1}^{i} alt_j \cap original|}{|\bigcup_{j=1}^{i} alt_j|} \\
quality(i, original) &= correct(i, original) \cdot complete(i, original)
\end{aligned}$$

The results of the tests are shown in Fig. 2. The completeness of the result increases as we increment the number of alternatives used, while the overall quality decreases even faster. However, this may not be bad, since it is desirable to increase the probability of detect *at least one* topic. The problem comes if the correctness of the result becomes low enough to misunderstand the topic of the conversation. That threshold is to be determined taking into consideration the next phase of the perception, i.e. the interpretation.

In Fig. 3 we can see that, if we filter all but the useful keywords used to determine the topic of the conversation, the results are slightly better, because the keywords are less common and therefore less misunderstandables.

The inclusion of additional recognition threads produces an effect similar to the previous one. It increases completeness and decreases overall correctness and quality of results. However, it hardly changes the numbers obtained with the other procedure, so it is not considered for the agents to share the recognition results in case they are listening the same discourse.

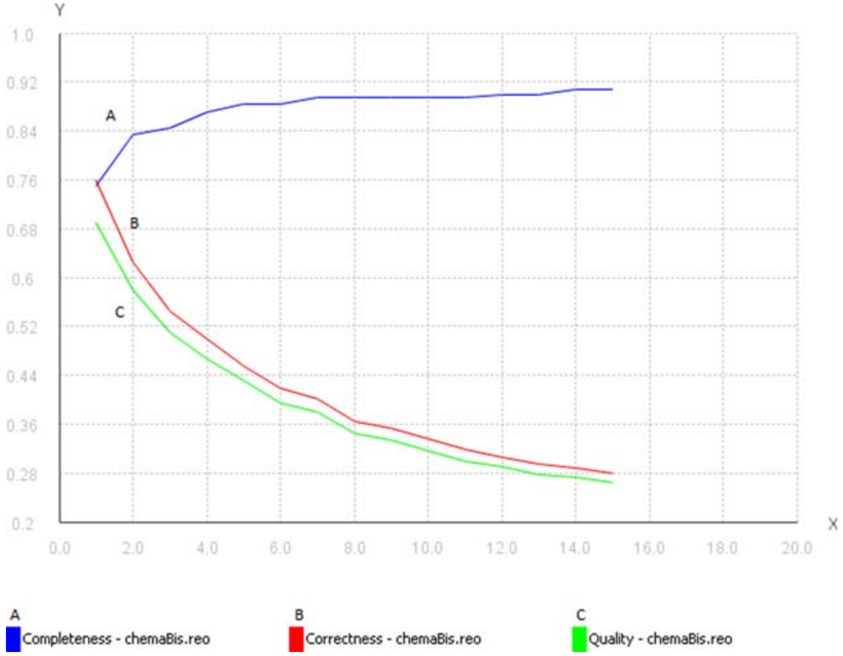


Fig. 2. Results of the recognition quality tests

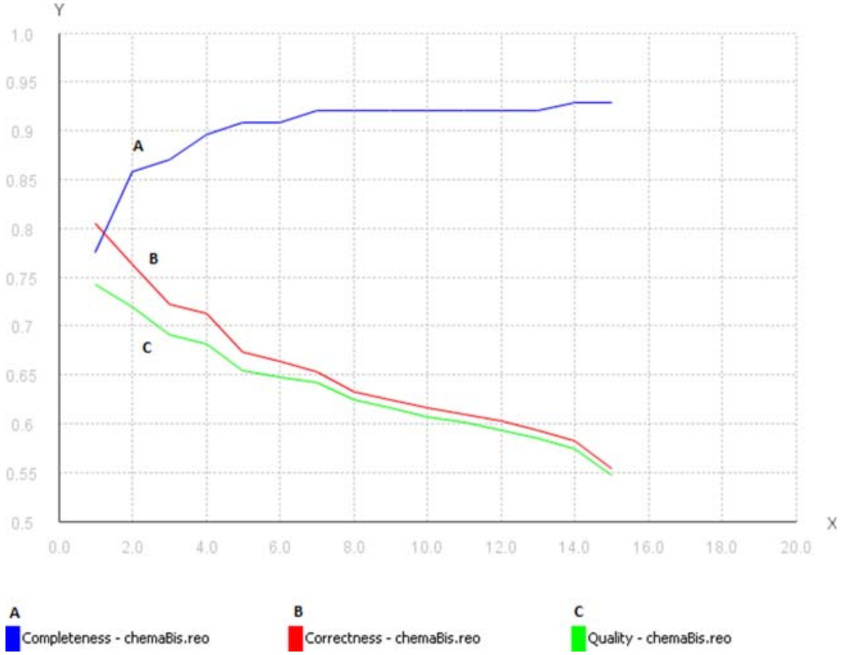


Fig. 3. Results of the recognition quality tests, considering only main keywords

4.2 Effect of Interpretation Quality and Alternatives

The interpretation component makes use of case-based reasoning technology in order to achieve the objective of determine the topic associated to a certain discourse. This paradigm has been chosen due to its flexibility in order to adjust the results of the reasoning, since it is possible to modify the structure of the case descriptions and the similarity functions without changing the rest of the system. Also, this paradigm is general enough to adopt very different interpretation strategies.

A simple interpretation strategy is used in the system, which is defined as follows: it is assumed that a discourse has a certain topic if and only if it contains one or more keywords related to that topic. That keywords are chosen so that if they are contained in a discourse, it *necessarily* relates to the corresponding topic, maybe among others (i.e. in the cases in which those words were contained, that topic always were one of the related).

Those keywords are called in this paper words with a *strong meaning*, opposing with words with *weak meaning*, which are found in different discourses without common topics. There are different grades of *weak words*, depending on the number of different topics they can relate independently. Thus, the *strong words* are considered to have grade 1, since they have only one possibility in order to determine the topic, and the *weak words* are considered to have grade 2 to N, where N is the max number of topics. Here, the grade represents the possibilities in order to determine the topics related by those words.

This system makes use of *strong words*, because of the simplicity of the processing, but it is considered as future work to include *weaker words*, taking into consideration that they must be treated differently in order to concede them less priority to decide the final topic of the discourse.

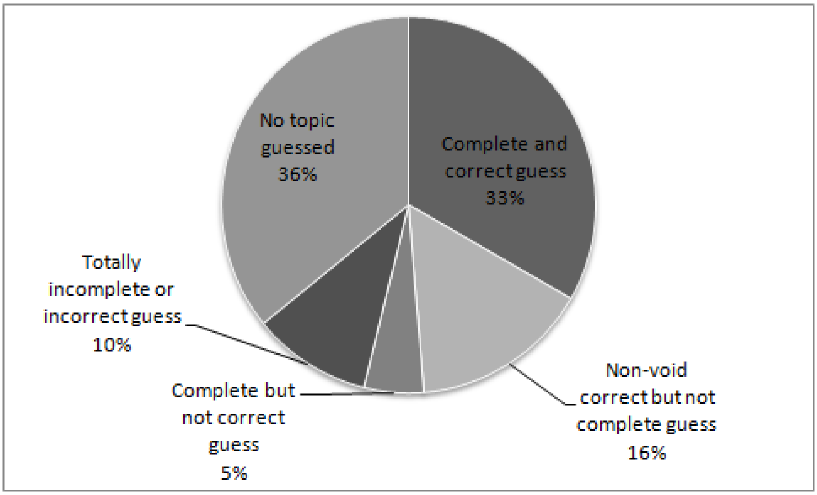


Fig. 4. Percentage of interpretation successes on the tests

In Fig. 4 is shown the results of interpretation made on the actual discourses spoken (i.e. without recognition error). The concepts of completeness and correctness have been also used. Both complete and correct guesses and correct but not complete guesses are desirable. Complete and not correct guesses may also be acceptable, but inaccurate. As we can see, the interpretation is yet weak mostly due to the amount of discourses in which topic cannot be determined. This circumstance may be solved by adding the weaker words to the interpretation process, since the inability to determine any topic comes from the inexistence of identified keywords in the discourse.

4.3 Impact of Recognition Quality on Interpretation

In graphic 5 we can see the impact of the recognition quality on an idealistic interpreter (i.e. the interpretation made by a person). The most of the interpretations (around 86%) could be made only using the first alternative. Around 11% of the interpretations needed the extra words contained in the second alternative, and barely 3% of interpretations needed the third or later alternatives in order to detect the topics correctly.

Also assuming the idealistic interpreter, no incorrect interpretation was made during tests due to inclusion of extra recognition alternatives, whilst it improves the probability of detecting a related topic, so we can assume that the use of extra alternatives in the interpretation of the topic should not be a problem. However, since there is no significant improvement detected after third alternative, this number may be used as threshold, in order to obtain better interpretation without reducing too much the quality of recognitions.

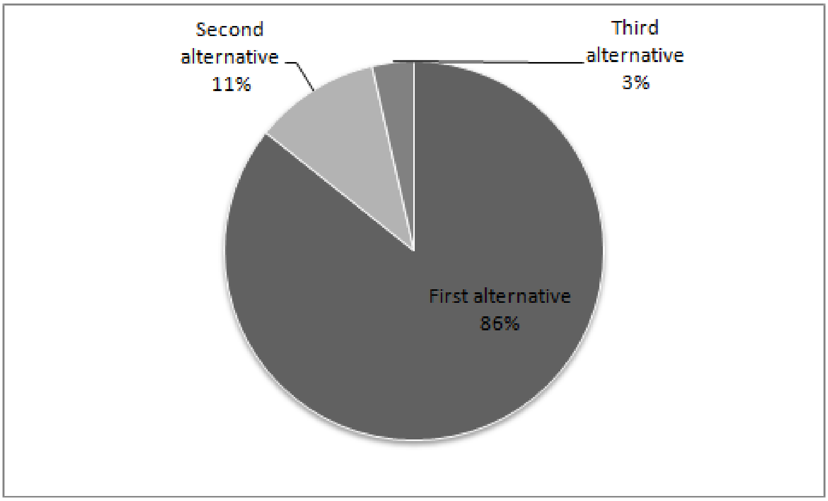


Fig. 5. Percentage of interpretations that needed each alternative to determine

5 Conclusions and Future Work

An alternative of chatbots systems for general domain language understanding has been studied in the context in which the inputs are contaminated by many sources of error. This alternative consists basically in identifying isolated words instead of patterns, since the structure of a pattern may be corrupted due to the errors. With this technique, the biggest bottleneck come from the phrases in which no topic is identified. For this reason, the addition of more alternatives in the recognition process has been tested in order to increase the amount of words obtained, and its impact on the interpretation process has been studied, concluding that it is slightly improved.

Also, a piece of future work has been identified, which consist in extending the interpretation process by including *weaker words*, to fill the gaps of the phrases in which no topic is identified, but this will make necessary to define an algorithm that gives preference to certain words, depending on its grade.

Acknowledgements. This work has been developed with support of the program "Grupos UCM-Comunidad de Madrid" with grant CCG07-UCM/TIC-2765, and the project TIN2005-08501-C03-01, funded by the Spanish Council for Science and Technology.

References

1. Cleverbot web site, http://www.cleverbot.com/
2. CloudGarden, TalkingJava web site, http://www.cloudgarden.com/JSAPI/
3. Corchado, J.M., Glez-Bedia, M., de Paz, Y., Bajo, J., de Paz, J.F.: Replanning mechanism for deliberative agents in dynamic changing environments. Computational Intelligence 24(2), 77–107 (2008)
4. Eckert, W., Kuhn, T., Niemann, H., Rieck, S., Scheuer, A., Schujattakamazzini: A spoken dialogue system for german intercity train timetable inquiries. In: Proc: European Conference on Speech Technology, pp. 1871–1874 (1993)
5. Fernández de Alba, J.M., Pavón, J.: Talking Agents Design on the ICARO Framework. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 494–501. Springer, Heidelberg (2009)
6. Griol, D., Sánchez, N., Carbó, J., Molina, J.M.: Context-Aware Approach for orally accessible Web Services. In: SAIAW Workshop on Soft approaches to information access on the Web, IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT09, pp. 171–174 (2009)
7. Griol, D., Arroyo, A., Patricio, M.A., Molina, J.M.: Spoken Dialogue Systems for Enhanced Interactions in Virtual Worlds. In: Third International Workshop on User-Centric Technologies and applications (MADRINET'09), pp. 45–52 (2009)
8. Griol, D., Snchez-Pi, N., Carb, J., Molina, J.M.: A Context-Aware Architecture to Provide Adaptive Services by means of Spoken Dialogue Interaction. In: International Conference on Artificial Intelligence (ICAI'09), 2009 World Congress in Computer Science, Computer Engineering, and Applied Computing WORLDCOM 2009, vol. II, pp. 912–918 (2009)
9. Jabberwacky web site, http://www.jabberwacky.com/

10. Mikic, F.A., Burguillo, J.C., Llamas, M., Rodrguez, D.A., Rodrguez, E.: CHARLIE: An AIML-based Chatterbot which Works as an Interface among INES and Humans. In: Proc. EAEEIE 2009 (2009)
11. Morfeo Community, ICARO Project Website, http://icaro.morfeo-project.org
12. Personality Forge web site, http://www.personalityforge.com/index.php
13. Wallace, R.S.: Artificial Intelligence Markup Language (AIML), http://docs.aitools.org/aiml/spec/
14. Wallace, R.S.: The Anatomy of A.L.I.C.E., http://www.alicebot.org/anatomy.html

Supervisory Control and Automatic Failure Detection in Grid-Connected Photovoltaic Systems

Fernando Agustín Olivencia Polo[1], Jose J. Alonso del Rosario[2], and Gonzalo Cerruela García[3]

[1] MAGTEL Systems Director and Scientific Colaborator in Departament of Computer and Numerical Analisys, University of Cordoba. C/Gabriel Ramos Bejarano, 114.
Polígono Industrial Las Quemadas, 14014 Córdoba (Spain)
fernando.olivencia@magtel.es

[2] Departamento de Física Aplicada, Universidad de Cádiz. Avda Rep. Saharaui s/n,
Puerto Real, 11510, Cádiz
josejuan.alonso@uca.es

[3] Departament of Computer and Numerical Analisys. University of Cordoba. Campus Universitario de Rabanales, Edificio C2, Planta-3. E-14071 Córdoba (Spain)
gcerruela@uco.es

Abstract. This paper is about a new online monitoring system for PV system operators based on the SCADA operational philosophy. The analysis module in the SCADA system manages the information using OLAP cubes, thus it allows improving the query of big data quantities and realize multidimensional analyses in an efficient way. For fail detection this paper proposes a procedure based on the comparison of the real measured production in the inverters regarding to the predicted production using others elements in the PV park.

Keywords: Supervisory Control and Data Acquisition (SCADA), renewable energy, intelligent systems, predictive maintenance, failure detection.

1 Introduction

At present there is a huge dependency on fossil fuels for energy production. This increases the emission of contaminating gases, insecurity caused by a possible absence of fuel supply and difficult cost predictability for the rapid change of energy market prices. To deal with this problem the use of renewable energies has been promoted [1]. In the European Union the accumulated and installed potency of photovoltaic (PV) and thermoelectric solar energy was multiplied by four between 2005 and 2008, up to 9.050 MWp.

Some renewable energies take as a principal problem presenting a high variability and dependency on the random nature of climatological phenomena. It is very important to know and predict the quantity of generated energy and the failures in generation systems, for a reliable use and successful integration into our daily energy supply. Legislation in many countries considers penalizations and economic bonuses to the producers of renewable energies according to the fulfillment grade of their production commitment.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 458–467, 2010.

To monitor and control a grid-connected PV system, generally a Supervisory Control and Data Acquisition System (SCADA) is used. Traditional SCADA systems with similar architectures have been described before for various applications [2-5]. Typically the data flow in a SCADA system is limited to the fixed field connections between hardware and a master station. Special applications such as the control of large PV systems need a wide range of devices and components physically connected. The SCADA system described in this paper is based on the concept of one server and multiple client architecture.

In recent years the interest of a great part of the scientific community has been focused on predicting possible fails to minimize the impact on transport networks. Regular performance checks on the functioning of grid-connected PV system need an advanced monitoring of the processes and the statistical analysis of the time series obtained using a large number of distributed sensors, data loggers or other intelligent monitoring devices, and requires intensive and high cost maintenance, only economical for large PV systems. To secure economical profits, effective methods to detect system failures in a rapid way are of paramount importance to increase the efficiency of a PV System.

Traditionally automatic failure detection in photovoltaic systems involves the registering of a great quantity of electrical variables as currents flowing through solar panels, voltages on batteries together with environmental data such as irradiance, temperature, etc. Some proposals try to reduce the complexity in the failure detection using few variables and more complex statistical analyses [6-8].

Other proposals do not use sensors connected to the elements that can fail. In this case, the inference is done by comparing the real production of the photovoltaic grid relative to the ideal production obtained from the irradiance information, the technical data of the installed devices, geographical localization, etc. In reference [9] this kind of system was implemented. The solar irradiance is derived on an hourly basis from the data of meteorological satellite Meteosat-8 by applying an enhanced version of the Heliosat method [10]. The operator of the system introduces the information to the major system components, manufacturer, type and number of modules; the real energy of a PV system is recorded for every hour and compared daily by the automated failure detection routine [11], this algorithm detects the occurrence of an error and informs the operator about it.

In this paper, we present an original SCADA system to monitoring and control grid-connected PV systems and a new approach to failure detection. The method is based on the analysis of the production series and in comparison to the production for other elements in the PV park. The proposed model allows, in relative terms, determining the elements that have failed most in the plant, comparing the real production vs. the predicted one by the model.

This work has been organized in the following way: In Section 2 the SCADA system to monitor and control de PV grid-connected plant is presented, describing its system architecture, and functionality. Section 3 describes a new method to detect failures and the experimental results are commented and analyzed. Finally, Section 4 describes the principal advances reached with the investigation and future lines.

2 Supervisory Control and Data Acquisition System (SCADA)

A photovoltaic park is composed of all the necessary elements to generate electric power, the following ones being essential:

- **Solar tracker (solar panels):** Receives the solar energy and transforms it into continuous electric current.
- **Inverter:** Transforms the continuous electric current generated by a trackers' group into alternating current which, after being conditioned appropriately in the centre of transformation, is injected into the distribution network of electricity.
- **Counting/Register:** This element takes into account the electric power produced by the solar plant, and using the stored registers, produces the invoicing for the distributing company of electricity.

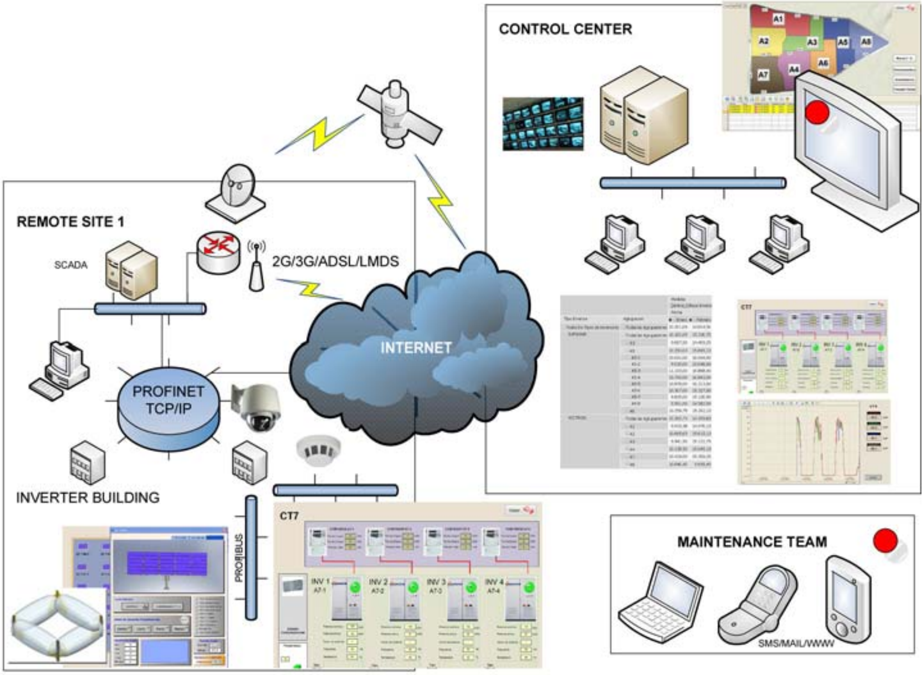


Fig. 1. SCADA System architecture

Figure 1 shows the architecture for the SCADA system proposed in this work and is composed by the following modules:

- **Management system:** This module allows monitoring the behavior of the grid connected PV system; it shows alerts when possible errors are detected, and permits the configuration and control of certain parameters in the park.
- **Distributed sensors:** A set of anemometers, temperature sensors, meteorological stations and irradiance sensors distributed in the solar park.

- **Master - slave control panels**: Every couple of trackers shares a small control panel to make in-situ the data acquisition (remote site) and to send control commands to the trackers.
- **Consolidation board:** They are located in the inverter's huts and concentrate the data from the solar trackers, from the inverters and from the sensors associated with this area.
- **Master Terminal Unit (MTU):** It has the function of collecting the data from all the consolidation boards and provides an HMI interface with the operator. It opens as a high accessibility cluster.
- **Analysis system:** This management system has been designed to process more than 30.000 variables in real-time. With the aim to analyze the time series corresponding to these variables, an On-Line Analytical Processing (OLAP) tool was developed [12].
- **Communications system:** It is composed of five networks:

a) *Main Ethernet TCP/IP*: This network is constituted by a ring of optical fibers that interconnects the MUT with the consolidation boards.

b) *Distribution network 1*: This network uses the standard of communications for field bus (PROFIBUS DP) and interconnects the consolidation board with the solar panels.

c) *Distribution network 2*: In this case a communications protocol of seven levels in the OSI model (MODBUS) is used. This protocol is based on the client/server architecture and it allows interconnecting the consolidation boards with the inverters and sensors.

d) *Master-Slave communications network*: Every couple of solar trackers has its own communication network MODBUS for supervision and control.

e) *3G-Link*: Allows remote access to the MTU using a console session or a web access; with this network it is possible to send SMS and e-mails to the operators indicating the system alarms.

The SCADA system proposed in this work has been implemented and proved. At present it is used in a photovoltaic plant supported by MAGTEL in the province of Cordova (Spain). This solar park is constituted by 1.342 solar trackers of two axes with a peak potency of 100 KW, 6,1 MWp in total.

This system allows us to monitor in real-time the production of the plant as well as detects the failure occurrence in solar trackers and inverters. However, it is not possible to identify internal errors in the plant design or in some elements such as fuses, individual photovoltaic modules or protection elements.

Using the information offered by the SCADA system properly it is possible to make general failure identification or to detect the malfunction of passive elements, and even to predict the energy production and the possible occurrence of failures in the plant. The next section describes the failure detection procedure implemented in the SCADA system presented.

3 Automatic Failure Detection

The failure or bad operation of one PV element is the deviation of the current behavior compared to expected functioning. In this context if it is possible to obtain a model of the behavior of PV elements, the numerical deviations in the variables of the system compared to the expected values to be an indicator of fail presence.

Another way to detect the presence of failure in the elements is based on finding significant discrepancies with respect to equal or very similar PV elements.

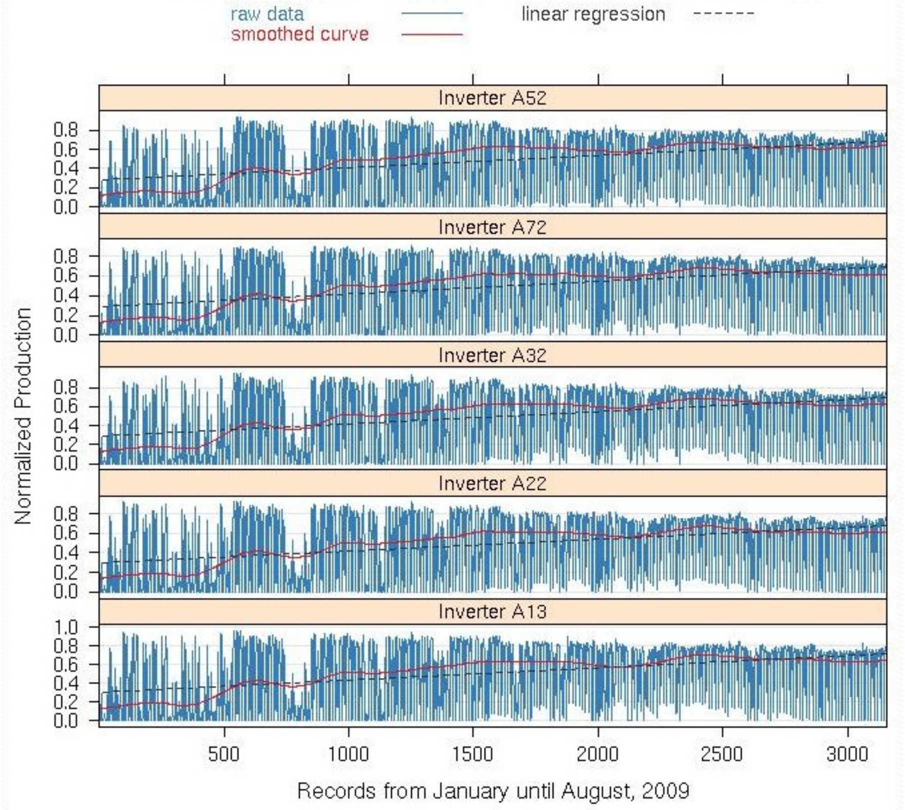


Fig. 2. PV Park normalized energy production from January until august of 2009

To represent the behavior (good or bad operation) of the PV park we can use the production of an element normalized compared to its own peak capacity production. In figure 2 the hourly time series corresponding to the normalized production of five elements in the PV park from January to August (2009) have been represented. If the PV elements were identical their production would be extremely similar, because they principally depend on the irradiance and ambient temperature, and both variables are almost uniform in the PV park.

We can use the cross-correlation of two time series as an indicator of similar behaviour for the variables that they represent. In normal operation conditions the correlation of variables representing the production of two elements in the PV park is practically equal to 1. When some element fails or is badly configured, a significant difference appears in its behavior, and the correlation value goes down. Figure 3 (left zone) shows the correlation matrix corresponding to the time series for 56 inverters production (1232 solar trackers) in the solar plant. On the right part in this figure the correlations for random selected inverters are shown.

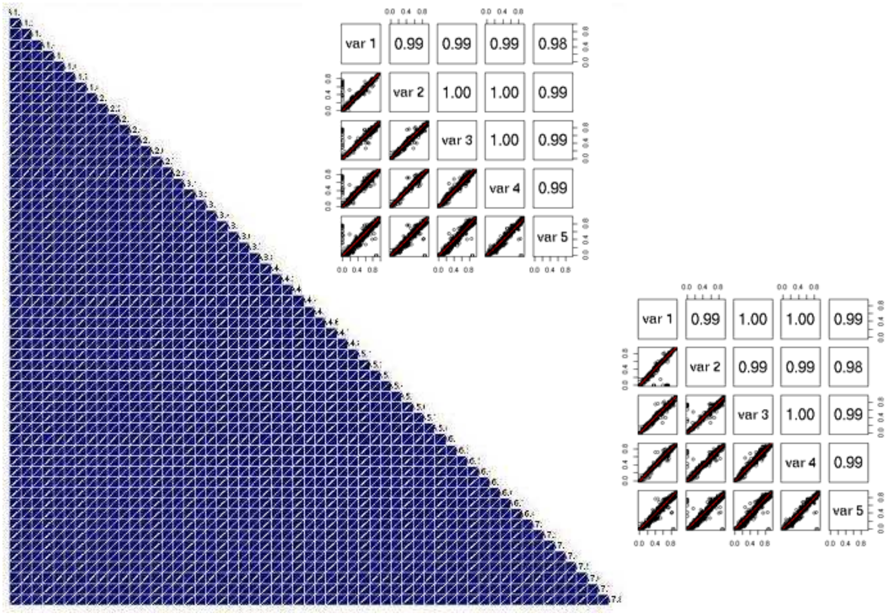


Fig. 3. Correlation Matrix

In the analysis the minimum value was found for the interrelation between the inverters A42 and A57. The linear regression corresponding to the series of inverter production A42 and A57 (figure 4) has been described by the following coefficients (table 1):

Table 1. Linear regression coefficients (A57 vs. A42)

R	**R^2**	**Adjusted R^2**	**Std. Error**	**Change Statistics**					**D.W.**
				R^2 Change	F Change	df 1	df2	Sig. Change	
0.967	0.936	0.936	0.0775	0.936	45863.77	1	3156	0.00	0.816

The production capacity of the inverter A57 is 93.6 % of the production capacity of the element A42, although it should be practically equal. For this reason we can predict that in the analyzed period, the inverter A57 has suffered a problem.

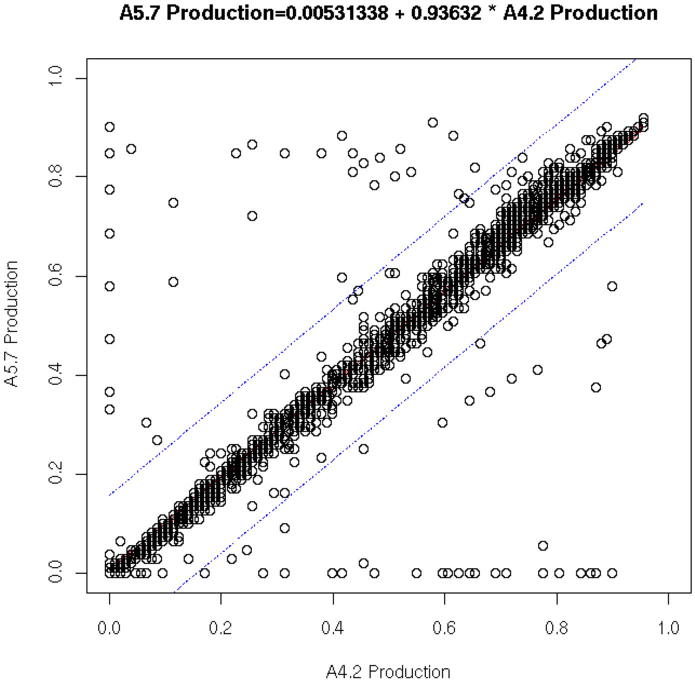


Fig. 4. Predicted production for A5.7 inverter using the A 4.2 inverter production

In the field tests to verify the proposed system for failure detection and classification, the maintenance workers determined, in situ, that there was an error in the configured operation thresholds for the inverter A57, this configuration bug was the main cause of the capacity difference with regard to the inverter A42.

3.1 Automatic Failure Detection Procedure

The proposed algorithm to detect failure is based on the comparison of the real measured production of the inverters with regard to the predicted production using others elements in the PV park . This routine not only detects the system malfunctions but also the provision of information about the most possible failure source and about the protocol to reach a rapid solution.

Figure 5 shows the principal operations in the failure detection procedure; at first some general parameters must be fixed, for example, the detection frequency parameter (ptfa) fix the period of time between two successive failure occurrence samplings, a high value for ptfa diminishes the answer time when a failure appears but it uses a minor number of samples in the prediction of the inverter production. The confidence interval in error detection (confl) reduces the uncertainties in the production estimation, and the parameter ways to notify bugs (wnb), set the priority level for failure and the alarm type in the SCADA system.

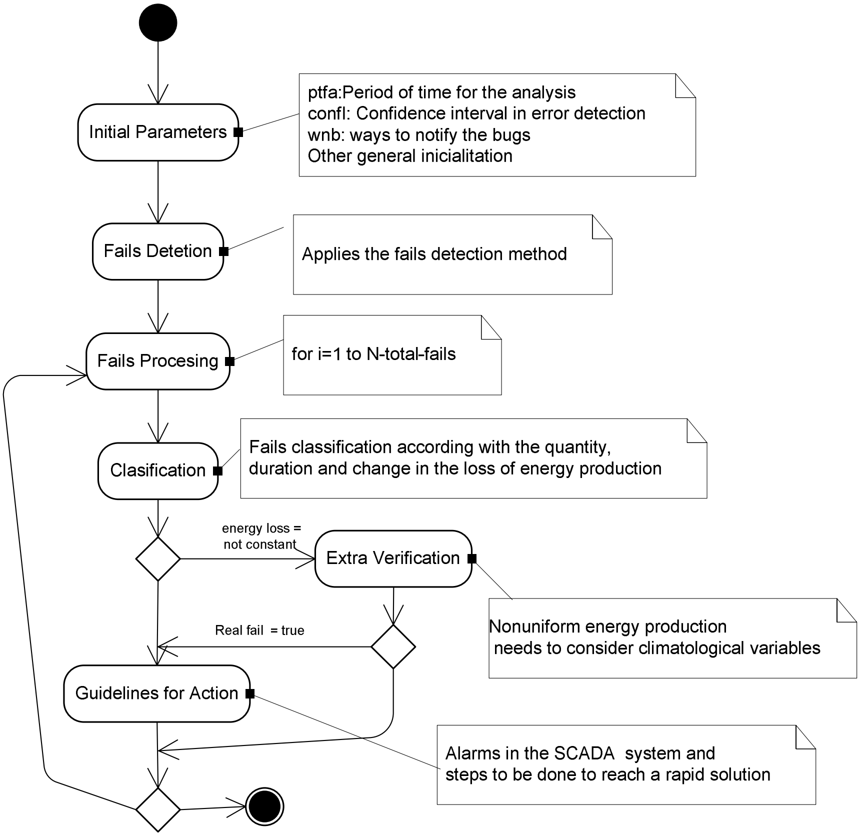

Initial Parameters
ptfa:Period of time for the analysis
confl: Confidence interval in error detection
wnb: ways to notify the bugs
Other general inicialitation
Fails Detetion
Applies the fails detection method
Fails Procesing
for i=1 to N-total-fails
Clasification
Fails classification according with the quantity, duration and change in the loss of energy production
energy loss = not constant
Extra Verification
Nonuniform energy production needs to consider climatological variables
Real fail = true
Guidelines for Action
Alarms in the SCADA system and steps to be done to reach a rapid solution

Fig. 5. Failure detection procedure

The failure detection process manages entire steps in the failure detection routine. It checks the real production with regard to predicted production identifying the inverters that are working incorrectly. For all the problematic inverters (N-total-fails) a classification failure procedure is applied. It analyzes some properties of the energy loss such as amount, duration and frequency to classify the failure. For example, inverter or control device defects produce a total blackout in energy production; a constant loss of energy can be caused by defects in some modules in the solar tracker, dirt in the photovoltaic cell or an error in the configuration operation thresholds for the inverter. When the loss of energy is not constant the failure are owing to low power losses, shadows, very high temperature or overheating in inverters.

It is possible to discriminate between the possible causes that generate no constant energy losses using the information coming from the meteorological stations and from the information about maintenance tasks coming from the OLAP module in the SCADA system. For these cases an additional cross-check is realized, discarding for example possible failures due to high temperatures when the real temperature is below 30 Celsius degrees or false bugs due to partial disconnections to carry out maintenance tasks.

Finally, the failure detection procedure proposes an action plan according to the fail classification and enables the alarms in the SCADA system.

4 Conclusions and Future Works

A supervisory control for grid-connected photovoltaic system has been developed and proved, with this system it is possible to monitor and analyse system behaviour in photovoltaic parks. For example, we can study the influence of weather in energy production, module temperature effects on the system performance, monitor solar radiation to improve photovoltaic efficiency, etc. The SCADA system proposed in this paper manages the information using OLAP cubes, thus it allows improvement of the query of big data quantities and carries out multidimensional analyses in an efficient way.

The possibility to estimate the energy production for one element knowing an other production series, allows us to detect the presence of a failure, using a procedure consisting in the comparison of the real measured production for the element with regard to the predicted production using the rest of elements in the plant. The proposed procedure not only detects the system malfunctions but also provides information about the most probable failure source, the protocol to reach a rapid solution and set off the alarms in the SCADA system sending emails and SMS to the responsible operators.

At present we are improving the SCADA System using it for other renewable energy sources, such as wind farms, and extending the functionality to predict the energy production according to meteorological variables.

References

1. Singh, R., Sood, Y.R.: Polices for promotion of renewable energy sources for restructured power sector. In: Proceedings of the 3rd International Conference on Electric Utility Deregulation and Reconstructing and Power Technologies, 1-5 Nanjuing China (2008)
2. Cardoso, F.J.A.: A universal system for laboratory data acquisition and control. IEEE Trans. Nucl. Sci. 472, 154–157 (2000)
3. Singlachar, R., Mukherjee, B.: An advanced PC-PLC-based SCADA system for a commercial medical cyclotron. Nucl. Instrum. Methods Phys. Res. A 399, 396–406 (1997)
4. Dieu, B.: Application of the SCADA system in waste-water treatment plants. ISA Trans. 40, 267–281 (2001)
5. Patel, M., Cole, G.R., Pryor, T.L., Wilmot, N.A.: Development of a Novel SCADA System for Laboratory Testing. ISA Transactions 43(3), 477–490 (2004)
6. Guasch, D., Silvestre, S., Calatayud, R.: Proceedings of 3rd World Conference on Automatic failure detection in photovoltaic systems, pp. 2269–2271 (2003)
7. Silvestre, S., Guasch, D., Goethe, U., Castarier, L.: Improved PV battery modeling using Matlab. In: Proc. of the Seventeenth European Photovoltaic Solar Energy Conference and Exhibition, Munich, Germany, pp. 507–509 (2001)
8. Guasch, D., Silvestre, S.: Dynamic battery model for Photovoltaic applications. Progress in Photovoltaics: Research and applications 11, 193–206 (2003)

9. Drews, A., Keizer, A.C., Beyer, H.G., Lorenz, E., Betcke, J., van Sark, W.G.J.H.M., Heydenreich, W., Wiemken, E., Stettler, S., Toggweiler, P., Bofinger, S., Schneider, M., Heilscher, G., Heinemann, D.: Monitoring and remote failure detection of grid-connected PV systems based on satellite observations. Solar Energy 81(4), 548–564 (2007)
10. Hammer, A., Heinemann, D., Hoyer, C., Kuhlemann, R., Lorenz, E., Mueller, R.W., Beyer, H.G.: Solar energy assessment using remote sensing technologies. Remote Sensing of the Environment 86, 423–432 (2003)
11. Stettler, S., Toggweiler, P., Wiemken, E., Heydenreich, W., de Keizer, A.C., van Sark, W.G.J.H.M., Feige, S., Schneider, M., Heilscher, G., Lorenz, E., Drews, A., Heinemann, D., Beyer, H.G.: Failure detection routine for grid-connected PV systems as part of the PVSAT-2 project. In: Proceedings of the 20th European Photovoltaic Solar Energy Conference & Exhibition, Barcelona, Spain, pp. 2490–2493 (2005)
12. Olivencia Polo, F.: On-Line Analytical Processing in Photovoltaic System. MAGTEL Systems Technical report MGTSYS0508, Cordoba, Spain (2008)
13. Emrich, C., Acosta, R., Kalu, A., Kifer, D., Wilson, W.: Supervisory Control and Data Acquisition Experiment Using the Advanced Communications Technology Satellite. Techical report FSEC-CR-1296-01, Florida Solar Energy Center/University of Central Florida (2001)

Data-Driven Prognosis Applied to Complex Vacuum Pumping Systems

Florent Martin[1,2], Nicolas Meger[1], Sylvie Galichet[1], and Nicolas Becourt[2]

[1] University of Savoie, Polytech'Savoie, LISTIC,
Domaine Universitaire BP80439, 74944 Annecy-le-Vieux, France
{florent.martin,nicolas.meger,sylvie.galichet}@univ-savoie.fr
[2] Alcatel Vacuum technology
98 Avenue de Brogny, 74009 Annecy, France
{florent.martin,nicolas.becourt}@adixen.fr

Abstract. This paper presents a method to address system prognosis. It also details a successful application to complex vacuum pumping systems. The proposed approach relies on an automated data-driven learning process as opposed to hand-built models that are based on human expertise. More precisely, using historical vibratory data, we first model the behavior of a system by extracting a given type of episode rules, namely First Local Maximum episode rules (FLM-rules). A subset of the extracted FLM-rules is then selected in order to further predict pumping system failures in a datastream context. The results that we got for production data are very encouraging as we predict failures with a good time scale precision. We are now deploying our solution for a customer of the semi-conductor market.

Keywords: Data mining, episode rules, predictive maintenance, prognosis, vibratory data.

1 Introduction

In the current economic environment, industries have to minimize production costs and optimize the profitability of equipments. Fault prognosis is a promising way to meet these objectives. Early detection of system behavior deviation indeed permits a better management of production means by anticipating rather than undergoing failures. Most of the prognosis applications come from medicine [1] or aerospace [8, 12]. In medicine, prognosis is defined as "the prediction of the future course and outcome of disease processes, which may either concern their natural course or their outcome after treatment" [1]. In aerospace domain, prognosis is defined as detecting the precursor signs of a system misfunction and predicting how much time is left before a major failure [12]. Research works in aerospace are very close to our application. We thus adopt the same point of view about prognosis objectives.

Because of their complex kinematic, the kind of vacuum pumping systems we want to monitor may have high vibration levels even if they are in good running

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 468–477, 2010.

conditions. It is thus difficult to develop monitoring methods based on expert knowledge or physical models that would rely on signal thresholding. In this challenging context, a data-driven approach is preferred. In industrial applications, most of data-driven approaches are neural network-based [12]. Unfortunately, neural networks are "limited by their inability to explain their conclusions" [4]. Therefore we propose to extract symbolic rules, namely episode rules [9], from a long sequence of production data, for describing systems behavior. More precisely, we propose to extract First Local Maximum-rules (FLM-rules) as defined in [10]. FLM-rules are episode rules having an optimal temporal window width, which means that these rules are likely to appear within a specific temporal window width. This temporal window width can vary from a rule to another. In opposition to neural networks, such kind of rules is more interpretable. A subset of these rules is then used as a rule-based system to proceed to prognosis, i.e. to prognose seizings of pumping systems when considering our industrial application. This subset is generated by selecting the most predictive FLM-rules. Our contribution can be summarized as:

– running a recent data mining algorithm extracting FLM-rules from industrial vibratory data,
– providing a method for selecting the most predictive FLM-rules,
– providing a method for merging FLM-rules temporal information in order to prognose failures in a precise temporal window.

This paper is structured as follows: Section 2 reviews existing works in data mining applied to prognosis. Section 3 introduces our industrial application and describes the way we select and preprocess data. Section 4 presents the notion of FLM-rules while Section 5 details our most predictive rules selection process. In Section 6, we explain how to proceed to real time prediction, which includes matching FLM-rules in a data stream and merging their predictions. Finally, experimental results are presented in Section 7 while Section 8 concludes and draws perspectives.

2 Related Work

Although maintenance is a manufacturing area that could benefit a lot from data mining solutions, few applications have been identified [5, 12] so far. Most of them are diagnosis applications [12] and mainly relate to aerospace [8] or to network analysis applications [6]. Prognosis applications often meet difficulties in predicting how much time is left before failure. End-users indeed have to set the width of the temporal window that is used for learning a model and for predicting failures [3].

In [8], Letourneau and al. present an approach that aims at predicting aircraft component faults. They rely on data mining techniques to build models from historical data. These models are then used to predict failures. More precisely, for each component, heterogeneous data (numerical, text) originating from measurements and maintenance reports are collected. Learning datasets are then defined according to the component replacement occurrences found in maintenance reports by selecting the data that have been recorded before and after replacements. Failure periods are defined by considering temporal windows (about 10% of the datasets time scale) that start just before replacement dates. All temporal aspects are user-defined. Classifiers are finally extracted using decision tree, nearest-neighbor or naive-bayes techniques. It is to note that the choice of the width of the failure periods significantly impacts results as it is used both for modeling and for predicting failures.

In [6], a data mining application, known as the *TASA project*, is presented. It aims at extracting episode rules that describe a network alarm flow. These rules syntactically take the temporal aspect into account. They are known as *episode rules*. They are selected according to a frequency (or support) measure and a confidence measure (for more formal definitions, the reader is referred to [6]). In practice, users have to set the maximum time span of the episode rule occurrences so as to make extractions tractable. Episode rule occurrences whose time span is greater than this maximum time span are indeed not considered. This approach can still generate a large amount of rules and experts are required to browse interesting rules according to their knowledge. Though this approach has not been designed for performing prognosis, it inspired some works that actually aim at predicting failures. For example, in [3], the authors propose a method that searches for previously extracted episode rules in datastreams. As soon as an episode rule is recognized, it is used for predicting future events. More precisely, they propose to build and to continuously maintain a queue of events which are likely to contribute to occurrences of episode rules. Once the premiss of a rule is matched, they proceed to prediction by adding the maximum time span to the occurrence date of the first symbol of the episode rule. This method is interesting as it avoids scanning back the entire datastream but still, a crucial temporal information, i.e. the maximum time span, has to be set by users. Moreover, it has only been tested on synthetic data.

Beside asking for temporal parameters, the methods that are reviewed in this section all impose a same temporal window width (or maximum time span) for all possible models, for all possible rules. However, it is quite difficult to justify the use of a same maximum time span for each rule. It has thus been proposed in [10] to extract *FLM-rules*. These episode rules indeed come along with their respective *optimal temporal window sizes*. In [11], we proposed a preliminary approach for using optimal window sizes so as to establish prediction dates. An experiment on synthetic datasets was also presented. In this paper, we propose an improved version of this technique. We also describe a successful prognosis experiment that has been run on real production datasets acquired from vacuum pumping systems.

3 Application and Data Preprocessing

The aim of the approach described in this paper is to generate a valid model to identify abnormal behaviors of complex vacuum pumping systems (i.e. with a complex kinetic) and to predict a major default mode. Those systems are running under really severe and unpredictable conditions. They basically transfer gas from inlet to outlet. One major default mode that can alterate this function is the seizing of the pump axis. Seizings can be provoked by many causes such as heat expansion or gas condensation. Preventive maintenance plannings are not efficient. Therefore, Alcatel Vacuum Technology initiated a predictive maintenance project. With this aim in view, the quadratic mean (Root Mean Square, denoted *RMS*) of the vibration speed over 20 frequency bands has been collected over time at a frequency of 1/80 Hz. In real vibratory environments, vibration signals follow a zero-mean gaussian distribution. RMS is thus the standard deviation of the vibration speed [7]. It is measured by accelerometers and it is equal to the power of the vibration speed. Available data cover more than 2 years for 64 identical pumping systems. In order to build the learning dataset, we selected data that directly relate to some serious and fairly common failures we want to anticipate, i.e. first seizings. More precisely, we decided to build our learning dataset by only considering the data that are acquired before the first occurrence of such a failure. Indeed, after the first seizing, systems are seriously damaged and learning from potentially degraded systems would bias models. Moreover, if we succeed in predicting seizings, we will not observe such conditions. So we build learning sequences that start at the system startup date and end at the first failure occurrence. We got 13 doubtless sequences that end with the first occurrence of a seizing.

In order to detect evolutions of systems, for each system, we first determine the signal signature corresponding to good running conditions. The measured signal is then compared to this reference. Measurements represent the power the equipment is subjected to. It can be decomposed as follows: $P = P_0 + P_d$, with P the measured power, P_0 the power in good running conditions and P_d the power of the default. According to experts, P_0 is defined for each frequency as the standard deviation calculated 24 hours after system power-on (so as to get a stable signal). It is calculated using sliding windows. Each time a new measure arises, P_0 is updated if the computed value is lower than the previous one. It generally takes up to 2 weeks to get a stable P_0.

The kind of default we are looking for generates a high level of power (P_d is high). Thus, for each frequency band we decide to focus on the values of the maximal envelope of measured power P. In order to perform FLM-rule extraction, the maximal envelope has to be encoded into symbols. First, to qualify the default severity, the ratio P/P_0 is computed and discretized using three levels. Then, we define a dictionary of 240 symbols, each symbol being associated with three pieces of information: the frequency band, the default severity and the duration at that severity level. We also introduce a specific symbol to represent seizing occurrences. Finally, we got 13 sequences containing 2000 symbols on average along with their occurrence dates. It is thus not recommended to use standard algorithms that extract patterns from a *collection of short sequences* (i.e. *base of sequences* [2]). Indeed short sequences generally do not contain more than about 500 events. Pattern extraction from a single *large sequence* [6, 10, 9] is then focused on. 13 sequences were thus concatenated into a single large sequence D. In order to avoid the extraction of FLM-rule occurrences spreading over various subsequences Seq_i, a large time gap between each initial sequence is imposed (see Fig. 1) and the *WinMiner* algorithm [10] is used to extract FLM-rule occurrences. WinMiner can indeed handle a maximum time gap constraint (denoted as *maxgap*) between occurrences of symbols.

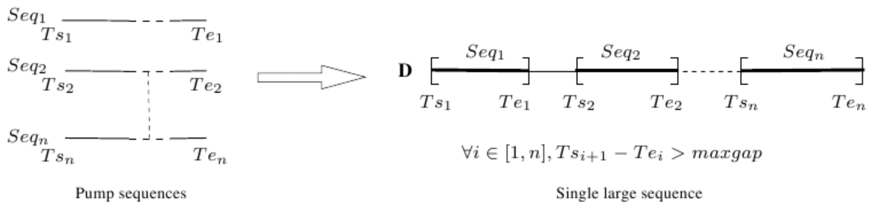


Fig. 1. Sequence building

4 FLM-Rules

In this section, we succinctly introduce the notion of FLM-rules. For formal definitions, reader can refer to [10]. Let us first consider the data we are dealing with, i.e. *events*. An event is a time-stamped occurrence of a symbol. When gathering those events, one can build an *event sequence* which is thus a sequence of symbols that come along with their respective occurrence dates. The whole sequence is ordered with respect to those occurrence dates. FLM-rules are patterns that occur in such a dataset. They are build upon *serial episodes* which are totally ordered sequences of symbols. For example, $A \rightarrow B$ is a serial episode stating that symbol B occurs after symbol A. Occurrences of such episodes are time intervals of the form $[t_s, t_e]$ with t_s the occurrence date of its first symbol (A for episode $A \rightarrow B$) and t_e the occurrence date of its last symbol (B for episode $A \rightarrow B$). In order to meet applicative needs and to make extraction tasks tractable, *minimal occurrences* are considered [9]. A minimal occurrence of an episode α is a time interval $[t_s, t_e]$ containing α and such that there is no other occurrence $[t'_s, t'_e]$ verifying $[t'_s, t'_e] \subset [t_s, t_e]$. Furthermore, episodes having at least two consecutive events that are separated by more than *maxgap* units of time are discarded. *maxgap* is user-defined. Let $A \rightarrow B$ and $A \rightarrow B \rightarrow C$ be two serials episodes. Using these episodes, one can build the episode rule $A \rightarrow B \Rightarrow C$. It means that if premiss $A \rightarrow B$ occurs in an event sequence, then the conclusion C is likely to occur. Episodes rules are characterized by two measures:

- the *support*: the number of occurrences of an episode rule over the whole sequence. The support of $A \rightarrow B \Rightarrow C$, denoted by $support(A \rightarrow B \Rightarrow C)$, is equal to the number of occurrences of episode $A \rightarrow B \rightarrow C$, denoted by $support(A \rightarrow B \rightarrow C)$.

- the *confidence*: the observed conditional probability of observing the conclusion of an episode rule knowing that the premiss already occurred. Confidence of $A \rightarrow B \Rightarrow C$ is thus defined as follows: $confidence(A \rightarrow B \Rightarrow C) = \frac{support(A \rightarrow B \rightarrow C)}{support(A \rightarrow B)}$.

Those measures are used for selecting episode rules according to a minimum support threshold σ and a minimum confidence threshold γ. As proposed in [10], support and confidence can be defined for each *window width*, i.e. the maximum time span of episode rule occurrences. If, for a given episode rule λ, and for the shortest possible window width w,

- the support of λ is greater or equal to σ,
- the confidence c_w of λ is greater or equal to γ,
- there exists a window width $w'|w' > w$ such that confidence of λ for w' is $decreaseRate$ % lower than c_w,
- there is no window width between w and w' for which confidence is higher than c_w,

then, episode rule λ is said to be a *First Local Maximum-rule* or *FLM-rule*. Parameter $decreaseRate$ is user-defined and allows for selecting more or less pronounced local maxima of confidence with respect to window widths. The window width w corresponding to a first local maximum is termed as the *optimal window size* of FLM-rule λ. This can be interpreted as: "if I observe the premiss of λ at t_0, then the probability of observing its conclusion is very high beetwen t_0 and $t_s + w$. If γ is set to 100%, then its conclusion must appear within t_0 and $t_s + w$". Using FLM-rules makes sense when dealing with temporal vibratory data. Indeed, each FLM-rule we extracted in the context of our application had a unique pronounced local maximum that differs from one rule to another.

5 Rule Selection

FLM-rules can be extracted using the *WinMiner* algorithm that is proposed in [10]. Extracted FLM-rules give us a description of the behavior of the vacuum pumping systems. However, as we aim at predicting seizings, we only retain FLM-rules concluding on the symbol "seizing". Furthermore, we select a subset of these rules, the most predictive ones, in order to perform prognosis. Our selection process is inspired from the well known *leave-one-out cross validation* technique. It involves considering alternatively each subset Seq_i (see Fig. 1), ending with a failure, as a validation set while other subsets form the learning set. We thus alternate FLM-rules extractions (with the same parameter values) to get descriptions from the training set and validations of these descriptions by matching extracted FLM-rules. A FLM-rule is matched in the validation set if it occurs and if its occurrence time span is lower or equal to its optimal window size. Once our selection process ends, we get a set of FLM-rules that do not trigger any false alarm on validation sets. A same FLM-rule can be extracted at several iterations, each iteration providing a different optimal window width.

In this case, its optimal window size is set to the most observed value. The set of selected FLM-rules is termed as the *FLM-base*. Back to our application, as seizings can originate from very different causes, and as we only have 13 subsets relating to a seizing, the minimum support threshold σ is set to 2. In order to extract the most confident rules, the minimum confidence threshold γ is set to 100%. Finally, the parameter *decreaseRate* is set to 30% to select pronounced/singular optimal window sizes and the *maxgap* constraint is set to 1 week to consider very large optimal window sizes. Indeed, when searching for the optimal window size of an episode rule, confidence and support measures are computed for window widths that are lower or equal to the number of events of the rule multiplied by the *maxgap* constraint.

6 Real Time Prediction

Real time prediction first relies on the matching of FLM-rule premisses. Then, for each matched premiss, a time interval within conclusions/seizings should occur is computed. Those aspects are detailed in Section 6.1. As many different premisses can be matched, we may get different time intervals for a single failure occurrence. We thus propose in Section 6.2 a method for merging these informations and for providing a single time interval, namely the *forecast window*.

6.1 Matching FLM-Rules Premisses in Data Streams

In order to match premisses of the FLM-rules belonging to the FLM-base, we build a queue of event occurrences whose time span is lower than W, the largest optimal window width of the FLM-base. This queue is maintained by removing events whose occurrence date t' is lower than $t - W$, with t the current system time. We thus make sure that enough data are kept for being able to identify the premisses of all the FLM-rules that form the FLM-base. Each time a new event occurs, it is added to the queue. If that one corresponds to the suffix (last event) of the premiss of a FLM-rule r belonging the the FLM-base, a premiss matching process is launched. We scan the queue through an observation window $W_r^o =]t_r, t_0[$ with t_0 the date at which rule matching process is launched and $t_r = t_0 - w_r$, with w_r the optimal window width of rule r. Indeed, in the worst case scenario, the conclusion of rule r is about to occur at $t_0 + 1$ and the earliest date of occurrence of the first symbol of its premiss is $t_0 + 1 - w_r = t_r + 1$. As proposed in [3], in this observation window, we search for the latest minimal occurrence of the premiss of r which amounts in finding the occurrence date of its first event, denoted ts_r and defined as follows: let Ts_r be the set of the occurrence dates of the first event of the premiss occurrences of rule r that occurs in $]t_r, t_0]$. The date ts_r is the single element in Ts_r such that $\nexists\ t \in Ts_r$ with $ts_r \neq t$ and $t > ts_r$ and such that the conclusion of rule r does not appear in $]ts_r, t_0]$. Then, by definition of FLM-rules, we forecast the conclusion of rule r in $w_r^f =]t_0, tc_r]$ with $tc_r = ts_r + w_r$. By construction $t_0 < tc_r < t_0 + maxgap$. Figure 2 illustrates this process for rule $A \rightarrow B \rightarrow C \Rightarrow P$.

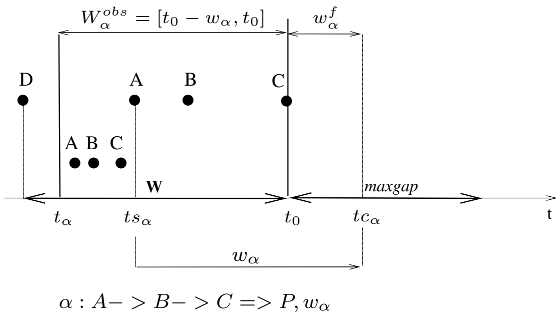

$W_\alpha^{obs} = [t_0 - w_\alpha, t_0]$
w_α^f
D
A
B
C
A B C
W
maxgap
t_α
ts_α
t_0
tc_α
t
w_α
$\alpha : A- > B- > C => P, w_\alpha$

Fig. 2. Matching rule $A \rightarrow B \rightarrow C \Rightarrow P$

6.2 Merging FLM-Rules Predictions

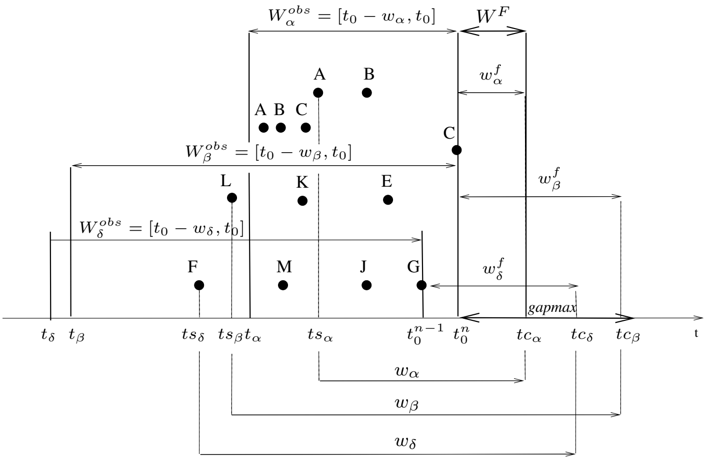
Let t_0 be the date at which a FLM-rule premiss is matched. For each matched premiss of rule r, its conclusion should occur in $w_r^f =]t_0, tc_r]$ with 100% confidence (if the minimum confidence is set to 100%). Let Tc be the set of all prediction dates tc_r that are active, i.e. that are greater than t_0 (those prediction dates can be computed before and at t_0). The associated failure prediction time interval $]t_s^f, t_e^f]$, also termed as the forecast window W^F, is such that $t_s^f = t_0 \wedge t_e^f = min(Tc)$. By choosing the min operator to aggregate prediction dates, this forecast window is defined to be the earliest one. Figure 3 provides a forecast window established using rules α, β, δ that have been recognized at t_0^{n-1} and t_0^n.

7 Experimental Evaluation

In order to evaluate the accuracy of our forecast, we consider two cases:

- each time our forecast method foresees a seizing, we check if seizing really occurs in the given forecast window $W^F =]t_s^f, t_e^f]$.
- each time t_0 a new event arises, and if no warning is triggered, we check if a seizing occurs in $]t_0, t_0 + maxgap]$. We extracted rules under $maxgap$, a maximum time gap between events. We thus can not predict any occurrence of conclusions of FLM-rules after $t_0 + maxgap$.

We applied the real time prediction method on 2 datasets: the 13 sequences used to build our FLM-base (dataset 1) and 21 new sequences of production data (dataset 2). Using these datasets, we simulated 2 datastreams and made respectively 24125 and 32525 forecasts. Results of evaluations are given in Table 1 and in Table 2, using confusion matrices. We denote $\widehat{failure}$ the number of forecasts stating that pump will seize and $\widehat{healthy}$ the number of forecasts that do not foresee anything. Though we could access few data relating to failures so far, results are encouraging as we predicted 10 seizings out of 13 with 99, 97% of



$\alpha : A- > B- > C => P, w_{\alpha}$

$\beta : L- > K- > E- > C => P, w_{\beta}$

$\delta : F- > M- > J- > G => P, w_{\delta}$

Fig. 3. Merging prediction information of FLM-rules

accuracy on dataset 1 and as we foresaw 2 upcoming seizings with 98, 75% accuracy on dataset 2. The 262 false-positive forecasts correspond to the 3 seizings that we did not predict. Furthermore, the 20 and 404 false alarms (tables 1 and 2) stating that pump will seize, were generated on pump that really seized few days later. Though the forecast window was not precise enough, diagnostic was right. Earliest failure predictions provided by our software prototype arise at the latest 3 hours before failure really occurs and, most of the time, more than 2 days before. This is enough to plan an intervention. We are now deploying the proposed approach for a client of the semi-conductor market.

Table 1. Results for dataset 1

	failure	*healthy*
failure	492	262
healthy	20	23351

Table 2. Results for dataset 2

	failure	*healthy*
failure	300	0
healthy	404	31821

8 Conclusion and Perspectives

In this paper, we present a data mining approach for modeling pumping systems by means of FLM-rules and for forecasting failures using a subset of these rules, namely the FLM-base. Using vibratory data, we applied this approach in an industrial context in which pumping systems are running under severe and unpredictable conditions. Results are encouraging as we forecast failures with a good accuracy, i.e. more than 98% on both learning data and new data. Moreover, using our predictions, enough time is left to technical teams to plan an intervention. Future work directions include introducing fuzzy logic for merging prediction dates so as to provide end-users with gradual warnings.

References

[1] Abu-Hanna, A., Lucas, P.J.F.: Prognostic models in medicine ai and statistical approaches. Jr. of Methods of Information in Medicine 40, 1–5 (2001)

[2] Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of the 11th Intl. Conf. on Data Engineering, pp. 3–14 (1995)

[3] Cho, C., Zheng, Y., Chen, A.L.P.: Continuously matching episode rules for predicting future events over event streams. In: Dong, G., Lin, X., Wang, W., Yang, Y., Yu, J.X. (eds.) APWeb/WAIM 2007. LNCS, vol. 4505, pp. 884–891. Springer, Heidelberg (2007)

[4] Coiera, E.: Automated signal interpretation. Technical report, Hewlett-Packard Lab. (1993)

[5] Harding, J.A., Shahbaz, M., Kusiak, A.: Data mining in manufacturing: a review. Jr. of Manufacturing Science and Engineering 128(4), 969–976 (2006)

[6] Hatonen, K., Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H.: Tasa: Telecomunications alarm sequence analyzer or: How to enjoy faults in your network. In: Proc. of the IEEE Network Operations and Management Symposium, pp. 520–529 (1996)

[7] Jens, T.B.: Mechanical Vibration and Shock Measurements. Bruel & Kjaer (1973)

[8] Letourneau, S., Famili, F., Matwin, S.: Data mining for prediction of aircraft component replacement. IEEE Intelligent Systems and their Applications 14(6), 59–66 (1999)

[9] Mannila, H., Toivonen, H.: Discovering generalized episodes using minimal occurrences. In: Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining (KDD), pp. 146–151 (1996)

[10] Meger, N., Rigotti, C.: Constraint-based mining of episode rules and optimal window sizes. LNCS, vol. 3302, pp. 313–324. Springer, Heidelberg (2004)

[11] Le Normand, N., Boissiere, J., Meger, N., Valet, L.: Supply chain management by means of flm-rules. In: Proc. of the 12th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 29–36 (2008)

[12] Schwabacher, M., Goebel, K.: A survey of artificial intelligence for prognostics. In: Working Notes of 2007 American Institute in Aeronautics and Astronautics Fall Symposium: AI for Prognostics, AIAA (2007)

Design Issues and Approach to Internet-Based Monitoring and Control Systems

Vu Van Tan and Myeong-Jae Yi

School of Computer Engineering and Information Technology
University of Ulsan, San-29, Moogu-2 dong, Namgu, Ulsan 680-749, South Korea
{vvtan,ymj}@mail.ulsan.ac.kr

Abstract. Manufacturers require an efficient reaction to critical events occurring at the process plants. Device data need to be integrated into business processes in a standardized and flexible way. Current research is limited due to a late indication of changes in the production environments, a delay of the implementation of changed production plans, and changes in technology. In this study, design issues arising from features of Internet-based monitoring and control applications have been addressed and an approach based on the Service-Oriented Architecture (SOA) and the OPC (Openness, Productivity, and Collaboration; formerly "OLE for Process Control") technology for such applications is proposed by integrating the enterprise systems with shop floor activities. Remote operators use the proposed system that collects all data from distributed control systems (DCSs) or programmable logic controllers (PLCs) to control complex manufacturing processes. Complex processes, i.e., process control and monitoring functions, will be spanned from devices on the shop floors to enterprise systems. Security of remote invocations in the heterogeneous environments is also ensured. The simulation results indicate the proposed system has good performance and is acceptable to Internet-based monitoring and control applications.

1 Introduction

The most successful research has been demonstrated that the Internet is a powerful tool for distributed collaborative works [11,3,14,8]. The emerging Internet technologies offer unprecedented interconnection capability and ways of distributing collaborative works, and have a great potential to bring the advantages of these ways of working to high-level control of process plants. These advantages include: (i) enabling remote monitoring and adjustment of devices on the shop floors or plants, (ii) enabling collaboration between plant managers situated in geographically diverse locations, and (iii) allowing the business to relocate the physical location of staff easily in response to business needs.

A monitoring and control system plays a central role to the classical automation field. The main functions of such a system are to gather signals produced production facilities and programmable logic controllers (PLCs), to combine them to control relevant contexts, to visualize them, and to provide functionalities to operate them. Existing manufacturing execution systems (MESs) and

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 478–488, 2010.

enterprise resource planning (ERP) systems are loosely connected via inflexible proprietary hierarchical systems. In general, access to specific device functionality is often restricted and may not be tailored to the specific requirements of the enterprise service. Integration for monitoring and control systems with other MES systems is therefore needed.

Remote operators in a distributed manner demand better and faster ways to achieve data and to react to plant functionalities anywhere and at anytime due to Internet environments. As the next generation technology for control systems, the concept of Internet-based monitoring and control has been proposed and introduced in recent years. Most research focused on Internet-based process control has only resulted in small-scale capability [1,3,5]. The OPC Foundation now is working on supporting XML and web services as well as in the OPC Unified Architecture (UA) specifications [12], so that Internet-based monitoring and control using XML and web services will become a reality.

Little research has been done on developing systematic design methods or guidelines for the design of Internet-based monitoring and process control systems. Unfortunately, the design methodologies used for the computer-based monitoring and control system, as they do not consider the Internet environment issues such as time delays caused by the Internet traffic, concurrent user access, web services, and security issue. For example, an Internet-based monitoring and control system has uncertainty about who the users are, how many users there will be, etc. It also has a variable working load. Several implementations for Internet-based monitoring and control systems discuss the limitation caused by the Internet environments [14]. Internet time delay and collaboration of multiple users are actually essential issues that must be addressed in the design of Internet-based monitoring and control systems.

This work attempts to provide a new approach based on the Service-Oriented Architecture (SOA) and the new OPC UA technology as a solution to integration of the enterprise systems and the shop floors for Internet-based monitoring and control. The necessary mechanisms are provided for an easy and quick design of the common process monitoring and control systems. Security of remote invocations in the heterogeneous environments is ensured.

This paper is organized as follows: The next section reviews related works. Section 3 deals with the role of an Internet-based monitoring and process control system and its design issues. An approach to integrate the enterprise systems and the process plants is presented in Section 4. The security of remote invocations is ensured in Section 5. Section 6 provides the performance evaluation of a reference implementation. Section 7 concludes some remarks and future works.

2 Related Work

Integration of functionality and information from field devices at the shop floors into the Internet of Services (IoS) is required in automation industry [3,1]. However, the exchange of data between production control level and the enterprise application level is often done either manually or semi-automatically. Given that the development of monitoring and control systems cannot be regarded as standalone systems, it has to take into account that there must be many interfaces to related IT-systems. An integration architecture for these systems with other MES systems and enterprise systems is needed.

A web service based shop floor integration infrastructure, called SOCRADES [3], is an architecture for next-generation industrial automation systems. By integrating future manufacturing services with services from all corporate functions via SOA, more flexible processes with high information visibility will emerge. This architecture is composed of web services hosted on smart devices within the manufacturing in terms of Internet of Things (IoT).

Due to the integration of web services on the shop floor, devices have possibilities of interacting seamlessly with the back-end systems [3]. For example, current products from Siemens Automation use the SOAP protocol for accessing tag based data from devices as well as display panels to PCs. However, they neither provide methods for discovering other web service-enabled devices nor ways for maintaining a catalogue of discovered devices.

The potential of SOAs for information and control flows in the shop floor domain, integrating applications as well as human workers as loosely coupled service providers was investigated [15,1]. Changes due to the current development through SOA are moving towards a promising approach to integrate shop floor devices and ERP systems more strongly [15]. Although several European projects like SIRENA [1] or SODA [11] demonstrated the feasibility and the benefit of embedding web services in devices used for production, these approaches do not offer an infrastructure or a framework for device supervision or device life cycle.

A MES middleware that makes efficient usage of shop floor data and generates meaningful key performance indicators is the SAP Manufacturing Integration and Intelligence (SAP MII) [10]. It uses a web server to extract data from multiple sources and stores them to database for history analytical purpose. In contrast to most other MES systems, the SAP MII can offer its functionalities over web services and provides connectors for legacy shop floor devices to integrate them into the SOA landscape. Its drawback is that every device has to communicate to the system using a driver that is tailored to the database connectivity.

A joined initiative for manufacturing domain object and control flow standardizations is ANSI/ISA-95 [9]. Although MES systems have been developed as gateways between the enterprise applications and the shop floors, they have to be tailored to the individual group of devices and protocols that exist on the shop floors. The OPC UA technology [12] is the next generation technology for secure, reliable, and inter-operable transport of raw data and preprocessed information from the process plants into production planning or the ERP system. It is based on XML, web services, and SOA to share information in more complex data structure formats with enterprise level MES and ERP systems in a way that they can understand. It embodies all the functionalities of the existing OPC standards and expands on top of them.

3 Monitoring and Control System and Its Design Issues

3.1 Architectural Overview and Infrastructure

Computer-based monitoring and control have been widely used in process and factory automation. Applications often come from standalone computer-based monitoring and control to local computer network-based control such as a DCS. A typical computer-based monitoring and control system is hierarchical. In general, it can be divided into the following levels: (i) plant-wide optimization level, (ii) supervisory level, (iii) regulatory level, and (iv) process and protection level [14]. The global database and the plant data processing are located at the top level where considerable computing power exists. Process database and supervisory control located at the second level where advanced control functions are implemented.

The purpose of establishing Internet-based monitoring and control systems is to enhance rather than replace computer-based monitoring and control systems. It can be achieved by adding an extra Internet level in the hierarchy. This extra level should be placed in the existing control system hierarchy according to the monitoring and control requirements. It might be linked with these existing systems via plant-wide optimization level or supervisory level or regulatory level.

3.2 Delays in Internet-Based Monitoring and Control Systems

Data exchanged between devices on the process plants and Internet-based clients allow that the clients not only monitor behaviors of the process plants, but also respond immediately to changes in quality. One of the difficulties in Internet-based monitoring and control is the dynamic delay caused by the Internet traffic.

Time delays according to interactions of the Internet-based monitoring and control system can be modeled in Fig. 1. The total time of performing a control action per cycle (t) is shown as follows:

$$t = t_1 + t_2 + t_3 + t_4, \tag{1}$$

where t_1 is the time delay in making control decision by a remote operator, t_2 is the time delay in transmitting a control command from the remote operator to the local system (sever), t_3 is the execution time of the local system to perform the control action, and t_4 is the time delay in transmitting the information from the local system to the remote operator.

The Internet time delay increases with distances, for example t_2 and t_4, but this delay also depends on the number of nodes traversed. It strongly depends on the Internet load [14]. In summary, the Internet time delay is characterized by the processing speed of nodes, the load of nodes, the connection bandwidth, the amount of data (the size of messages), and the transmission speed. The Internet time delay $T_d(k)$, i.e., t_2 and t_4, at instant k can be described as follows [8]:

$$T_d(k) = \sum_{i=0}^{n}(\frac{l_i}{C} + t_i^R + t_i^L(k) + \frac{M}{b_i}) = \sum_{i=0}^{n}(\frac{l_i}{C} + t_i^R + \frac{M}{b_i}) + \sum_{i=0}^{n} t_i^L(k) \tag{2}$$

The equation (2) can be rewritten as follows:

$$T_d(k) = d_N + d_L(k), \tag{3}$$

where l_i is the ith length of link, C is the speed of light, t_i^R is the routing speed of the ith node, $t_i^L(k)$ is the delay caused by the ith node's load, M is the amount of data, and b_i is the bandwidth of the ith link. d_N is independent of time and $d_L(k)$ is a time-dependent term.

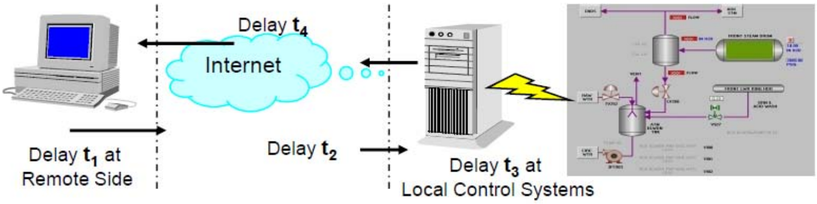

Delay t_4
Internet
Delay t_1 at Remote Side
Delay t_2
Delay t_3 at Local Control Systems

Fig. 1. The time delay in an Internet-based monitoring and control system

In this study, the required size of a message exchanged between the client and the server is purposely reduced as well. The header of the messages is still XML in order to pass firewall. The body of the messages is encoded in binary data to reduce the size of the messages.

3.3 User Access

When developing an Internet-based monitoring and control system, the features of this system are multiple users and the uncertainty about who the users are, how many users there are, and where they are. In the Internet-based monitoring and control systems, the operators cannot see each other. Several mechanisms can solve control conflict problems between multiple users and coordinate their operations. Three steps dealing with multi-user access can be listed as follows:

(1) Users with different priorities should be used. The user with a high priority can immediately overwrite the commands issued by users with lower priorities. In general, an easy way to identify a user is the user account when the user logs in the system.
(2) The system will be blocked for a certain period of time and might reject all further commands from users with equal or lower priorities if a new command is accepted.
(3) Only allowing a single user to operate the system in the tuning operation. This operation includes start-up, shut-down, and emergency handling. These command services will be held for only single user like a senior control engineer until this user logs out. Other users only can monitor operations.

3.4 Security Issue

It is difficult to secure the Internet-based monitoring and control systems because there is a possibility that the attackers over the Internet trying to cause failures through open network architectures. For simplicity, it can be assumed that the secure accesses to the local systems like DCSs, PLCs, etc., can be guaranteed by existing security methods. Thus, only the security of remote invocations caused by the remote users at the Internet is considered.

Security of remote invocations includes the authentication of clients, the encryption of massages, and the access control. In addition, possible scenarios may include a heavy Internet traffic time delay, more than one user trying to send a command or a method at the same time, or the method is not suitable for the current situation.

3.5 Integration Issue

The development of Internet-based monitoring and control systems cannot be regarded as stand-alone systems, it has to take into account that there must be many interfaces to related IT-systems. The use of wrappers is possible and reliable to wrap existing systems.

For integration of devices into sever side, some research has been focused on this field to propose flexible approaches to device integration by using Electronic Device Description Language (EDDL) [4] and Field Device Tool (FDT)/Device Type Manager (DTM) [6], especially for the new OPC UA technology [7,13]. In this study, the device type definitions are generated using the device information provided by the FDT/DTM and EDDL. They are composed of the parameters, default values, and the communication channels. Each device is represented by a device object in the server. The device object includes variables that hold the data items of the data model. It covers meta information such as unit, range, values, parameters, etc.

To apply the SOA paradigm in the device space using web services technology, the Devices Profile for Web Services (DPWS) was defined [15]. The SIRENA project was the first implementation of DPWS for simple, low-cost, embedded devices. Its results are used in the SOCRADES project as well as in the SODA project. Therefore, combined use of the DPWS and the OPC UA standard could provide a rich framework for device-level SOA. Integration of devices into the business IT-landscape through SOA is a promising approach to digitalize physical objects and to make them available to IT systems [1,3,15]. It is making chances to researchers for proposing their approaches to web services enabled device.

4 Approach to Internet Based Control and Monitoring

4.1 System Architecture Overview

Internet-based manufacturing provides new possibilities not only for static, data centric integration of the shop floors into an overall enterprise architecture, but

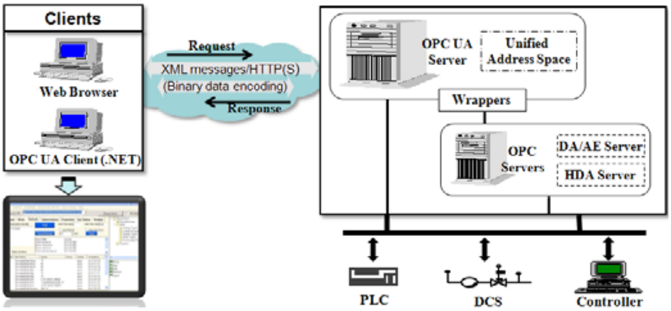


Fig. 2. Integration architecture for Internet based monitoring and control

also for full process control integration of field devices by means of SOA, XML, and web services. In addition to the introductions of web services at the level of the production devices or facility automation, there are still big problems in communication between device-level SOA and the one that is used in back-end systems. These problems can be overcome by using a middleware between the back-end applications and the services which are offered by devices, service mediators, and gateways. At least two different ways to couple networked embedded devices with enterprise services are: (i) a direct integration of device-level services in business processes, and (ii) exposure of device functionalities to the application layer via a middleware layer, e.g., OPC UA server.

The concrete architecture for integration of the enterprise systems and the process plants is proposed in Fig. 2. This architecture focuses on leveraging the benefits of existing technologies and takes them to next level of integration to ensure Internet-based monitoring and control of devices at the plants. All types of data such as *current data*, *historical data*, and *alarms and events* and commands defined as method services such as *start*, *stop*, etc., are covered.

4.2 Information Model and Access

Three types of data such as current data, historical data, and alarms and events with different semantics in the process industry and building automation are required. For example, the current value of a temperature sensor, an event resulting from a temperature threshold violation, and the historic mean temperature need to be collected for making a decision. Modeling of and access to data are important issues on designing and implementing an industrial application.

Modeling Data: A unified object model for three types of data is shown in Fig. 3. This model indicates: (i) current and historical data can be stored in variables; (ii) the commands to control devices at the process plants can be considered as method services for execution; (iii) the occurrence of an alarm or event from hardware devices can be considered as an event service. A common view of the field instruments up to the enterprise level should be provided in an efficient way. The unified object model defines objects in terms of variables,

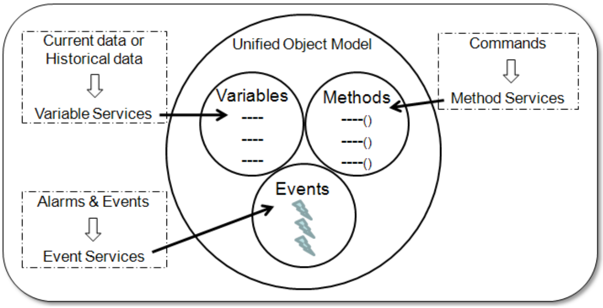


Fig. 3. A unified object model for current data, historical data, alarms and events, and commands used in Internet-based monitoring and control systems

events, and methods. It also allows relationships to other objects to be expressed. Objects and their components are represented in the address space as a set of nodes described by attributes and interconnected by references [12].

Accessing Data: The operator in the client side can monitor and acquire data or events from devices at the process plants through OPC servers, and also can write data to field devices if possible. The monitoring modes such as `Disabled`, `Sampling`, and `Reporting` can be set for each item. Operations `Read` and `HistoryRead` are used for achieving data from items and operations `Write` and `HistoryUpdate` are used for writing data to devices, etc.

4.3 Implementation Strategy

As the design principles of the human interface for highly automated processes, the human interface tasks are divided into two types of functions according to operational goals: (i) process control functions and (ii) process monitoring functions. The design objective for Internet-based monitoring and control systems is to enable the remote operators to appreciate more rapidly what is happening in the process plants and to provide a more simulating problem-solving environment outside the central control room. The software of the system can be divided into two parts like a server side and a client side. The client side interacts with users to perform their tasks while the server side is an OPC UA server that connects to devices for executing monitoring and control tasks.

Modeling the underlying process to make it available through services is required for engineers or developers who develop the Internet-based monitoring and control systems. To create a server address space based on the process with the OPC UA technology, the steps have to be done: (i) preparation of a logical model based on the real environment, (ii) preparation of a model according to the OPC UA notation, (iii) preparation of an XML file that describes the model, and (iv) development of an OPC UA server.

5 Security Solution

Security of remote invocations for Internet-based monitoring and control applications is very important. The required security objects for those applications such as availability, non-repudiation, accountability, authentication, confidentiality, integrity, and reliability should be ensured [12]. The XML security consisting of XML signature and XML encryption is the key concept to meet the required security objectives [2]. Thus, digital signatures are suitable for such applications.

The XML signature is a system to encode digital signatures in an XML document. It uses the same standards and technologies of cryptography as usual digital signatures. The basic of digital signatures is asymmetric cryptography. In contrast to symmetric cryptography, an entity in asymmetric cryptography creates a key pair in which one of them is the public key that can safely be published. By this way data can be encrypted for only the private key to decipher. The private key can be also used to generate digital signatures that are used to verify the public key. The steps in order to sign a message and to verify the signed message by communication partner using digital signatures are included:

(1) Calculating the hash code for data to be signed.
(2) Encrypting the hash code using private key.
(3) Attaching the hash code and the encrypted hash code into the message before transmitting this message.
(4) The receiver of the singed message will decrypt the encrypted hash code with the public key of the communication partner. If the decrypted hash code and the plain hash code of the message are identical then the communication partner is the expected one. Otherwise the message has to be discarded or rejected because of a potential attack.
(5) The receiver calculates the hash code of the same data and compares the two has codes. If the hash codes are identical the the integrity of the message has been guaranteed. Otherwise the data were modified that must be discarded.

6 Performance

In this study, several performance tests by verifying various items in the client side were performed. The setups were composed of the OPC UA server and client running on Windows XP Service Pack 2 with 2.66 Ghz Intel Pentium VI CPU and 1 Gb PC 3200 DDRAM. The time taken for operation `Read` to fetch a various number of items under various conditions was set out to measure. The comparison of the time taken for operation `Read` under the fixed conditions (binary encoding, signature and encryption, 256-bit Basic) with either using HTTP or TCP is shown in Fig. 4(a). Using binary data encoding for the messages exchanged between the client and the server will therefore improve the system performance much because of reducing the size of the messages.

The comparison of the time taken for operation `Read` under the fixed conditions (binary encoding, HTTP, and 256-bit Basic) with using signature and

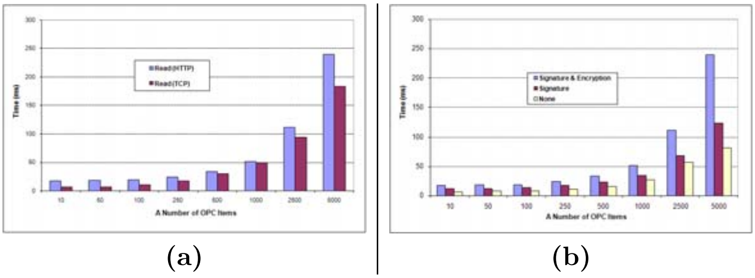


Fig. 4. Time taken for operation `Read`: (a) using either *HTTP* or *TCP* protocol; (b) different security modes – *Signature and Encryption*, *Signature*, and *None*

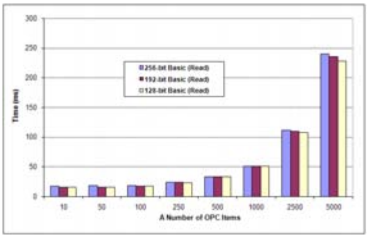


Fig. 5. The comparison of the time taken for operation `Read` under the fixed conditions: *binary encoding*, *signature and encryption*, and *HTTP*

encryption, signature, and none, respectively, is shown in Fig. 4(b). It indicates the overall performance of the proposed system when using both signature and encryption is approximately about two times slower than the case of not using both ones. It also indicates the overall performance of the proposed system when using only signature is approximately about three second times faster than the case of using both signature and encryption. The time taken for operation `Read` under the fixed conditions (binary encoding, signature and encryption, and HTTP) using different lengths of bit Basic for signature and/or encryption to a SOAP message can be compared in Fig. 5. That is, there is no much difference when using different lengths of bit Basic.

7 Concluding Remarks and Future Works

An extensible approach to the integration of enterprise systems and shop floors used for modern process and factory automation was suggested. The simulation results have indicated the performance of the proposed approach is acceptable to many Internet-based monitoring and control applications. The design issues for Internet-based monitoring and control applications were discussed. They ensure that the proposed approach allows a long-term development of modern process monitoring and control systems. All production processes will be exposed from the enterprise applications down to field devices on the plant floor or shop floors in a flexible and extensible way.

The future works are to use the proposed architecture for the implementation of a control and monitoring approach, and to investigate and deploy the proposed approach by real process monitoring and control applications.

Acknowledgements. This work was supported in part by the Korean Ministry of Knowledge Economy and Ulsan Metropolitan City through the Network-based Automation Research Center at University of Ulsan and by the Korea Research Foundation Grant funded by the Korean Government (KRF-2009-0076248).

References

1. Bohn, H., Bobek, A., Golatowski, F.: SIRENA – Service Infrastructure for Real-time Embedded Networked Devices: A Service Oriented Framework for Different Domains. In: Proceedings of the Int. Conf. on Systems and Int. Conf. on Mobile Comm. and Learning Tech., p. 43. IEEE CS Press, Los Alamitos (2006)
2. Braune, A., Hennig, S., Hegler, S.: Evaluation of OPC UA Secure Communication in Web browser Applications. In: Proceedings of the IEEE Int. Conf. on Industrial Informatics, pp. 1660–1665. IEEE Press, Los Alamitos (2008)
3. de Souza, L.M.S., Spiess, P., Guinard, D., Köhler, M., Karnouskos, S., Savio, D.: SOCRADES – A Web Service Based Shop Floor Integration Infrastructure. In: Floerkemeier, C., Langheinrich, M., Fleisch, E., Mattern, F., Sarma, S.E. (eds.) IOT 2008. LNCS, vol. 4952, pp. 50–67. Springer, Heidelberg (2008)
4. EDDL - Electronic Device Description Language, http://www.eddl.org/
5. Eppler, W., Beglarian, A., Chilingarian, S., Kelly, S., Hartmann, V., Gemmeke, H.: New Control System Aspects for Physical Experiments. IEEE Transactions on Nuclear Science 51(3), 482–488 (2004)
6. FDT-JIG Working Group: FDT Interface Specification, version 1.2.1. FDT Joint Interest Group (2005), http://www.fdtgroup.org/
7. Grossmann, D., Bender, K., Danzer, B.: OPC UA based Field Device Integration. In: Proceedings of the SICE Annual Conference, pp. 933–938 (2008)
8. Han, K.H., Kim, S., Kim, Y.J., Kim, J.H.: Internet Control Architecture for Internet-based Personal Robot. Autonomous Robots 10, 135–147 (2001)
9. Instruments, Systems and Automation Society, http://www.isa.org/
10. SAP xApp Manufacturing Integration and Intelligence, SAP document number 50 082 980 (2007)
11. SODA, http://www.soda-itea.org/Home/default.html (accessed June 2008)
12. The OPC Foundation (2008): The OPC Unified Architecture Specification: Parts 1-11. Version 1.xx (2008), http://www.opcfoundation.org/Downloads.aspx
13. Yamamoto, M., Sakamoto, H.: FDT/DTM Framework for Field Device Integration. In: Proceedings of the SICE Annual Conference, pp. 925–928 (2008)
14. Yang, S.H., Chen, X., Alty, J.L.: Design Issues and Implementation of Internet-based Process Control Systems. Control Engineering Practice 11, 709–720 (2003)
15. Zeeb, E., Bobek, A., Bohn, H., Golatowski, F.: Service-Oriented Architectures for Embedded Systems Using Devices Profile for Web Services. In: Proc. of the 21st Int. Conf. on Advanced Inf. Networking and App. Workshops, pp. 956–963 (2007)

Simul-EMI II: An Application to Simulate Electric and Magnetic Phenomena in PCB Designs

Juan-Jesús Luna-Rodríguez[1], Ricardo Martín-Díaz[1], Manuel Hernández-Igueño[1],
Marta Varo-Martínez[2], Vicente Barranco-López[3], Pilar Martínez-Jiménez[2],
and Antonio Moreno-Muñoz[1]

[1] University of Cordoba, Department of Computer Architecture, Electronic and Electronic Technology. 'Leonardo da Vinci' building (Campus Rabanales)
14071 Cordoba, Spain

[2] University of Cordoba, Department of Applied Physics. 'Einstein' building (Campus Rabanales)
14071 Cordoba, Spain

[3] University of Cordoba, Department of Electrical Engineering. 'Leonardo da Vinci' building (Campus Rabanales)
14071 Cordoba, Spain

el1luroj@uco.es, p42madir@uco.es, p42heigm@uco.es,
fa2vamam@uco.es, el1balov@uco.es, fa1majip@uco.es,
amoreno@uco.es

Abstract. In this work, an application to simulate electric and magnetic phenomena during design of printed circuit boards (PCB) is presented. The commercial schematic simulation software currently uses advanced models of components, but not of connections. However, with Simul-EMI II it is possible to get results very close to physical reality of the electronic circuits, in the same environment of schematic simulation. The parasitic effects considered are: connection resistance (traces, solder pads and vias), insulation resistance (prepreg, air, solder mask, etc.), self inductance and capacity (traces, solder pads and vias) and mutual inductances and capacities. For that, several applications of data mining, parametric model extraction and knowledge management have been developed in the MATLAB™ environment. Finally, the results of a PCB simulation with PSPICE™ before and after Simul-EMI II application, together with the electronics laboratory validation tests, are shown.

Keywords: EMI simulation, PCB simulation, EDA application, connection model, model generation, trace coupling, crosstalk, signal reflection.

1 Introduction

The Printed Circuit Boards (PCB) are the most important circuit technology in the electronic industry. The increase of connection density and signal frequency has been the trend in the last years [1]. In that sense, the separation and routing of

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 489–498, 2010.

traces[1] in connection layers have become one of the most critical design tasks. Currently, electronic design engineers need new software technologies (e.g. in simulation) to develop circuits in environments as Computer Aided Design (CAD), Computer Aided Engineering (CAE) or Electronic Design Automation (EDA). The main purpose of that is to reduce the cost and time of electronic circuit development, whereas the reliability and function complexity of the modern circuits are increased [2].

In practice, the probability of design errors and operation failures during the development of a circuit is high and that increases its cost and completion time. This is mainly due to the fact that modern electronic circuits are very complex hardware systems. The components are connected by means of hundreds or even thousands of traces, vias (plated holes that connect several layers) and pad solders, whose behaviour differ from ideal connections. [3]

The electrical simulation is usually done during the schematic circuit design by means of simulators such as PSPICE™ or Electronic Work Bench™ (EWB), in a Computer Aided Simulation (CAS) environment. These computer applications use advanced models of the components but not of the connections, which are considered as ideal connections. To understand the importance of this fact, Fig. 1 shows an example of the difference between an ideal connection (connecting two resistors) in a schematic circuit and a real connection in the PCB, where there are multiples electrical and magnetic phenomena. [4]

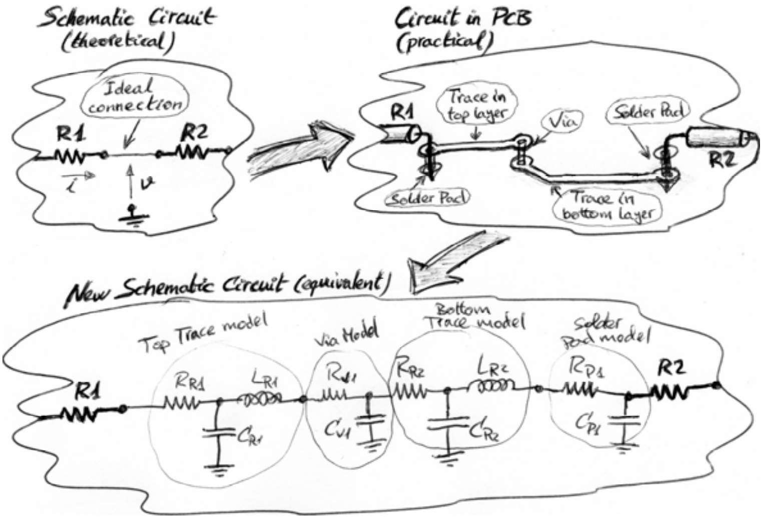


Fig. 1. Changes of circuit connection model after its implementation on a PCB

Currently there are plenty of commercial solutions to simulate the electromagnetic interference (EMI) phenomena of printed circuits, e.g. *SI Verify* by Zuken™ or *IE3D* by Zeland Software™, which are based in Finite Element Modeling/Method (FEM).

[1] Thin copper sheets, also called tracks, used to connect the pins of components.

But these are independent of the electric simulation environments, most popular among electronic designers. In addition, their acquisition and maintenance cost is very high and they are not easy to use. [5]

2 General Description of Simul-EMI II

The main aim of “Simul-EMI II” project is to provide an affordable alternative (cheap, easy and fast) to expensive commercial EMI simulation software, with results very close to the physical reality of the electronic circuits. For this purpose, our application includes most electric and magnetic interference phenomena (coupling, crosstalk, reflections, etc.) that are actually produced in a PCB. In this simulation environment the all near field EMIs can be simulated, which include the resistance effects (contact and isolation), inductance (self and mutual) and capacity (self and mutual) of traces, solder pads and vias [6]. The far field phenomena (electromagnetic radiation) are not supported in Simul-EMI II, for the time being.

The main premises for the Simul-EMI II development have been: working in the same electrical simulation environment as in the schematic design (PSPICE in this case) and using only the normally available information from the PCB design software. In Fig. 2, a scheme of the general steps of our simulation environment is shown, where the typical workflow can be easily understood.

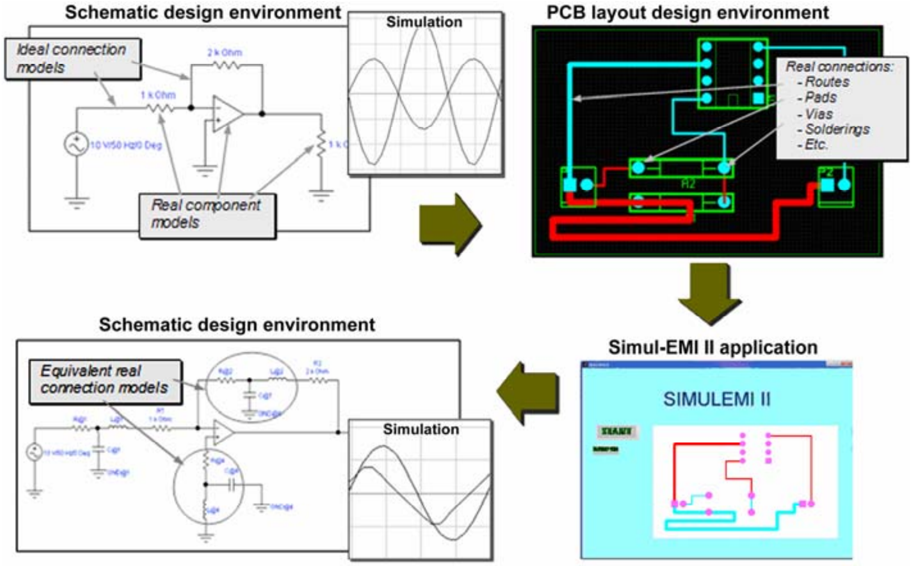


Fig. 2. Steps in the development process of an electronic circuit using Simul-EMI II

Briefly, Simul-EMI II can read the schematic design from EDA software (with ideal connections) and then the physical design of the corresponding PCB is interpreted. Finally, a new schematic circuit (compatible with PSPICE™) is redesigned automatically, which includes electromagnetic models of the actual connections. In order to achieve this, a MATLAB™ program is executed with help from the EXCEL™ spreadsheet. This program performs the following tasks:

1. Data mining:
   - Net-list and component-list reading.
   - "Gerber" files (geometric data) interpretation.
   - Technological information acquisition.
2. Extraction of models:
   - Conversion of geometric and technological data to parasite parameters.
   - Management of data base of EMI knowledge.
3. Redesign of schematic circuit:
   - Equivalent connection models builder.
   - Net-list rebuilder.

Consequently, Simul-EMI II can be considered as an "application to design" (in the EDA context) related to Data Mining in field of the Artificial Intelligence (AI).

3 EMI Phenomena Modelling

In the computer application presented in this paper, a knowledge database on near field EMIs has been included. The parasitic effects considered in this database are: connection resistance, insulation resistance, self inductance and capacity and mutual inductances and capacities [3]. These parasitic elements are the ones that cause the most disturbing phenomena in the electronic circuits, such as crosstalk and reflections, especially when high frequency signals are present.

The calculation of connection return resistance may be more complicated when the same path return is used to finish loops of multiple signals. Considering a distribution multiple traces of the "microstrip line" type sharing the return signal through the ground plane, a resistance matrix, as shown in (1), can be defined to establish the interaction between conductors and their return paths. [1]

$$\mathbf{R} = \begin{pmatrix} R_{11} & R_{12} & \cdots & R_{1j} \\ R_{21} & R_{22} & \cdots & R_{2j} \\ \cdots & \cdots & \cdots & \cdots \\ R_{i1} & R_{i2} & \cdots & R_{ij} \end{pmatrix} . \tag{1}$$

The values showed on the main diagonal (R_{11}, R_{22}, …, R_{ij}) represent the conductor resistances in free loop, including the resistance of its return path when the others are conducting zero current. The terms outside the main diagonal represent the mutual resistance between each conductor. Finally, note that the dielectric losses of insulations, the side etching factor of the conductors and its temperature dependence are also included in the matrix formulation of resistive parasitic effects. [3]

Parasitic capacitances are present between any two conductive surfaces at different voltage. For traces of a PCB, this means that there will be capacities between each other and also between them and the reference plane [1]. The relations of self and mutual capacity on PCB traces can be defined by means of a capacity matrix, similar to the previous resistance matrix.

$$\mathbf{C} = \begin{pmatrix} +C_{11} & -C_{12} & \cdots & -C_{1j} \\ -C_{21} & +C_{22} & \cdots & -C_{2j} \\ \cdots & \cdots & \cdots & \cdots \\ -C_{i1} & -C_{i2} & \cdots & +C_{ij} \end{pmatrix}. \tag{2}$$

The mutual capacitances are out of the diagonal in (2) and represent the charges between a set of conductors. In the circuit, these charges are considered capacitive couplings and must appear as negative in the matrix. The self capacitance C_{11}, C_{22}, …, C_{ij} represent the capacitive coupling between conductors with the GND conductor. [3]

As it was done with the parasitic resistances and parasitic capacitances, an inductance matrix including self and mutual inductances related to each of the PCB traces can be defined. [1]

$$\mathbf{L} = \begin{pmatrix} L_{11} & L_{12} & \cdots & L_{1j} \\ L_{21} & L_{22} & \cdots & L_{2j} \\ \cdots & \cdots & \cdots & \cdots \\ L_{i1} & L_{i2} & \cdots & L_{ij} \end{pmatrix}. \tag{3}$$

The inductance matrix (3) shows the inductance of each track with his return in the main diagonal, when currents in the rest of the conductors are zero. The non-diagonal terms represent the coupling inductance that appears between traces. [3]

In Fig. 3 the equivalent model of a continuous transmission line, applicable to the traces on a PCB, is shown. This model divides the transmission line (the trace) in a set of standard length segments.

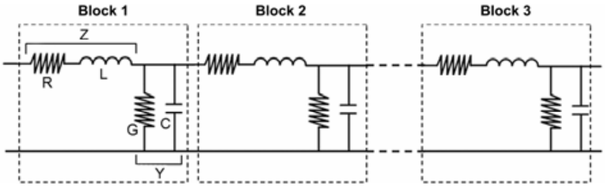


Fig. 3. Characteristic impedance of transmission lines from R, L, G and C parameters

The series impedance consists of a resistance R coupled with an inductance L. The shunt admittance is a conductance G in parallel with a capacitance C. The values of R, L, C and G represent the cumulative amount of resistance, inductance, capacity and conductance per unit length, respectively, in the transmission line. [1]

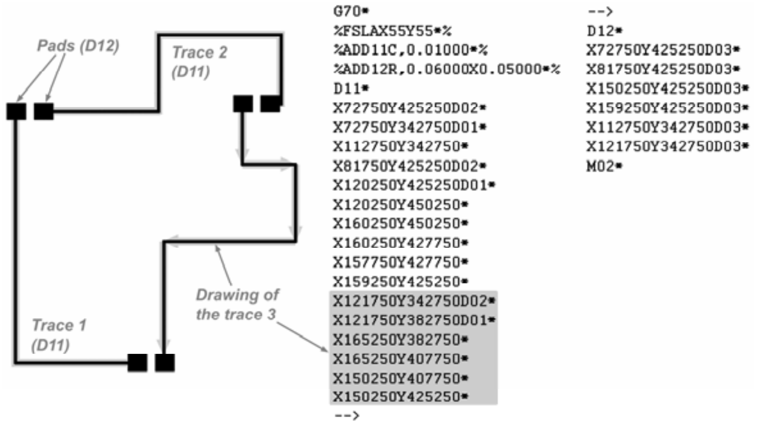
The advantage of the connection equivalent model based on transmission lines with infinite blocks in series (Fig. 3) is that adding a new block does not change the input impedance of the whole structure. [7]

4 Data Mining and Automatic Generation of Parametric Models

To calculate the actual connection models, the geometric data of PCB design are required, as well as some technological information (e.g. layer stack, materials, etc.). Geometrical information of the traces, solder pads and vias are contained in the design software database, but these are not usually accessible from another computer application. [8]

However, this information can be extracted by applying data mining to GERBER[2] files (all EDA software can generate them), which are typically used to send drawings of the different layers to the manufacturer [9]. Then, as the GERBER format contains only graphic information, it has been necessary to implement an "interpreter" to transform the drawing into dimensions. [10]

The numerical extraction phase of coordinates from a GERBER file has been automated with MATLAB™, working with Microsoft EXCEL™ [11], since it is possible to separate and classify the numerical data and ASCII characters in the spreadsheet in a simple and effective way.



```
G70*
%FSLAX55Y55*%
%ADD11C,0.01000*%
%ADD12R,0.06000X0.05000*%
D11*
X72750Y425250D02*
X72750Y342750D01*
X112750Y342750*
X81750Y425250D02*
X120250Y425250D01*
X120250Y450250*
X160250Y450250*
X160250Y427750*
X157750Y427750*
X159250Y425250*
X121750Y342750D02*
X121750Y382750D01*
X165250Y382750*
X165250Y407750*
X150250Y407750*
X150250Y425250*
-->
```

```
-->
D12*
X72750Y425250D03*
X81750Y425250D03*
X150250Y425250D03*
X159250Y425250D03*
X112750Y342750D03*
X121750Y342750D03*
M02*
```

Fig. 4. A simple layout of top connections and its corresponding extended GERBER file

In order to simplify and systematize the final parameterization algorithm, each trace is divided into segments. For example, track 3 of the layout shown in Fig. 4 will be processed as 5 traces, corresponding to its 5 segments.

In order to calculate the "length" of each trace, it has only been necessary to draw each segment step by step, while a program counter updates that parameter. The inference of the parameter "separation" has been somewhat more complicated, because our MATLAB™ program must check the parallelism of each trace segment with others traces, and how much of its length they are facing. [12]

A study about the influence of distance on the parasitic capacitance between two traces is shown in Fig. 5. This study has allowed to identify a threshold value (~10mm) from which the coupling effects between two traces are discarded.

[2] The extended GERBER is the currently used format for obtaining the manufacturing artworks by photoplotter, and it is specified in the RS-274X standard by Barco Graphics.

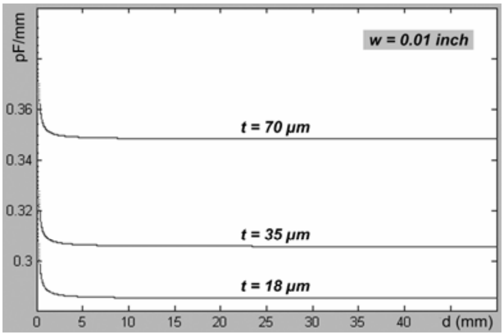


Fig. 5. Study of mutual capacity depending on the distance between traces

The calculated geometrical parameters, together with some data entered via user interface, are now used for the model extraction. For this, a database of knowledge about the parasitic effects described in section 3 (EMI Phenomena Modelling) has been implemented with the available formulas in the current literature. [13] [14] [15]

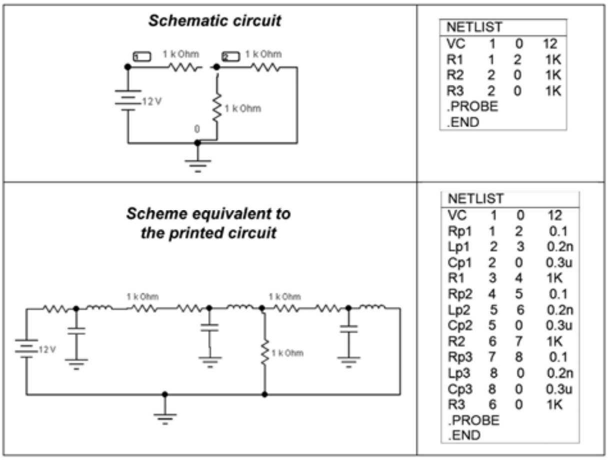
5 Reconstruction of the PCB Equivalent Schematic Circuit

The description of schematic circuits in file form has been traditionally done by the "net-list" format. This file contains all the information related to the circuit components and the connections between them. A file containing this information can be obtained directly from the CADSTAR™ through a tool called "report generator", which is only a processor of scripts with access to the PCB design software database.

In addition to the data of components and connections (obtained from the net-list file) it is necessary to relate each net with the component pins and their exact positions on the PCB. Below, the simple program that extracts this information from the database of CADSTAR™ is showed.

```
Text "Component   Value      Position X   Position Y"
New Line
For Each Component (Component, Testpoint)
  If Component.Comp Side = All on Top Side
    For Each Pin
      Component.Name
      Component.Part Name
      Text "     "
      Pin.Position X
      Text ""
      Pin.Position Y
    Next
  End If
Next
```

To simulate the printed circuit in the same environment of the schematic circuit (PSPICE™), including models of the actual connections, the original net-list is modified by inserting new nodes [16]. Fig. 6 shows an example of the reconstruction of the original net-list to include EMI phenomena of the PCB.



```
NETLIST
VC    1   0   12
R1    1   2   1K
R2    2   0   1K
R3    2   0   1K
.PROBE
.END
```

```
NETLIST
VC    1   0   12
Rp1   1   2   0.1
Lp1   2   3   0.2n
Cp1   2   0   0.3u
R1    3   4   1K
Rp2   4   5   0.1
Lp2   5   6   0.2n
Cp2   5   0   0.3u
R2    6   7   1K
Rp3   7   8   0.1
Lp3   8   0   0.2n
Cp3   8   0   0.3u
R3    6   0   1K
.PROBE
.END
```

Fig. 6. A net-list example of a simple circuit and its reconstruction using SIMULEMI II

The final result of this process, automated by using programming functions in MATLAB™ code, is a net-list file compatible with PSPICE™ to simulate an actual PCB electronic circuit like a schematic circuit [12] [16].

6 Tests and Validation

To validate the Simul-EMI II application, a simple passive voltage divisor circuit has been designed. As shown in Fig. 7, the traces routing (layout) on the PCB is pretty bad (it has been done on purpose) in order to highlight the EMI phenomena.

The practical net-list (with actual connections) of this circuit after the application of Simul-EMI II includes more than 70 lines. Each of these lines corresponds to a parasitic effect, which is not considered in the theoretical net-list based on ideal connections.

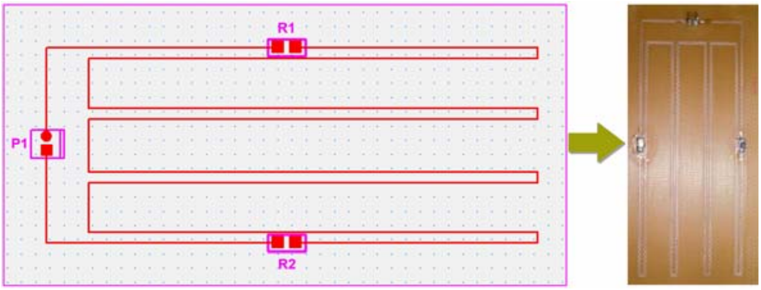


Fig. 7. Passive voltage divisor in a printed circuit board designed (badly) in CADSTAR

Subsequently, this circuit has been manufactured and there have been several tests in the laboratory, using a wave generator Tektronix AFG 3022B Dual Channel – Arbitrary / Function Generator (250MS/s – 25MHz) and an oscilloscope Tektronix TDS2002B Two Channel Digital Storage Oscilloscope (1GS/s – 60MHz). In Fig. 8, the results of simulation with PSPICE™ before and after Simul-EMI II application, together with the electronics laboratory tests, are shown.

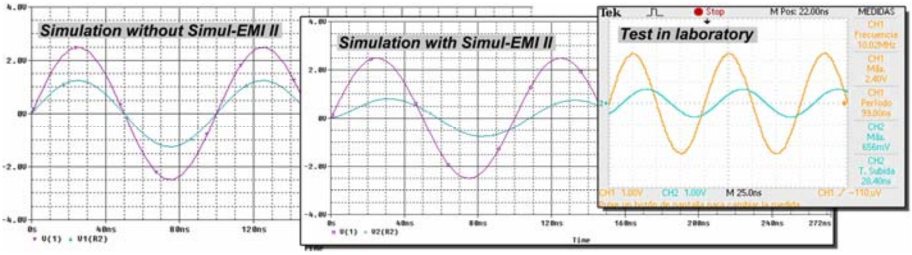


Fig. 8. Simulation with sinusoidal excitation signal of 10MHz frequency and 5Vpp range

As shown, the simulation results with Simul-EMI II are very similar to the tests conducted in the laboratory. However this does not happen in the simulations without Simul-EMI II. Laboratory tests with frequencies below 1 MHz showed no parasitic effects, and consequently they have not been included in this paper.

7 Conclusions

The schematic simulation software currently uses advanced component models, but connection models are simple (ideal). However, with the application presented in this work is possible to get results very close to the physical reality of the electronic circuits, in the same environment of schematic simulation. Simul-EMI II can read the schematic design from EDA software (with ideal connections) and then the physical design of the corresponding PCB is interpreted. Finally, a new schematic circuit (compatible with PSPICE™) is redesigned automatically, which includes electromagnetic models of the actual connections.

Given the results of the validation tests, we can conclude that the parasitic effects of the connections are irrelevant below 1MHz. However, above 3MHz these effects can modify substantially the expected (ideal) behaviour of the circuit.

Moreover, comparative analysis of simulations carried out with Simul-EMI II and the measurements performed in the laboratory confirm the validity of this computer application as a useful tool for electronic design.

References

1. Jonson, H., Graham, M.: High Speed Signal Propagation - Advanced Black Magic. Prentice Hall, Upper Saddle River (2003)
2. Abhari, R., Eleftheriades, G.V., Deventer-Perkins, E.: Physics-Based CAD Models for the Analysis of Vias in Parallel-Plate Environments. IEEE Transactions on Microwave Theory and Techniques 49(10), 1697–1707 (2001)

3. Thierauf, S.C.: High-Speed Circuit Board Signal Integrity. Artech House, Norwood (2004)
4. Luna-Rodríguez, J.J.: Diseño de Circuitos Impresos: un Manual Teórico-Práctico con CadStar. Universidad de Córdoba, Córdoba (2008)
5. Klee, H.: Simulation of Dynamic Systems with MATLAB and Simulink. CRC Press, Boca Raton (2007)
6. Braithwaite, N., Weaver, G.: Electronic Materials inside Electronic Devices. The Open University, London (1990)
7. Scarlatti, A., Holloway, C.L.: An equivalent transmission-line model containing dispersion for high-speed digital lines with an FDTD implementation. IEEE Transactions on Electromagnetic Compatibility 43(4), 504–514 (2001)
8. Kusiak, A., Kurasek, C.: Data Mining of Printed-Circuit Board Defects. IEEE Transactions on Robotics and Automation 17(2) (2001)
9. Choudhary, A.K., Harding, J.A., Tiwari, M.K.: Data mining in manufacturing: a review based on the kind of knowledge. J. Intell. Manuf. 20, 501–521 (2009)
10. Barco Graphis, N.V.: Gerber RS-274X Format. User's Guide. Gent. (2001)
11. Artwork Conversion Software, Inc., `http://www.artwork.com/gerber/274x/rs274x.htm`
12. Downey, A.B.: Physical Modeling in MATLAB. Green Tea Press, Needham (2008)
13. Abhari, R., Eleftheriades, G.V., Deventer-Perkins, E.: Analysis of Differential Vias in a Multilayer Parallel Plate Environment Using a Physics-Based CAD Model. In: IEEE International Microwave Symposium, Phoenix, pp. THIF-09-4 (2001)
14. Chen, H., Li, Q., Tsang, L., Huang, C.C., Jandhyala, V.: Analysis of a Large Number of Vias and Differential Signaling in Multilayered Structures. IEEE Transactions on Microwave Theory and Techniques 51(3), 818–829 (2003)
15. Kouzaev, G.A., Nikolova, N.K., Deen, M.J.: Circular-Pad Via Model Based on Cavity Field Analysis. IEEE Microwave and Wireless Components Letters 13(11), 481–483 (2003)
16. Quintáns, C.: Simulación de Circuitos Electrónicos con OrCAD® 16 Demo. Marcombo, Barcelona (2008)

Networked Control System Design on the Basis of Two Disk Mixed Sensitivity Specifications

Guido Izuta

Yonezawa Women's College,
Department of Social Information,
992-0025 Yonezawa, Yamagata, Japan
izuta@yone.ac.jp

Abstract. In this investigation we are concerned with the controller design for a control system over the network on the grounds of the two disk mixed sensitivity problem. Thus, it suggests a design procedure to establish a filter type controller, which can be tuned graphically to satisfy the stability and the performance specifications. For this purpose, a mix of frequency domain and state space model approaches are adopted, in which the state feedback decoupling technique plays a key role.

1 Introduction

Advances in network communication technologies have been broadening and increasing the range of possibilities for advanced industrial production systems implementations, amongst them are the automatic control of industrial processes over the network, tele-operation and tele-robotics, to name just a few [1,7]. These industrial problems are in the scope of an interdisciplinary research area that spans control systems engineering, computer science and digital signal processing among others (see for example [11] and references therein). From the control theory perspectives, due to the challenging and involving mathematical technicalities inherent in the systems controlled over the network models, as the modeling and handling of data transmission delays, the controller design problem is one of the main research issues to be settled before one can move on to the on-site practical details [6].

Motivated by these basic control theoretical needs, we focus in this paper on the controller design problem for the networked control system model with transmission delays between the plant and controller and vice verse. Essentially, the design objective is the two disk mixed sensitivity specifications, which are basic feedback system design requirements expressed in an H_∞ control theory jargon [5]. Thus, since the control model uncertainties and the influence of the disturbances on the system are taken into account, the design procedure is concerned with the robust stability as well as the robust performance of the system. The levels of stability and performance can be specified by the designer/practitioner within the limits constrained by the two disk mixed sensitivity specifications. To achieve the aims, unlike the most papers on this topic (see for example [2]

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 499–508, 2010.

and references therein), we adopt here the classical frequency domain approach and the state feedback decoupling [8], which allow us to translate the controller design problem into the one of tuning filters parameters by means of graphical plots. Nevertheless this technique is quite related to the author's previous papers on control design of time delayed systems [3,4], in this paper we go further in the sense that here the control systems have delays in both inputs or outputs sides.

Finally, the paper is organized as follows. Section 2 enunciates the problem, section 3 provides the formalization of the system and the problem. In section 4 we provide the results of the paper, and in section 5 presents a numerical example. Finally, a few concluding remarks are written down 6.

2 Problem Formulation

In this work, we are concerned with the networked control system shown in figure 1, in which $u(t)'s$ are the control inputs, $y(t)'s$ are the outputs, $d(t)'s$ are the exogenous inputs (disturbances), and $r(t)'s$ are the reference signals. Furthermore, the following assumptions and conditions are made:

1. the disturbances act on the inputs of the system,
2. the control system is yielded by the system itself $G_p(s)$ and additive model uncertainties $I + \Gamma(s)W_T(s)$ due to the input side disturbances,
3. the controller $P(s)$ is basically an output feedback controller,
4. the controller $Q(s)$ is a free parameter type controller depending on the inputs and errors of the control system,
5. the control inputs and errors are transmitted through the network in the same packet, and such that the delay from the control system to the controller and from the controller to control is δ.

Under these settings, the problem to be handled here reads

Problem 1. Design a controller satisfying the two disk mixed sensitivity conditions [5]

$$\begin{array}{c} \| W_S(j\omega)S(j\omega) \|_\infty < 1 \\ \| W_T(j\omega)T(j\omega) \|_\infty < 1 \end{array} \tag{1}$$

where $S(j\omega)$ and $T(j\omega)$ are the sensitivity and complementary sensitivity functions, respectively; and $W_S(j\omega)$ and $W_T(j\omega)$ are the related weighting functions.

3 Preliminaries

In this section, we give the mathematical formulation of the system and the problem to be solved. To begin with, the control system is firstly stated as

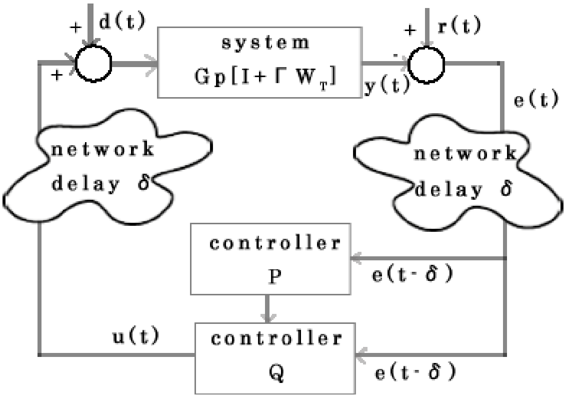


Fig. 1. Networked control system

Definition 1. *The state space description of the control system $G_P(s)$ in figure 1 is given by*

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t-\delta) \\ y(t) &= Cx(t) \end{aligned} \tag{2}$$

where $x(t) \in \Re^{n\times 1}$ are the states, $u(t) \in \Re^{m\times 1}$ mean the inputs, $y(t) \in \Re^{m\times 1}$ are the outputs; A, B and C are real valued matrices of appropriate size; and (A, B), (C, A) are controllable and observable, respectively. Thus,

$$G_P(s) = G(s)e^{-s\delta}\left[I + \Gamma(s)W_T(s)\right], \tag{3}$$

in which $G(s) = C(sI-A)^{-1}B$ is the stable nominal model with no zeroes on the complex axis and such that the state feedback decoupling operation is applicable on $G(s)^T$, and $\parallel \Gamma(j\omega)W_T(j\omega) \parallel_\infty$ is the model uncertainties such that

$$\begin{aligned} &\parallel \Gamma(j\omega) \parallel_\infty < 1 \\ &\mid W_T(j\omega) \mid \begin{cases} < 1, 0 < \omega < \omega_T \\ = 1, \omega = \omega_T \\ > 1, \omega_T < \omega \end{cases} \end{aligned} \tag{4}$$

Remark 1. The state feedback decoupling assumption means that there exist $\mu_i' s$ $(\mu > 0)$ such that

$$\begin{aligned} &CA^{k-1}b_i = 0, \quad k = 1, \cdots, \mu_i - 2 \\ &CA^{\mu_i - 1}b_i \neq 0, \end{aligned} \tag{5}$$

where $b_i' s$ are the column vectors of matrix B. Furthermore,

$$\begin{aligned} &\Psi = [CA^{\mu_1 - 1}b_1 \ \cdots \ CA^{\mu_m - 1}b_m] \\ &\text{rank}\, \Psi = \text{m} \end{aligned} \tag{6}$$

holds.

Now, the controller in terms of the data transmission delays with a real valued matrix P to be designed follows as

Definition 2. *The controller $P(s)$ is a dynamic output feedback controller with the an internal memory scheme such that the past values of $u(t)$ can be used as given by the model*

$$\begin{aligned} \dot{p}(t) &= Ap(t) + Bu(t-\delta) - P\{e(t-\delta) + Cp(t)\} \\ e(t) &= r(t) - y(t), \end{aligned} \tag{7}$$

in which $p(t) \in \Re^{n\times 1}$.

Moreover, the controller that works as a free parameter and produces the control system inputs is as provided in the next paragraph.

Definition 3. *The controller is composed as $Q(s) = Q_S(s)Q_c$, in which Q_c is a real valued matrix, and $Q_S(s)$ is a dynamic system modeled by the following system of differential equations in the time domain:*

$$\begin{aligned} \dot{q}(t) &= A_q q(t) + B_q\{e(t-\delta) + Cp(t)\} \\ u(t) &= C_q q(t) + D_q\{e(t-\delta) + Cp(t)\}, \end{aligned} \tag{8}$$

where $A_q \in \Re^{q_A\times q_A}$, $B_q \in \Re^{q_A\times n}$, $C_q \in \Re^{m\times q_o}$, and $D_q \in \Re^{m\times m}$. Thus, in the Laplace transformed domain, $Q_S(s)$ translates into $Q_S(s) = C_q(sI - A_q)^{-1}B_q + D_q$.

Taken these into account, we reduce the networked control system into a feedback control system composed by the control $G_p(s)$ and the controller $G_C(s)$ as given in figure 2.

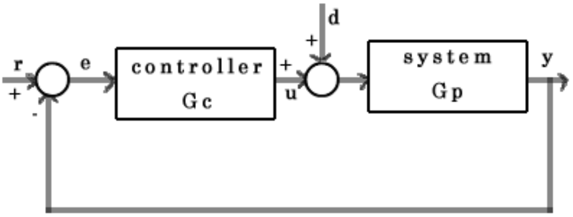


Fig. 2. Equivalent feedback control system

On stating the system in figure 2 in a more formal terms, we have

Definition 4. *The controller $G_C(s)$ to be designed is the transfer function from $e(s)$ to $u(s)$, which is described as*

$$\begin{aligned} G_C(s) = {} & [I - Q(s)C(SI - A + PC)^{-1}Be^{-s\delta}]^{-1} \\ & \times Q(s)[I - C(sI - A + PC)^{-1}K]e^{-s\delta}. \end{aligned} \tag{9}$$

Thus these expressions and some algebraic matrix computations yield the sensitivity and complementary sensitivity functions, which are as in the following definition.

Definition 5. *The sensitivity function $S(s)$ defined as the transfer function from the disturbances to the control system inputs and expressed by*

$$S(s) = [I - Q(s)C(SI - A + PC)^{-1}Be^{-s\delta}], \tag{10}$$

and the complementary sensitivity function $T(s)$ given the transfer function from the disturbances to the outputs of the controller; or equivalently $T(s) = I - S(s)$, and is given by

$$T(s) = Q(s)C(SI - A + PC)^{-1}Be^{-s\delta}. \tag{11}$$

Finally, the weighting sensitivity function $W_S(j\omega)$ is as given next.

Definition 6. *The weighting sensitivity $W_S(j\omega)$ is a function endowed with poles at the origin in order to improve the system performance at low frequencies, and expressed by*

$$\mid W_S(j\omega) \mid \begin{cases} = \infty, \omega = 0 \\ > 1, \;\; 0 < \omega < \omega_S \\ = 1, \;\; \omega = \omega_S \\ < 1, \;\; \omega_S < \omega \end{cases} \tag{12}$$
$$\omega_S < \omega_T$$

which is rewritten, for the purposes hereafter, as a sum of asymptotically stable $W_{S-}(s)$ and non-asymptotically stable $W_{S+}(s)$ terms as

$$W_S(s) = W_{S-}(s) + W_{S+}(s) \tag{13}$$

in which $W_{S+}(s)$ is such that

$$W_{S+}(s)I = C_S(sI - A_S)^{-1}B_S \tag{14}$$

where $I \in \Re^{m\times m}$, $A_S = \mathrm{diag}[A_{S1} \cdots A_{Sm}] \in \Re^{n_S \times n_S}$, $B_S = \mathrm{diag}[B_{S1} \cdots B_{Sm}] \in \Re^{n_S\times m}$, $C_S = \mathrm{diag}[C_{S1} \cdots C_{Sm}] \in \Re^{m\times n_S}$, and

$$C_{Sk}(sI - A_{Sk})^{-1}B_{Sk} = W_{S+}(s), \; \forall k. \tag{15}$$

4 Results

In this section, we show how to pursue a controller subject to the design objectives. Firstly, we quote a quite simple fact related to the sensitivity objective.

Proposition 1. *A necessary condition for the design objective $\| W_s(s)S(s) \|_\infty < 1$ be satisfied is that the poles of $W_S(s)$ located on the complex axis are canceled by the zeroes of $S(s)$.*

Proof. It is straightforward from noting that if there were no zero-pole cancellations, the condition would not be fulfilled.

Next, in order to translate the design specifications into expressions with filter terms we make use of the state feedback decoupling assumption in the following way.

Proposition 2. *Let $\varXi$, $\varPi$, $\varUpsilon$, Q and P be*

$$\begin{array}{l} \varXi(s) = \text{diag}\left[\frac{\upsilon_{i\mu_i}}{s^{\mu_i}+\upsilon_{i1}s^{\mu_i-1}+\cdots+\upsilon_{i\mu_i}}\right], \\ \varPi = [(A^{\mu_1} + \sum_{k=1}^{\mu_1} \upsilon_{1k}A^{\mu_1-k})b_1 \ \cdots \ (A^{\mu_m} + \sum_{k=1}^{\mu_m} \upsilon_{mk}A^{\mu_m-k})b_m], \\ \varUpsilon = \text{diag}\,[\upsilon_{1\mu_1} \ \cdots \ \upsilon_{m\mu_m}], \\ Q = \varUpsilon\varPsi^{-1}, \\ P = \varPi\varPsi^{-1} + KC^T(\varPsi^{-1})^T\varUpsilon^2\varPsi^{-1} \end{array} \tag{16}$$

where K, $K = K^T \geq 0$, is the solution of the Ricatti equation stabilizing $(A - \varPi\varPsi^{-1}C)^T - C^T(\varPsi^{-1})^T\varUpsilon^2\varPsi^{-1}CK$; i.e, the solution of

$$K(A - \varPi\varPsi^{-1}C)^T + (A - \varPi\varPsi^{-1}C)K - KC^T(\varPsi^{-1})^T\varUpsilon^2\varPsi^{-1}CK = 0 \tag{17}$$

Then

$$\begin{array}{l} QC(sI - A_P)^{-1}B = \varXi_I(s)\varXi(s) \\ A_P = A - PC \end{array} \tag{18}$$

where $\varXi_I(s)$ is an inner function given by

$$\varXi_I(s) = I - \varUpsilon\varPsi^{-1}C(sI - A_P)^{-1}KC^T(\varPsi^{-1})^T\varUpsilon \tag{19}$$

Proof. Firstly, from the state feedback decoupling of $G^T(s)$ we have that

$$\begin{array}{l} \varUpsilon\varPsi^{-1}C(sI - A + \varPi\varPsi^{-1}C)^{-1}B \\ = \varUpsilon\varPsi^{-1}[I + C(sI - A)^{-1}\varPi\varPsi^{-1}]^{-1} \times C(sI - A)^{-1}B \\ = \varXi(s) \end{array} \tag{20}$$

Now, pre and post multiplications by $\varUpsilon\varPsi^{-1}[I + C(sI - A)^{-1}\varPi\varPsi^{-1}]^{-1}\varPsi\varUpsilon$ and $\varXi^{-1}(s)$ render

$$\varUpsilon\varPsi^{-1}C(sI - A)^{-1}B\varXi^{-1}(s) = I + \varUpsilon\varPsi^{-1}C(sI - A)^{-1}\varPi\varUpsilon^{-1} \tag{21}$$

Since

$$\text{rank}\{\varUpsilon\varPsi^{-1}C(j\omega I - A)^{-1}B\varXi^{-1}(j\omega)\} = m, \forall\omega \geq 0 \tag{22}$$

(21) has no zeroes on the complex axis. Thus, first transposing (21), then applying inner-outer factorization [9] onto its right hand side, we have

$$\varUpsilon\varPsi^{-1}C(sI - A)^{-1}B\varXi^{-1}(s) = \varXi_O(s)\varXi_I(s) \tag{23}$$

where $\varXi_I(s)$ is as in (19) and $\varXi_O(s)$ is

$$\varXi_O(s) = I + \varUpsilon\varPsi^{-1}C(sI - A)^{-1}[\varPi\varUpsilon^{-1} + KC^T(\varPsi^{-1})^T\varUpsilon] \tag{24}$$

Now, to establish the proposition, pre and post multiply (23) by $\varXi_O^{-1}(s)$ and $\varXi(s)$, respectively; and carry out algebraic manipulations.

The matrix P as defined above allows us to quote the following simple but useful claim.

Corollary 1. *Let P be as in (16). Then, there exists a matrix T_1 such that the following relation is fulfilled.*

$$T_1A_S - A_PT_1 = BC_S \tag{25}$$

Proof. It is straightforward due to the fact that A_S and $A - PC$ have no common eigenvalues. Thus, using it and recalling the Ricatti equation [10], the result follows.

Moreover, a dual result as the one presented in [3] is established here.

Corollary 2. *Let Q be as in (16). Then, QCT_1 is a row full rank matrix, and (QCT_1, A_S) is observable. In addition, if*

$$T = \begin{bmatrix} QCT_1 \\ M \end{bmatrix} \tag{26}$$

is non singular for some matrix M. Then (A_{12}, A_{22}) obtained from

$$TA_ST^{-1} = \begin{array}{cc} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} & \begin{array}{c} m \\ n_s - m \end{array} \\ \begin{array}{cc} m & n_s - m \end{array} & \end{array} \tag{27}$$

is also observable.

Proof. It follows the same reasoning as presented in [3].

Taking these corollaries into consideration, we can define $Q_S(s)$ that yields the conditions required to guarantee proposition 1.

Proposition 3. *Let $Q_S(s)$ be*

$$\begin{aligned}
A_q &= A_{22} - K_SA_{12} \\
B_q &= A_{21} - K_SA_{11} + A_qK_S \\
C_q &= \bar{C}_ST^{-1}\begin{bmatrix} 0 \\ I_{n_s - m} \end{bmatrix} \\
D_q &= \bar{C}_ST^{-1}\begin{bmatrix} I_m \\ 0 \end{bmatrix} + C_qK_S \\
\bar{C}_S &= \mathrm{diag}[C_{s1}e^{A_{S1}\delta}, \cdots, C_{sm}e^{A_{Sm}\delta}].
\end{aligned} \tag{28}$$

Then the unstable poles of $W_{S+}(s)$ in $S(s)W_{S+}$ are canceled out.

Proof. It suffices to show that the poles of $W_{S+}(s)$ are unobservable in $S(s)W_{S+}$. For this, let us focus on the response of the system

$$[e^{s\delta}I - Q_S(s)QC(sI - A_P)^{-1}B]C_S(sI - A_S)^{-1}x_S(0),\ x_S(0) \in \Re^{n_S} \tag{29}$$

which is expressed by the system

$$\begin{aligned} &\dot{z}(t) = \hat{A}z(t) \\ &y_S(t) = \hat{C}z(t) \\ &z(0) = \begin{bmatrix} 0 \\ 0 \\ x_S(0) \end{bmatrix}, \hat{A} = \begin{bmatrix} A_q & B_qQC & 0 \\ 0 & A_P & BC_S \\ 0 & 0 & A_S \end{bmatrix}, \hat{C} = \begin{bmatrix} -C_q & -D_qQC & \bar{C}_S \end{bmatrix} \end{aligned} \tag{30}$$

Now, define $\hat{T}$ as

$$\hat{T} = \begin{bmatrix} I & 0 & T_2 \\ 0 & I & T_1 \\ 0 & 0 & I \end{bmatrix} \tag{31}$$

where T_2 is taken as

$$T_2 = \begin{bmatrix} -K_S & I \end{bmatrix} T. \tag{32}$$

From these and algebraic matrix calculations, one establishes the following expressions

$$\begin{aligned} T_2A_S - A_qT_2 &= B_qQCT_1 \\ C_qT_2 + D_qQCT_1 &= \bar{C}_S \end{aligned} \tag{33}$$

Thus, changing the coordinates of the system as $z(t) = \hat{T}\hat{z}(t)$, we have

$$\hat{T}\hat{A}\hat{T}^{-1} = \begin{bmatrix} A_q & B_qQC & 0 \\ 0 & A_P & 0 \\ 0 & 0 & A_S \end{bmatrix}, \hat{C}\hat{T} = \begin{bmatrix} -C_q & -D_qQC & 0 \end{bmatrix} \tag{34}$$

which means that in fact the unstable poles of $W_{S+}(s)$ are unobservable in $S(s)W_{S+}(s)$ as desired.

Finally, the controller design is accomplished by means of the following theorem.

Theorem 1. *Let f_i be an entry of Ξ and its denominator be written as*

$$\begin{cases} (s^2 + 2\rho\omega_n s + \omega_n^2)^{\frac{\mu_i}{2}}, & \mu_i : \text{even}, \\ (s^2 + 2\rho\omega_n s + \omega_n^2)^{\frac{\mu_i - 1}{2}}(s + \omega_n), & \mu_i : \text{odd}, \\ 0 < \omega_n \leq \omega_T, & 1/\sqrt{2} \leq \rho \leq 1 \end{cases} \tag{35}$$

and set $\rho = 1/\sqrt{2}$, and $\omega_n = \omega_T$. Then, if the graphs $1/\mid W_S(j\omega)\mid$, $\bar{\sigma}(S(j\omega))$, $1/\mid W_T(j\omega)\mid$, and $\bar{\sigma}(T(j\omega))$ are such that $1/\mid W_S(j\omega)\mid > \bar{\sigma}(S(j\omega))$ and $1/\mid W_T(j\omega)\mid > \bar{\sigma}(T(j\omega))$, which are graphical meanings of (1), for some values within the interval given in (35), then these ρ, ω establish a controller.

Proof. It follows straightforwardly from corollaries (1)-(2), and proposition (3).

Remark 2. The theorem yields a controller design procedure, which allows us to tune the system to guarantee the overall stability as well as the the performance. Basically, it means that one starts by plotting the graphs of $\mid W_S(j\omega) \mid$, $\bar{\sigma}(S(j\omega))$, $\mid W_T(j\omega) \mid$ and $\bar{\sigma}(T(j\omega))$ and aim to obtain a pattern as shown in figure 3. For this, the first step is to assure the stability of the system by decreasing ω_n initialized to ω_T, or increasing ρ from $1/\sqrt{2}$ up to 1. In terms of graphs, one looks for ω_n giving the maximum interval in between $\mid W_T(j\omega) \mid$ and $\bar{\sigma}(T(j\omega))$. Once stability is reached, one concerned also with the system performance will proceed to the tuning being aware of $\mid W_S(j\omega) \mid$ and $\bar{\sigma}(S(j\omega))$.Obviously, due to the relation $S(s) + T(s) = I$, we trade better system performance for lower stability margin, and vice versa.

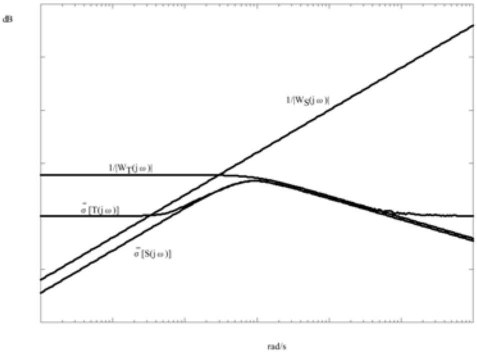


Fig. 3. Controller design philosophy

5 Numerical Example

In this section, a numerical example is presented to show how the computations work. Firstly, let the system and weighting functions be given by

$$\begin{aligned} G(s) &= \begin{bmatrix} \frac{-7.242(s-2.768)}{(s+1.958)(s+0.1165)} & \frac{-394.500(s-5.118)}{(s^s+5.840s+18.810)} \\ \frac{0.0799(s-7.410)}{(s+1.040)(s+0.130)} & \frac{-2.710(s-0.359)}{(s+1.430)(s+0.198)} \end{bmatrix}, \\ \delta &= 0.4 \\ W_S(s) &= \frac{0.01}{s^2}, \quad W_T(s) = \frac{(s+3)}{80} \end{aligned} \tag{36}$$

Then, starting with $\rho = 1/\sqrt{2}$ and $\omega_n = 80$ as in proposition 2, and plotting the graph as in figure 3 we tune the parameter to satisfy the design objectives, which is met for $\rho = 1/\sqrt{2}$ and $\omega_n = 70$. For these values, the matrix P is

$$\begin{bmatrix} 0.0332 & -0.0291 & -0.0317 & -0.0131 & -0.0196 & 0.0014 & 0.0010 & 0.0002 \\ -4.7916 & 4.1897 & 4.5755 & 1.8949 & 4.4760 & -0.5772 & -0.1414 & -0.0306 \end{bmatrix}^T . \tag{37}$$

Furthermore, the controller $Q(s) = C_q(sI - A_q)B_qQ + D_qQ$ is given by

$$\begin{aligned} A_q &= \begin{bmatrix} -0.1270 & 0.0000 \\ 0.0000 & -1.6500 \end{bmatrix}, \quad B_qQ = \begin{bmatrix} 0.2530 & -36.4860 \\ 46.3210 & -6693.2360 \end{bmatrix} \\ C_q &= \begin{bmatrix} 0.0040 & 0.0100 \\ 0.0829 & -0.0774 \end{bmatrix}, \quad D_qQ = \begin{bmatrix} -0.4456 & 64.3356 \\ 2.0344 & -295.1237 \end{bmatrix} \end{aligned} \tag{38}$$

6 Final Remarks

In this paper we proposed a controller design procedure for a networked control system with delays in the plant inputs and outputs. The controller is established by tuning filter parameters as in the PID control, which is the most common and widespread control scheme implemented in industrial processes. Thus, the design proposed here pursues a controller with the robustness characteristics of the modern control theory keeping in mind the reasoning adopted so far by the practitioner engineers.

References

1. Baillieul, J., Antsaklis, P.J.: Control and communications challenges in networked real time systems. Proceedings of IEEE 95(1), 9–28 (2007)
2. Nguang, S.K., Huang, D.: Robust control for uncertain networked control systems with random delays. LNCIS. Springer, Berlin (2009)
3. Watanabe, K., Izuta, G., Yamaguchi, T.: Output time delayed control system design subject to the mixed sensitivity problem. JSME International Journal 46(1), 232–238 (2003)
4. Izuta, G., Watanabe, K.: A note on the two disk mixed sensitivity design for control system with multiple output time delays. Systems and Control Letters 47(1), 375–382 (2002)
5. Francis, B., Doyle, J., Tannenbaum, A.: Feedback Control Theory. Macmillan, New York (1992)
6. Naghshtabrizi, P., Hespanha, J., Xu, Y.: Survey of recent results in networked control systems. Proceedings of IEEE 95(1), 9–28 (2007)
7. Branicky, M.S., Zhang, L., Phillips, S.M.: Stability of networked control systems. IEEE Control Systems Magazine 21(1), 84–99 (2001)
8. Falb, P.L., Wolovich, W.A.: Decoupling in the design and synthesis of multivariable control systems. IEEE TAC AC(12), 651–659 (1967)
9. Mita, T.: H_∞ control. Shokodo, Tokyo (1994) (in Japanese)
10. Suda, N., Kodama, S.: Matrix theory for control systems. SICE (Society of Instrumentation and Control Engineers of Japan), Tokyo (1978) (in Japanese)
11. Shi, Y., Bo, Y.: Output feedback stabilization of networked control systems with random delays modeled by markov chains. IEEE TAC 54(7), 1668–1674 (2009)

A Study of Detecting Computer Viruses in Real-Infected Files in the *n*-Gram Representation with Machine Learning Methods

Thomas Stibor

Fakultät für Informatik
Technische Universität München
thomas.stibor@in.tum.de

Abstract. Machine learning methods were successfully applied in recent years for detecting new and unseen computer viruses. The viruses were, however, detected in small virus loader files and not in real infected executable files. We created data sets of benign files, virus loader files and real infected executable files and represented the data as collections of n-grams. Our results indicate that detecting viruses in real infected executable files with machine learning methods is nearly impossible in the n-gram representation. This statement is underpinned by exploring the n-gram representation from an information theoretic perspective and empirically by performing classification experiments with machine learning methods.

1 Introduction

Detecting new and unseen viruses with machine learning methods is a challenging problem in the field of computer security. Signature based detection provides adequate protection against known viruses, when proper virus signatures are available. From a machine learning point of view, signature based detection is based on a prediction model where no generalization exists. In other words, no detection beyond the known viruses can be performed. To obtain a form of generalization, current anti-virus systems use heuristics generated by hand.

In recent years, however, machine learning methods are investigated for detecting unknown viruses [1–5]. Reported results show high and acceptable detection rates. The experiments, however, were performed on *skewed data sets*, that is, not real infected executable files are considered, but small executable virus loader files. In this paper, we investigate and compare machine learning methods on real infected executable files and virus loader files. Results reveal that real infected executable files when represented as n-grams can *barely* be detected with machine learning methods.

2 Data Sets

In this paper, we focus on computer virus detection in DOS executable files. These files are executable on DOS and some Windows platforms only and can

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 509–519, 2010.

be identified by the leading two byte signature "MZ" and the file name suffix ".EXE". It is important to note that although DOS platforms are obsolete, the princinple of virus infection and detection on DOS platforms can be generalized to other OS platforms and executable files.

In total three different data sets are created. The benign data set consists of virus-free executable DOS files which are collected from public FTP servers. The collected files consist of DOS games, compilers, editors, etc. and are preprocessed such that no duplicate files exist. Additionally, files that are compressed with executable packers[1] such as lzexe, pklite, diet, etc. are uncompressed. The preprocessing steps are performed to obtain a clean data set which is supposed to induce high detection rates of the considered machine learning methods. In total, we collected 3514 of such "clean" DOS executable files.

DOS executable virus loader files are collected from the VXHeaven website.[2] The same processing steps are applied, that is, all duplicate files are removed and compressed files are uncompressed. It turned out however, that some files from VXHeaven are real infected executable files and not virus loader files. These files can be identified by inspecting their content with a disassembler/hexeditor and by their large file sizes. Consequently, we removed all files greater than 10 KBytes and collected in total 1555 virus loader files.

For the sake of clarity, the benign data set is denoted as $\mathcal{B}_{\text{files}}$, the virus loader data set as $\mathcal{L}_{\text{files}}$.

The data set of real infected executable files denoted as $\mathcal{I}_{\text{files}}$ is created by performing the following steps:

```
1: $\mathcal{I}_{\texttt{files}} := \{\}$
2: for each file.v in $\mathcal{L}_{\texttt{files}}$
3:  setup new DOS environment in DOS emulator
4:  randomly determine file.r from $\mathcal{B}_{\texttt{files}}$
5:  boot DOS and execute file.v and file.r
6:  if (file.r gets infected)
7:    $\mathcal{I}_{\texttt{files}} := \mathcal{I}_{\texttt{files}} \cup \{$file.r$\}$
```

Fig. 1. Steps performed to create real infected executable DOS files

These steps were executed on a GNU/Linux machine by means of a Perl script and a DOS emulator. A DOS emulator is chosen since it is too time consuming to setup a clean DOS environment on a DOS machine whenever a new file gets infected. The statement "`file.r gets infected`" at line 6 in Figure 1 denotes the data labeling process. Since only viruses were considered for which signatures are available, *no* elements in the created data sets are mislabeled. In total 1215 real infected files were collected.[3]

[1] Executable compression is frequently used to confuse and hide from virus scanners.

[2] Available at `http://vx.netlux.org/`

[3] For the sake of verifiability, the three data sets are available at `http://www.sec.in.tum.de/~stibor/ieaaie2010/`

3 Transforming Executable Files to n-Grams and Frequency Vectors

Raw executable files cannot serve properly as input data for machine learning methods. Consequently, a suitable representation is required. A common and frequently used representation is the hexdump. A hexdump is a hexadecimal representation of an executable file (or of data in general) and is related to machine instructions, termed opcodes. Figure 2 illustrates the connection between machine instructions, opcodes, and a hexdump. It is important to note, that the hexdump representation used and reported in the literature, is computed over a *complete* executable file which consists of a header, relocation tables and the executable code. The machine instructions are located in the code segment only and this implies that a hexdump obtained over a complete executable file may not be interpreted completely as machine instructions.

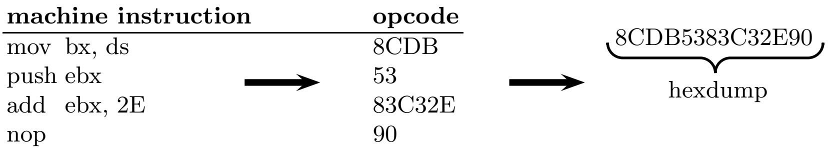


Fig. 2. Relation between machine instructions, opcodes and the resulting hexdump

In the second preprocessing step, the hexdump is "cut" into substrings of length $n \in \mathbb{N}$, denoted as n-grams. According to the literature an n-gram can be any set of characters in a string [6]. In terms of detecting viruses in executable files, n-grams are adjacent substrings created by shifting a window of length $s \in \mathbb{N}$ over the hexdump. The resulting number of created n-grams depends on the value of n and the window shift length s.

We created for each file an ordered collection[4] of n-grams by cutting the hexdump from left to right. Note that the collection can contain the same n-grams multiple times. The total number of n-grams in the collection when given n and s is $\lfloor (l - n)/s + 1 \rfloor$, where l denotes the hexdump length. In a final step, the collection is transformed into a vector of dimension[5] $d = 16^n$, where each component corresponds to the absolute frequency of the n-gram occurring in the collection. In the problem domain of text classification such vectors are named term frequency vectors [7]. We therefore denote such vectors as n-gram frequency vectors.

4 Creating and Selecting Informative n-Grams

For large values of n, the created n-gram frequency vectors are limited processable due to their high-dimensional nature. In this section we therefore address

[4] If not otherwise stated we denote an ordered collection as collection.

[5] An unary hexadecimal number can take 16 values.

the problem of *creating* and *selecting* the most “informative” n-grams. More precisely, the following two problems are addressed:

- How to *determine* the parameters n and s such that the most “informative” n-grams can be created?
- How to *select* the most “informative” n-grams once they are created?

Both problems can be worked out by using arguments from information theory.

Let X be a random variable that represents the outcome of an n-gram and $\mathcal{X}$ the set of all n-grams of length n. The entropy is defined as

$$H(X) = \sum_{x \in \mathcal{X}} P(X = x) \log_2 \frac{1}{P(X = x)}. \tag{1}$$

Informally speaking, high entropy implies that X is from a roughly uniform distribution and n-grams sampled from it are distributed all over the place. Consequently, predicting sampled n-grams is “hard”. In contrast, low entropy implies that X is from a varied distribution and n-grams sampled from it would be more predictable.

4.1 Information Loss in Frequency Vectors

Given a hexdump of a file, the resulting n-gram collection $\mathcal{C}$ and the created n-gram frequency vector $\mathbf{c}$. We are interested in minimizing the amount of information loss for reconstructing the n-gram collection $\mathcal{C}$ when given $\mathbf{c}$. For the sake of simplicity we consider the case $n = s$ and denote the cardinality of $\mathcal{C}$ as t. Assume there are r unique n-grams in $\mathcal{C}$, where each n-gram appears n_i times such that $\sum_{i=1}^{r} n_i = t$. The collection $\mathcal{C}$ can have

$$\frac{t!}{n_1! n_2! \cdots n_r!} = \binom{t}{n_1, n_2, \ldots, n_r} \tag{2}$$

many rearrangements of n-grams. Applying the theorem from information theory on the size of a type class [8], (pp. 350, Theorem 11.1.3) it follows:

$$\frac{1}{(t+1)^{|\Sigma|}} 2^{tH(P)} \leq \binom{t}{n_1, n_2, \ldots, n_r} \leq 2^{tH(P)} \tag{3}$$

where P denotes the empirical probability distribution of the n-grams, $H(P)$ the entropy of distribution P and Σ the used alphabet. Using $n = s$ it follows from (3) that $2^{\lfloor l/n \rfloor H(P)}$ is an upper bound of the information loss. To keep the upper bound as small as possible, the value of n has to be close to the value of l and the empirical probability distribution P has to be a varied distribution rather than a uniform distribution.

4.2 Entropy of n-Gram Collections

To support the statement in Section 4.1 empirically, the entropy values of the created n-gram collections are calculated. The values are set in relation to the

Table 1. Results of the expression $H(X)/H_{\max}$ for different parameter combinations of n and s. Smaller values indicate that more predictive n-grams can be created. The upper value in each row denotes the result of n-grams created from benign set $\mathcal{B}_{\text{files}}$, the middle value from virus infected set $\mathcal{I}_{\text{files}}$, the lower value from virus loader set $\mathcal{L}_{\text{files}}$.

s \ n	2	3	4
1	0.923 0.924 0.738	0.896 0.895 0.679	0.862 0.861 0.628
2	0.889 0.888 0.708	0.857 0.857 0.649	0.818 0.816 0.593
3		0.896 0.895 0.679	0.862 0.860 0.626
4			0.816 0.813 0.590

maximum possible entropy (uniform distribution) denoted as $H_{\max}$. An n-gram collection where the corresponding value of $H(X)/H_{\max}$ is close to zero, contains more predictive n-grams than the one where $H(X)/H_{\max}$ is close to one. The results for different parameter combinations of n and s are listed in Table 1.

One can see that larger n-grams lengths are more predictable than short ones. Additionally, one can observe that window shift length s has to be a multiple of two, because opcodes most frequently occur as two and four byte values. Moreover, one can observe that the result of $H(X)/H_{\max}$ for n-gram collections created from the benign set $\mathcal{B}_{\text{files}}$ and virus infected set $\mathcal{I}_{\text{files}}$ have approximately the same values. In other words, discriminating between those two classes is limited realizable. This observation is also identifiable in Section 6 where the classification results are presented. From a practical point of view, however, large values of n induce a computational complexity which is not manageable. As a consequence, we focus in the paper on n-gram lengths 2 and 3. We tried to run the classification experiments for $n = 4, 5, 6, 7, 8$ in combination with a dimensionality reduction described in the subsequent section, however, due to the resulting space complexity and lack of memory space[6] no results were obtained.

4.3 Feature Selection

The purpose of feature selection is to reduce the dimensionality of the data such that the machine learning methods can be applied more efficiently in terms of time and space complexity. Furthermore, feature selection often increases classification accuracy by eliminating noisy features. We apply on our data sets the mutual information method [7]. This method is used in previous work and appears to be practical [5]. A comparison study of other feature selection methods in terms of classifying malicious executables is provided in [9].

[6] A Quad-Core AMD 2.6 Ghz with 16 GB RAM was used.

Mutual Information Selection: Let U be a random variable that takes values $e_{ng} = 1$ (the collection contains n-gram ng) and $e_{ng} = 0$ (the collection does not contain ng). Let C be a random variable that takes values $e_c = 1$ (the collection belongs to class c) and $e_c = 0$ (the collection does not belong to class c).

The mutual information measures how much information the presence/absence of an n-gram contributes to making the correct classification decision on c. More formally

$$\sum_{e_{ng} \in \{0,1\}} \sum_{e_c \in \{0,1\}} P(U = e_{ng}, C = e_c) \cdot \log_2 \frac{P(U = e_{ng}, C = e_c)}{P(U = e_{ng}) P(C = e_c)}. \tag{4}$$

5 Classification Methods

In this section we briefly introduce the classification methods which are used in the experiments. These methods are frequently applied in the field of machine learning.

5.1 Naive Bayesian Classifier

A naive Bayesian classifier is well known in the machine learning community and is popularized by applications such as spam filtering or text classification in general.

Given feature variables $F_1, F_2, \ldots, F_n$ and a class variable C. The Bayes' theorem states

$$P(C \,|\, F_1, F_2, \ldots, F_n) = \frac{P(C)\, P(F_1, F_2, \ldots, F_n \,|\, C)}{P(F_1, F_2, \ldots, F_n)}. \tag{5}$$

Assuming that each feature F_i is conditionally independent of every other feature F_j for $i \neq j$ one obtains

$$P(C \,|\, F_1, F_2, \ldots, F_n) = \frac{P(C) \prod_{i=1}^{n} P(F_i \,|\, C)}{P(F_1, F_2, \ldots, F_n)}. \tag{6}$$

The denominator serves as a scaling factor and can be omitted in the final classification rule

$$\underset{c}{\operatorname{argmax}}\ P(C = c) \prod_{i=1}^{n} P(F_i = f_i \,|\, C = c). \tag{7}$$

5.2 Support Vector Machine

The Support Vector Machine (SVM) [10, 11] is a classification technique which has its foundation in computational geometry, optimization and statistics. Given a sample

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_l, y_l) \in \mathbb{R}^d \times Y, \quad Y = \{-1, +1\}$$

and a (nonlinear) mapping into a potentially much higher dimensional feature space $\mathcal{F}$

$$\Phi : \mathbb{R}^d \rightarrow \mathcal{F}, \qquad \mathbf{x} \mapsto \Phi(\mathbf{x}).$$

The goal of SVM learning is to find a separating hyperplane with maximal margin in $\mathcal{F}$ such that

$$y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) \geq 1, \quad i = 1, \ldots, l, \tag{8}$$

where $\mathbf{w} \in \mathcal{F}$ is the normal vector of the hyperplane and $b \in \mathbb{R}$ the offset. The separation problem between the -1 and $+1$ class can be formulated in terms of the quadratic convex optimization problem

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \; \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{l} \xi_i \tag{9}$$

$$\text{subject to } \; y_i(\langle \mathbf{w}, \Phi(\mathbf{x}_i)\rangle + b) \geq 1 - \xi_i, \tag{10}$$

where C is a penalty term and $\xi_i \geq 0$ slack variables to relax the hard constrains in (8). Solving the dual form of (9) and (10) leads to the final decision function

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}_i)\rangle + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{l} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right) \tag{11}$$

where $k(\cdot,\cdot)$ is a kernel function satisfies the Mercer's condition [12] and α_i Lagrangian multipliers obtained by solving the dual form.

In the performed classification experiments we used the Gaussian kernel $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|^2/\gamma)$ as it gives the highest classification results compared to other kernels. The hyperparameters γ and C are optimized by means of the inbuilt validation method of the used R package `kernlab` [13].

6 ROC Analysis and Results

ROC analysis is performed to measure the goodness of the classification results. More specifically, we are interested in the following quantities:

- true positives (TP): number of virus executable examples classified as virus executable,
- true negatives (TN): number of benign executable examples classified as benign executable,
- false positives (FP): number of benign executable examples classified as virus executable,
- false negatives (FN): number of virus executable examples classified as benign executable.

The *detection rate* (true positive rate), *false alarm rate* (false positive rate), and *overall accuracy* of a classier are calculated then as follow

$$\text{detection rate} = \frac{TP}{TP + FN}, \qquad \text{false alarm rate} = \frac{FP}{FP + TN}$$

$$\text{overall accuracy} = \frac{TP + TN}{TP + TN + FP + FN}.$$

Table 2. Classification results for n-gram length $n = 2$ and sliding window $s = 1, 2$ without feature selection

benign vs. virus loader

s	Method	Detection Rate	False Alarm Rate	Overall Accuracy
1	SVM	**0.9499** (±0.0191)	**0.0272** (±0.0075)	**0.9654** (±0.0059)
	Bayes	0.6120 (±0.0459)	0.0710 (±0.0110)	0.8315 (±0.0194)
2	SVM	**0.9531** (±0.0120)	**0.0261** (±0.0091)	**0.9674** (±0.0083)
	Bayes	0.7267 (±0.0244)	0.0692 (±0.0136)	0.8684 (±0.0122)

benign vs. virus infected

1	SVM	0.0370 (±0.0161)	**0.0731** (±0.0078)	**0.6982** (±0.0155)
	Bayes	**0.6039** (±0.0677)	0.6113 (±0.0679)	0.4438 (±0.0416)
2	SVM	0.0556 (±0.0153)	**0.0888** (±0.0125)	**0.6912** (±0.0206)
	Bayes	**0.6682** (±0.0644)	0.6533 (±0.0736)	0.4284 (±0.0465)

Table 3. Classification results for n-gram length $n = 2$ and sliding window $s = 1, 2$ with selecting 50 % of the original features by means of the mutual information method

benign vs. virus loader with feature selection

s	Method	Detection Rate	False Alarm Rate	Overall Accuracy
1	SVM	**0.9443** (±0.0151)	**0.0324** (±0.0057)	**0.9605** (±0.0035)
	Bayes	0.7897 (±0.0297)	0.1123 (±0.0187)	0.8577 (±0.0178)
2	SVM	**0.9354** (±0.0099)	**0.0353** (±0.0129)	**0.9558** (±0.0105)
	Bayes	0.8844 (±0.0301)	0.1653 (±0.0284)	0.8498 (±0.0223)

benign vs. virus infected with feature selection

1	SVM	0.0251 (±0.0128)	**0.0702** (±0.0201)	**0.6974** (±0.0213)
	Bayes	0.6084 (±0.0502)	0.5860 (±0.0456)	0.4637 (±0.0287)
2	SVM	0.0264 (±0.0120)	**0.0782** (±0.0147)	**0.6916** (±0.0184)
	Bayes	**0.6270** (±0.0577)	0.6120 (±0.0456)	0.4495 (±0.0312)

Moreover, to avoid under/overfitting effects of the classifiers, we performed a K-crossfold validation. That is, the data set is split in K roughly equal-sized parts. The classification method is trained on $K - 1$ parts and has to predict the not seen testing part. This is performed for $k = 1, 2, \ldots, K$ and combined as the mean value to estimate the generalization error. Since the data sets are large, a 10-crossfold validation is performed. The classification results[7] are depicted in Tables 2-5, where also the standard deviation result of the crossfold validation is provided.

The classification results evidently show (cf. Tables 2-5) that high detection rates and low false alarm rates can be obtained when discriminating between benign files and virus loader files. This observation supports previously published results [1, 2, 5]. However, for detecting viruses in real infected executables these methods are barely applicable. This observation is not a great surprise, because benign files and real infected files are from a statistical point of view nearly indistinguishable.

It is interesting to note that the SVM gives excellent classification results when discriminating between benign and virus loader files. However, poor results when discriminating between benign and virus infected files. In terms of the overall accuracy, the SVM still outperforms the Bayes method.

[7] We performed the experiments also with k-Nearest Neighbor classifier and Multilayer Perceptron and achieved nearly similar classification results. Due to lack of space, the results are not presented in this paper.

Table 4. Classification results for n-gram length $n = 3$ and sliding window $s = 1, 2, 3$ without feature selection

benign vs. virus loader

s	Method	Detection Rate	False Alarm Rate	Overall Accuracy
1	SVM	**0.9622** (±0.0148)	**0.0267** (±0.0080)	**0.9700** (±0.0080)
	Bayes	0.7493 (±0.0302)	0.0594 (±0.0106)	0.8820 (±0.0076)
2	SVM	**0.9560** (±0.0183)	**0.0266** (±0.0091)	**0.9680** (±0.0095)
	Bayes	0.8044 (±0.0210)	0.0595 (±0.0142)	0.8985 (±0.0108)
3	SVM	**0.9618** (±0.0089)	**0.0289** (±0.0094)	**0.9682** (±0.0071)
	Bayes	0.7488 (±0.0377)	0.0580 (±0.0101)	0.8828 (±0.0146)

benign vs. virus infected

1	SVM	0.0868 (±0.0266)	**0.0997** (±0.0162)	**0.6912** (±0.0239)
	Bayes	**0.6152** (±0.0742)	0.6110 (±0.0661)	0.4476 (±0.0352)
2	SVM	0.1155 (±0.0345)	**0.0979** (±0.0190)	**0.7001** (±0.0186)
	Bayes	**0.6668** (±0.0877)	0.6386 (±0.1169)	0.4398 (±0.0679)
3	SVM	0.0665 (±0.0204)	**0.0944** (±0.0157)	**0.6899** (±0.0195)
	Bayes	**0.6110** (±0.0844)	0.5995 (±0.0980)	0.4542 (±0.0563)

Table 5. Classification results for n-gram length $n = 3$ and sliding window $s = 1, 2, 3$ with selecting 50 % of the original features by means of the mutual information method

benign vs. virus loader with feature selection

s	Method	Detection Rate	False Alarm Rate	Overall Accuracy
1	SVM	**0.9619** (±0.0173)	0.0334 (±0.0086)	**0.9648** (±0.0066)
	Bayes	0.9281 (±0.0195)	0.1702 (±0.0242)	0.8597 (±0.0182)
2	SVM	**0.9620** (±0.0203)	0.0320 (±0.0082)	**0.9660** (±0.0107)
	Bayes	0.8205 (±0.0358)	0.0701 (±0.0151)	0.8962 (±0.0200)
3	SVM	**0.9549** (±0.0092)	0.0325 (±0.0073)	**0.9636** (±0.0064)
	Bayes	0.7362 (±0.0368)	0.0429 (±0.0094)	0.8891 (±0.0126)

benign vs. virus infected with feature selection

1	SVM	0.1057 (±0.0281)	**0.1024** (±0.0184)	**0.6944** (±0.0196)
	Bayes	**0.6380** (±0.0659)	0.6159 (±0.0829)	0.4482 (±0.0500)
2	SVM	0.1406 (±0.0372)	**0.1010** (±0.0220)	**0.7037** (±0.0276)
	Bayes	**0.6643** (±0.0844)	0.6287 (±0.1157)	0.4455 (±0.0657)
3	SVM	0.0611 (±0.0185)	**0.1001** (±0.0165)	**0.6842** (±0.0253)
	Bayes	**0.6075** (±0.0762)	0.5668 (±0.0823)	0.4764 (±0.0520)

Additionally, one can observe that selecting 50 % of the features by means of the mutual information criterion does not significantly increase and decrease the classification accuracy of the SVM. The Bayes method, however, benefit by the feature selection preprocessing step as a significant increase of the classification accuracy can be observed.

7 Conclusion

We investigated the applicability of machine learning methods for detecting viruses in real infected DOS executable files when using the n-gram representation. For that reason, three data sets were created. The benign data set contains virus free DOS executable files collected from all over the Internet. For the sake of comparison to previous work, we created a data set which consists of virus loader files collected from the VXHeaven website. Furthermore, we created a new data set which consists of real infected executable files. The data sets were transformed to collections of n-grams and represented as frequency vectors. Due to this transformational step a information loss occurs. We showed empirically and theoretically with arguments from information theory that this loss can be minimized by increasing the lengths of the n-grams. As a consequence however, this results in a computational complexity which is not manageable.

Moreover, we calculated the entropy of the n-gram collections and observed that the benign files and real infected files nearly have identical entropy values. As a consequence, discriminating between those two classes is limited realizable. Furthermore, we performed classification experiments and confirmed that discriminating between those two classes is limited realizable. More precisely, the SVM gives unacceptable detection rates. The Bayes and the cosine measure gives higher detection rates, however, they also give unacceptable false alarm rates.

In summary, our results doubt the applicability of detecting viruses in *real* infected executable files with machine learning methods when using the (naive) n-gram representation. However, it has not escaped our mind that learning algorithms for sequential data could be one approach to tackle this problem more effectively. Another promising approach is to learn the *behavior* of a virus, that is, the sequence of system calls a virus is generating. Such a behavior can be collected in a secure sandbox and learned with appropriate machine learning methods.

References

1. Schultz, M.G., Eskin, E., Zadok, E., Stolfo, S.J.: Data mining methods for detection of new malicious executables. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 38–49. IEEE Computer Society, Los Alamitos (2001)
2. Abou-Assaleh, T., Cercone, N., Keselj, V., Sweidan, R.: Detection of new malicious code using n-grams signatures. In: Second Annual Conference on Privacy, Security and Trust, pp. 193–196 (2004)
3. Reddy, D.K.S., Pujari, A.K.: *N*-gram analysis for computer virus detection. Journal in Computer Virology 2(3), 231–239 (2006)
4. Yoo, I.S., Ultes-Nitsche, U.: Non-signature based virus detection. Journal in Computer Virology 2(3), 163–186 (2006)
5. Kolter, J.Z., Maloof, M.A.: Learning to detect and classify malicious executables in the wild. Journal of Machine Learning Research 7, 2721–2744 (2006)
6. Cavnar, W.B., Trenkle, J.M.: N-gram-based text categorization. In: Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval, pp. 161–175 (1994)
7. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Cambridge University Press, Cambridge (2008)
8. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. Wiley Interscience, Hoboken (2006)
9. Cai, D.M., Gokhale, M., James, J.T.: Comparison of feature selection and classification algorithms in identifying malicious executables. Computational Statistics & Data Analysis 51(6), 3156–3172 (2007)
10. Boser, B.E., Guyon, I.M., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory (COLT), pp. 144–152. ACM Press, New York (1992)

11. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
12. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
13. Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A.: kernlab – an S4 package for kernel methods in R. Journal of Statistical Software 11(9), 1–20 (2004)

Data Mining Strategies for CRM Negotiation Prescription Problems[*]

Antonio Bella, Cèsar Ferri, José Hernández-Orallo,
and María José Ramírez-Quintana

DSIC-ELP, Universidad Politécnica de Valencia, Camí de Vera s/n,
46022 Valencia, Spain

Abstract. In some data mining problems, there are some input features that can be freely modified at prediction time. Examples happen in retailing, prescription or control (prices, warranties, medicine doses, delivery times, temperatures, etc.). If a traditional model is learned, many possible values for the special attribute will have to be tried to attain the maximum profit. In this paper, we exploit the relationship between these modifiable (or negotiable) input features and the output to (1) change the problem presentation, possibly turning a classification problem into a regression problem, and (2) maximise profits and derive negotiation strategies. We illustrate our proposal with a paradigmatic Customer Relationship Management (CRM) problem: maximising the profit of a retailing operation where the price is the negotiable input feature. Different negotiation strategies have been experimentally tested to estimate optimal prices, showing that strategies based on negotiable features get higher profits.

1 Introduction

In data mining, problem features (or attributes) have been usually classified as input and output features. A problem is said to be supervised if it has output features, and it is said to be unsupervised if it does not have output features. Input features can be of many kinds: numerical, nominal, structured, etc. In fact, many data mining methods have been specialised to specific kinds of input features. In supervised learning, it is usually assumed that the goal of a model is to predict an output value given an input. Consequently, a function is learned from inputs to outputs, which is eventually applied to new cases.

However, in many application areas not all input feature values are given. This does not mean that they are unknown (i.e., null), but that they can be modified or fine-tuned at prediction time. Consider a typical data mining problem: a loan granting model where loans are granted or not according to a model which has been learnt from previous customer behaviours. It is generally assumed that given the input feature values (the customer's personal information, the loan

[*] This work has been partially supported by the EU (FEDER) and the Spanish MEC/MICINN, under grant TIN 2007-68093-C02 and the Spanish project "Agreement Technologies" (Consolider Ingenio CSD2007-00022).

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 520–530, 2010.

amount, the operation's data, etc.) the model will provide an output (yes/no, a probability, a profit, etc.). But in real scenarios, the decision of whether the loan must be granted or not might change if one or more of the input feature values can be changed. Perhaps, a loan cannot be granted for 300,000 euros, but it can be granted for 250,000 euros. If the customer asks the bank's clerk "what is the maximum amount you can grant for this operation?", we have a sort of *inverse* problem. Since the model is not generally an analytical one (it is not generally a linear regression model but a neural network, SVM, decision tree or other kind of difficult-to-inverte models), the only possibility to give a precise answer to the customer is to try all the possible input combinations for the loan amount and find out the maximum value which is granted. After this inefficient and ad-hoc process, the clerk can give an answer such as: "According to our loan models, we can grant a maximum of 278,304 euros". Apart from this inefficiency, there is another more serious problem: the clerk knows that the model will give a negative answer to any amount above this maximum, for instance, 278,305 euros, which makes this loan a quite risky one.

This typical example shows that some input features are crucial in the way that they can be freely modified at prediction time. The existence of these special attributes makes it quite inefficient to develop a classification/regression model in the classical way, since whenever there is a new instance hundreds of possible values have to be tried for the special attribute in order to see which combination can attain the maximum profit, or, as the previous example, the maximum risk. Additionally, these features are frequently associated to negotiation scenarios, where more than one attempt or offer have to be made, by suitably choosing different values for this special input feature.

In this paper we analyse these special features that we call "negotiable features", and how they affect data mining problems, its presentation and its use for confidence probability estimation and decision making. As we will see, we can exploit the relation between these input features and the output to change the problem presentation. In this case, a classification problem can be turned into a regression problem over an input feature [1]. Also, we present a first general systematic approach on how to deal with these features in supervised models and how to apply these models in real negotiation scenarios where there are several attempts for the output value.

The paper is organised as follows. In Section 2 we focus on classification problems with one numerical negotiable feature, since this is the most frequent and general case, and it also includes prototypical cases, when the negotiable feature is price or time. We present a general approach to the inversion problem which transforms the classification problem into a regression one, where the negotiable feature is placed as the output. In Section 3, we describe a specific real scenario (an estate agent's) where negotiation can take place. We also develop some negotiation strategies using the previous approaches in Section 4. In Section 5, we experimentally evaluate models and negotiation strategies with a baseline approach. Section 6 includes the conclusions and future work.

2 Inverting Problem Presentation

As we have mentioned in the introduction, there are many data mining problems where one or more input attributes, we call negotiable features, can be modified at application time. Imagine a model which estimates the delivery time for an order depending on the kind of product and the units which are ordered. One possible (traditional) use of this model is to predict the delivery time given a new order. However, another use of this model is to determine the number of units (provided it is possible to play with this value) that can be delivered in a fixed period of time, e.g. one week. This is an example of an “inverse use” of a data mining model, where all inputs except one and the output are fixed, and the objective is to determine the remaining input value.

The inversion problem can be defined as follows. Consider a supervised problem, where input attribute domains are denoted by X_i, $i \in \{1, \ldots, m\}$, and the output attribute domain is denoted by Y. We denote the target (real) function as $f : X_1 \times X_2 \times \ldots \times X_m \to Y$. Values for input and output attributes will be denoted by lowercase letters. Hence, labelled instances are then tuples of the form $\langle x_1, x_2, \ldots, x_m, y\rangle$ where $x_i \in X_i$ and $y \in Y$. The inversion problem consists in defining the function $f^I : X_1 \times \ldots \times X_{i-1} \times Y \times X_{i+1} \times \ldots \times X_m \to X_i$, where X_i is the negotiable feature. In the above example, f is the function that calculates the delivery time of an order, the negotiable feature X_i is the number of delivered units and f^I calculates this number by considering fixed the delivery time.

In the inverting problem the property that we call *sensitive* is satisfied. Fixing the values of all the other input attributes $X_j \neq X_i$ for at least n examples from the dataset D ($n \leq |D|$ being n determined by the user depending on the problem, the presence of noise, $\ldots$), there are two different values for X_i producing different output values.

We also assume a monotonic dependency between the input attribute X_i and the output. This dependency is defined under the assumption that there is a strict total order relation for the output, denoted by $\prec$, such that for every two different possible values $y_a, y_b \in Y$, we have that either $y_a \prec y_b$ or $y_b \prec y_a$. This order usually represents some kind of profit, utility or cost. For numerical outputs, $\prec$ is usually the order relation between real numbers (either $<$ or $>$, depending on whether it is a cost or profit). For nominal outputs, $\prec$ usually sets an order between the classes. For binary problems, where POS and NEG represent the positive and negative class respectively, we can just set that $NEG \prec POS$. For more than two classes, the order relation can be derived from the cost of each class. Analogously, there is also a total order relation for the input denoted as $\preceq$. Based on this order, we can establish a monotonic dependency between the input and the output features. Thus, $\forall a, b \in X_i$, if $a \preceq b$ then $f(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_m) \prec f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_m)$ (monotonically increasing) or $\forall a, b \in X_i$, if $a \preceq b$ then $f(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_m) \succeq f(x_1, \ldots, x_{i-1}, b, x_{i+1}, \ldots, x_m)$ (monotonically decreasing).

The inversion problem is well-known [1] and seems simple at first sight, but many questions arise. First, is f^I also a function? In other words, for two different values for X_i we may have the same value for Y which will ultimately translate into two inconsistent examples for f^I (two equal inputs giving different outputs). Second, the fact that we have an example saying that a given loan amount was granted to a customer does not mean that this is the maximum amount that could be granted to the customer. Third, deriving probabilities to answer questions such as “which loan amount places this operation at a probability of 0.95 of being a profitable customer?” seem to be unsolvable with this new presentation.

But if we take a closer look at these issues, we see that although relevant, there is still a possibility behind this problem presentation change. First, many regression techniques work well for inconsistent examples, so this is not a big practical problem. Second, it is true that cases do not represent the maximum amount, but in many cases the examples represent deals and they are frequently not very far away from the maximum. Or, in the worst case, we can understand the new task as “inferring” the typical value for X_i such that the loan is granted to the customer. And third, we *can* also obtain probabilities in a regression problem. We extend this idea further below.

If we invert the problem, how can we address the original problem again? With the original model and for only two classes, it can be done by calculating $p(POS|\langle x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_m\rangle)$, for any possible value $a \in X_i$. From the inverted (regression) problem, we get: $\hat{a} = f^I(x_1, \ldots, x_{i-1}, POS, x_{i+1}, \ldots, x_m)$. If we think of $\hat{a}$ as the predicted maximum or minimum for a which makes a change on the class, a reasonable assumption is to give 0.5 probability for this, that is $p(POS|\langle x_1, \ldots, x_{i-1}, \hat{a}x_{i+1}, \ldots, x_m\rangle) = 0.5$.

The next step is to assume that the output for f^I follows a distribution with centre at $\hat{a}$. For instance, we can assume a normal distribution with mean at $\hat{a}$ and use the standard error (mean absolute error, mae, on the training set) as standard deviation σ. In other words, we use $N(\hat{a}, mae^2)$. Figure 1 shows an example of a normal distribution with centre at $\hat{a} = 305,677.9$ and standard deviation $\sigma = 59,209.06$ and its associated cumulative distribution function.

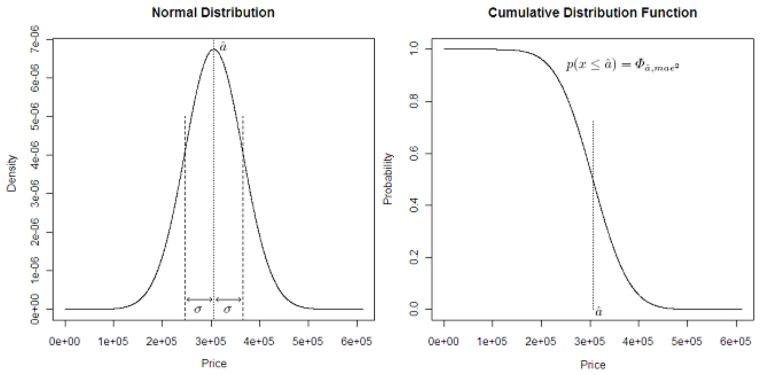


Fig. 1. Left: Example of a normal distribution $\hat{a} = 305,677.9$ and $\sigma = 59,209.06$. Right: Associated cumulative distribution function.

From here, we can derive the probability for any possible value a as the cumulative distribution function derived from the above normal, i.e., $\Phi_{\hat{a},mae^2}$.

Consequently, for solving the original problem, (1) we solve the inversion problem directly and (2) we use the predicted value of the negotiable feature as mean of a normal distribution with the standard error as standard deviation. We call this model *negotiable feature model.*

3 Negotiation Using Negotiable Feature Models: A Real Scenario

We are going to illustrate the approach we have presented in the previous section in a real example. In particular, we have studied the problem of retailing, where the (negotiable) input feature is the price (denoted by π) and the problem is a classification problem (buying or not).

We present an example using real data from an estate agent's, which sells flats and houses they have in their portfolio. We have several conventional attributes describing the property (squared metres, location, number of rooms, etc.), and a special attribute which is our negotiable feature, price. We will use the term "product" for properties to make it clear that the case is directly extensible to virtually any retailing problem where price is negotiable.

We start with the simplest negotiation scenario, where there are only one seller and one buyer who both negotiate for one product. One buyer is interested in one specific product. S/he likes the product and s/he will buy the product if its price is under a certain price that s/he is willing to pay for this product.In fact, if we reduce price to 0, the probability of having class POS approaches 1 and if we increase price to a very large amount, the probability of having class NEG approaches 1. Moreover, the relation between price and the class order $NEG \prec POS$ is monotonically decreasing.

Additionally, in our problem, the seller has a "minimum price" (denoted by π_{min}), which corresponds to the price that the owner has set when the product was included in the portfolio plus a quantity that the seller sets as fixed and variable costs. Any increment over this minimum price is profitable for the seller. Conversely, selling under this value is not acceptable for the seller. Therefore, the seller will not sell the product if its price is under this minimum price that s/he knows. Finally, the profit obtained by the product will be the difference between the selling price minus the minimum price: $Profit(\pi) = \pi - \pi_{min}$.

Obviously, the goal of the seller is to sell the product at the maximum possible price (denoted by π_{max}) which is the value such that the following equalities hold:

$$f(x_1, \ldots, x_{i-1}, \pi_{max}, x_{i+1}, \ldots, x_m) = POS$$
$$f(x_1, \ldots, x_{i-1}, \pi_{max} + \epsilon, x_{i+1}, \ldots, x_m) = NEG, \forall \epsilon > 0$$

In other words, the use for the model is: "Which is the maximum price that I can sell this product to this customer?". Logically, the higher the price the lower the probability, so the goal is more precisely to maximise the *expected profit*, which is defined as follows:

$$E\_Profit(\pi) = \hat{p}(POS|\langle x_1, \ldots, x_{i-1}, \pi, x_{i+1}, \ldots, x_m\rangle) \cdot Profit(\pi) \tag{1}$$

where $\hat{p}$ is the estimated probability given by the negotiable feature model.

To ease notation we will denote $\hat{p}(POS|\langle x_1, \ldots, x_{i-1}, \pi, x_{i+1}, \ldots, x_m\rangle)$ as $\hat{p}(POS|\pi)$, consequently, we can express (1) as:

$$E\_Profit(\pi) = \hat{p}(POS|\pi) \cdot Profit(\pi) \tag{2}$$

with the additional constraint, as mentioned, that $\pi \geq \pi_{min}$.

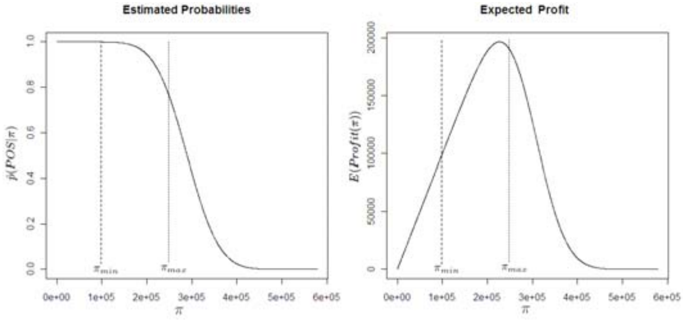


Fig. 2. Left: Example of estimated probabilities. Right: Associated expected profit. The minimum and maximum price are also shown.

So, if we have a model which estimates probabilities for the positive class, we can use formula (2) to choose the price that has to be offered to the customer. If probabilities are well estimated, for all the range of possible prices, this must be the optimal strategy. In Figure 2 an example of the plots that are obtained for the estimated probabilities and expected profit is shown.

But this is the case where we have one bid (one offer). In many negotiation scenarios, we have the possibility of making several bids, as in bargaining. In this situation, it is not so direct how to use the model in order to set a sequence of bids to get the maximum overall expected profit. For instance, if we are allowed three bids, the overall expected profit of a sequence of bids is defined as:

$$E\_Profit(\langle\pi_1, \pi_2, \pi_3\rangle) = \hat{p}(POS|\pi_1) \cdot Profit(\pi_1) + (1 - \hat{p}(POS|\pi_1)) \cdot \hat{p}(POS|\pi_2) \cdot Profit(\pi_2) + (1 - \hat{p}(POS|\pi_1)) \cdot (1 - \hat{p}(POS|\pi_2)) \cdot \hat{p}(POS|\pi_3) \cdot Profit(\pi_3),$$

where $\pi_1 > \pi_2 > \pi_3 \geq \pi_{min}$.

4 Negotiation Strategies

In the scenario described in Section 3 the seller is the agent who uses the negotiable feature models to guide the negotiation, while the buyer can only make the decision of buying or not the product.

When we have one possible offer, it is not sensible to set the price at the maximum price that our model predicts it can be sold, because in many cases, because of the prediction error, it will be overestimated and we will not sell the product. On the contrary, selling at the minimum price ensures that we sell as many products as possible, but we get minimum profit as well. In Section 3 we saw that an appropriate way of doing this is by using the expected profit.

Obviously, if the maximum price for the buyer is lower than the minimum price for the seller, the product is not sold. We will exclude these cases, since any strategy is not going to work well for them and it is not going to make any difference to include them or not in terms of comparison.

When we have more than one possible offer, we start with a first offer and if the price is less or equal than a price which is accepted by the buyer, s/he will buy the product. Otherwise, the seller can still make another offer and follow the negotiation.

It is clear that there exists an optimum solution to this problem when the seller can make "infinite" offers to the buyer, but it is inefficient and unfeasible. This idea consists in beginning the negotiation with a very high offer and make offers of one euro less each time, until the (patient and not very intelligent) buyer purchases the product or until the price of the product is the minimum price. In this case the product would be sold by its maximum price, because the buyer purchases the product when the price offered was equal to the maximum price considered by the buyer.

In what follows we propose several strategies. One is the "baseline" method which is typically used in real estate agent's. For cases with one single bid, we introduce the strategy called "Maximum Expected Profit" (MEP), which is just the application of the expected profit as presented in the previous section. For cases with more bids (multi-bid) we present two strategies: "Best Local Expected Profit" (BLEP) strategy and "Maximum Global Optimisation" (MGO) strategy. Let us see all of them in detail below:

- Baseline method (1 bid or N bids). One of the simplest methods to price a product is to increase a percentage to its minimum price (or base cost). Instead of setting a fix percentage arbitrarily, we obtain the percentage (called α) such that it obtains the best result for the training set. For example if we obtain that the best α is 0.4, it is expected that the best profit will be obtained increasing in 40% the minimum price of the properties. If we have only 1 bid, we will increase the minimum price of the flat by α. But, if we have N bids, we will have one half of the bids with a value of α less than the calculated α and the other half of the bids with a value of α greater than the calculated α. In particular, the value of α will increase or decrease by $\alpha/(N+1)$ in each bid. For example, for 3 bids and the previous sample the three values of α for three bids would be 50%, 40% and 30%. Therefore, the first offer would be an increase of 50% over the minimum price of the product, the second an increase of 40% and the third an increase of 30%.

- Maximum Expected Profit (MEP) strategy (1 bid). This strategy is typically used in marketing when the seller can only make one offer to the customer. Each price for an instance gives a probability of buying. This strategy chooses the price that maximises the value of the expected profit. $\pi_{MEP} = argmax_{\pi}(E\_Profit(\pi))$.
- Best Local Expected Profit (BLEP) strategy (N bids). This strategy consists in applying the MEP strategy iteratively, when it is possible to make more that one offer to the buyer. The first offer is the MEP, and if the customer does not accept the offer, his/her curve of estimated probabilities is normalised taking into account the following: the probabilities of buying that are less than or equal to the probability of buying at this price will be set to 0; and the probabilities greater than the probability of buying at this price will be normalised between 0 and 1. The next offer will be calculated by applying the MEP strategy to the normalised probabilities. In the case of the probability of buying which is associated to the price is the maximum probability, it will not be set to 0, because the expected profit would always be 0. Instead of this, the next offer is directly the half of the price. The pseudo-code is in Algorithm 1.
- Maximum Global Optimisation (MGO) strategy (N bids). The objective of this strategy is to obtain the N offers that maximise the expected profit by generalising the formula that we have presented in Section 3:

  $$\pi_{MGO} = argmax_{\langle \pi_1,\ldots,\pi_N \rangle}(E\_Profit(\langle \pi_1, \ldots, \pi_N \rangle) = argmax_{\langle \pi_1,\ldots,\pi_N \rangle}(\hat{p}(POS|\pi_1) \cdot Profit(\pi_1) + (1 - \hat{p}(POS|\pi_1)) \cdot \hat{p}(POS|\pi_2) \cdot Profit(\pi_2) + \ldots + (1 - \hat{p}(POS|\pi_1)) \cdot \ldots \cdot (1 - \hat{p}(POS|\pi_{N-1})) \cdot \hat{p}(POS|\pi_N) \cdot Profit(\pi_N)).$$

  Getting the N bids from the previous formula is not direct but can be done in several ways. One option is just using a Montecarlo approach with a sufficient number of tuples to get the values for the prices that maximise the expected profit.

Algorithm 1: BLEP strategy

```
Require: $N$, $epf$ (estimated probability function or curve)
Ensure: $\pi_{BLEB}$
  $\forall x, epf(x) \leftarrow \hat{p}(POS|x)$
  $\pi_1 \leftarrow \pi_{MEB}$
  $\pi \leftarrow \pi_1$
  for $\pi_i, i \in 2..N$ do
    if $epf(\pi) \neq \max_{x \in 0..\infty}(epf(x))$ then
      $\forall x, epf(x) \leftarrow 0$
      if $epf(x) \leqslant epf(\pi)$ then
        $epf \leftarrow normalise(epf, epf(\pi), \max_{x \in 0..\infty} epf(x))$
        {$normalise(f(x), mix, max)$: returns normalised function of $f(x)$ from values $min$ and $max$ to [0..1]}
      end if
      $\pi_i \leftarrow \pi_{MEB}$
      $\pi \leftarrow \pi_i$
    else
      $\pi_i \leftarrow \pi \div 2$
      $\pi \leftarrow \pi_i$
    end if
  end for
  $\pi_{BLEB} \leftarrow \langle \pi_1, \ldots, \pi_N \rangle$
```

5 Experiments

5.1 Experimental Settings

Experiments have been performed by using real data collected from an estate agent's. We have information of 2,800 properties (flats and houses) that were sold in the last months, for which we have the following attributes ("district", "number of rooms", "square metres" and the "owner's price"). The "owner's price" is the price which the owner wants to obtain for the property.

In the experiments we have assumed that the "owner's price" is some kind of "market price" and we have considered that it is the "maximum price". Although it is not always true because in some cases the buyer could have paid more than this for the property.

We have randomly split the dataset into a training set and a test set. 10% of the data are for training and the rest to test. This tries to simulate a realistic situation when there are not too many data for training. Therefore, the results refer to 2,520 properties, and learning is made from 280 flats. We applied the solutions proposed in Section 2 to the data. In particular we have used a $J48$ decision tree[1] (with Laplace correction and without pruning) implemented in the data mining suite WEKA [3]. Since the predicted probability curve given by a classifier (such as the J48 classifier) typically shows discontinuities and strong steps when varying a negotiable feature, we have smoothed it with a low-pass filter with Bartlett overlapping window [2]. The parameter of the window has been set to the "minimum price" divided by 400. The "inversion problem" solution has been implemented with the $LinearRegression$ and $M5P$ regression techniques, also from WEKA.

These three learning techniques have been used to guide the three negotiation strategies explained in Section 4 (for the MGO strategy we used a Montecarlo approach using 1,000 random triplets) and they are compared to the two baseline methods also mentioned in Section 4. In the experiments the number of bids is either one or set to three, i.e., $N = 3$. Summing up, we have nine negotiation methods based on learning techniques and also two baseline methods (without learning process) using the best possible α (80% for one bid and the triplet $\langle 100\%, 80\%, 60\% \rangle$ for three bids).

5.2 Experimental Results

In Table 1 we can observe the results obtained for each method, in terms of number of sold properties, total sold price (in euros) and total profit (in euros).

As we see in Table 1 all the negotiation methods outperform the baseline methods. For one bid, MEP is clearly better than the baseline method. For three bids, both BLEP and MGO are much better than the baseline method. Overall, MGO makes a global optimisation and hence get better results.

[1] In this problem, we only have data of sold properties (positive class), therefore, we have generated examples of the negative class with a price higher than the "owners's price".

Table 1. Results obtained by the negotiation strategies, baseline methods and reference methods (minimum and maximum profit). Sold price and profit measured in euros.

Method	Sold flats	Sold price	Profit
All flats sold at π_{min}	2,520	356,959,593	0
All flats sold at π_{max}	2,520	712,580,216	355,620,623
1 bid			
Baseline (80%)	1,411	200,662,464	89,183,317
MEP ($J48$)	1,360	302,676,700	129,628,471
MEP ($LinearRegression$)	1,777	354,973,300	159,580,109
MEP ($M5P$)	1,783	358,504,700	161,736,313
3 bids			
Baseline (100%, 80%, 60%)	1,588	264,698,467	124,483,288
BLEP ($J48$)	1,940	382,921,400	173,381,116
BLEP ($LinearRegression$)	2,056	400,953,200	174,832,025
BLEP ($M5P$)	2,063	404,009,700	176,874,221
MGO ($J48$)	1,733	390,529,770	176,020,611
MGO ($LinearRegression$)	1,918	475,461,200	232,600,223
MGO ($M5P$)	1,906	476,171,900	234,259,509

On the other hand, the regression techniques outperform the $J48$ decision tree, so, for this problem the solution of inverting problem presentation outperforms the improved classifier solution. This is especially dramatic for MGO, which is the method that depends most on a good probability estimation. This means that the inversion problem using a normal distribution to get the estimated probabilities turns out to be a very useful approach.

6 Conclusions

This paper introduces a number of new contributions in the area of data mining and machine learning which can be useful for many application areas: retailing, control, prescription, and others where negotiation or fine-tuning can take place. Nonetheless, the implications can affect other decision-making problems where data mining models are used.

The first major contribution is the analysis of the relation between negotiable features and problem presentation, and more specifically the inversion problem which happens naturally when we have to modify or play with the negotiable attribute. We have seen that using the monotonic dependency we can change of problem presentation to do the inversion problem (such as changing a classification problem into a regression), where the original problem is now indirectly solved by estimating probabilities using a normal distribution.

The second major contribution is its application to real negotiation problems. We have developed several negotiation strategies and we have seen how they behave for one or more bids in a specific problem of property selling. We have shown that our approach highly improves the results of the classical baseline method (not using data mining) which is typical in this area. In the end, we show that the change of problem presentation (from classification into regression problem, using the negotiable feature, price, as output) gets the best results for the case with one bid but also for the case with three bids.

Since this work introduces new concepts and new ways of looking at some existing problems, many new questions and ideas appear. For instance, we would like to analyse how to tackle the problem when there are more than one negotiable feature at a time, especially for the inversion problem approach (since we would require several models, one for each negotiable feature).

Finally, more complex negotiation strategies and situations can be explored in the future: the customer can counter-offer, several customers, etc.

References

1. Devroye, L., Györfi, L., Lugosi, G.: A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability). Springer, Heidelberg (1997)
2. Weisstein, E.W.: CRC concise encyclopedia of mathematics. CRC Press, Boca Raton (2003)
3. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Elsevier, Amsterdam (2005)

Mining Concept-Drifting Data Streams Containing Labeled and Unlabeled Instances

Hanen Borchani, Pedro Larrañaga, and Concha Bielza

Departamento de Inteligencia Artificial, Facultad de Informática
Universidad Politécnica de Madrid, Boadilla del Monte, 28660, Madrid, Spain
hanen.borchani@upm.es, {pedro.larranaga,mcbielza}@fi.upm.es

Abstract. Recently, mining data streams has attracted significant attention and has been considered as a challenging task in supervised classification. Most of the existing methods dealing with this problem assume the availability of entirely labeled data streams. Unfortunately, such assumption is often violated in real-world applications given that obtaining labels is a time-consuming and expensive task, while a large amount of unlabeled instances are readily available. In this paper, we propose a new approach for handling concept-drifting data streams containing labeled and unlabeled instances. First, we use KL divergence and bootstrapping method to quantify and detect three possible kinds of drift: feature, conditional or dual. Then, if any occurs, a new classifier is learned using the EM algorithm; otherwise, the current classifier is kept unchanged. Our approach is general so that it can be applied with different classification models. Experiments performed with naive Bayes and logistic regression, on two benchmark datasets, show the good performance of our approach using only limited amounts of labeled instances.

1 Introduction

Data streams are infinite flows of highly rapid generated instances, that pose several challenges on computing systems due to limited resources of time and memory. Furthermore, as they are continuously gathered over time, data streams are characterized by their concept-drifting aspect [6,18]. That is, the learned concepts and/or the underlying data distribution are constantly evolving over time, which make the current classifier out-of-date requiring to be updated.

Recently, learning from concept-drifting data streams has become an important task for dealing with a wide range of real-world applications including network monitoring, information filtering, fraud and intrusion detection, etc. Several methods have been proposed to detect concept drift in incoming data streams and update accordingly the classifier [2,8,9,13,18,20].

However, most of these methods assume the availability of labeled data to learn accurately, and in general, they continuously monitor the classification performance over time and detect a concept drift if there is a significant fall in the performance. Unfortunately, the availability assumption of entirely labeled data streams is often violated in real-world problems as true labels may be scarce and not readily available.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 531–540, 2010.

Semi-supervised learning methods are useful in such cases since they use large amounts of unlabeled data, together with the labeled data, to learn better classifiers [22]. However, they mainly assume that data is generated according to some stationary distribution, which is not true when learning from evolving data streams, where changes may occur over time.

In this paper, we propose a new semi-supervised learning approach for concept-drifting data streams. Our aim is to take advantage of unlabeled data to detect possible concept drifts and update, if necessary, the classifier over time even if only a few labeled data is available.

To this end, inspired by a previous work of Dasu et al. [3], we use the Kullback-Leibler (KL) divergence [14] to measure distribution differences between data stream batches. Then, based on a bootstrapping method [5], we determine whether the obtained KL measures are statistically significant or not, i.e. whether a drift occurs or not. However, our approach differs from Dasu's work in three key points: First, we do not only detect whether a drift occurs or not, but we further distinguish and monitor three possible kinds of drift: feature, conditional or dual. Second, we do not assume the availability of entirely labeled data streams, but we detect possible drifts using labeled and unlabeled instances. Moreover, we propose a general approach to learn from these instances. In fact, when any of the three possible kinds of drift is detected, a new classifier is learned using the Expectation Maximization (EM) algorithm [4]. EM has been widely used in semi-supervised learning showing that it improves classification accuracy especially when having a small number of labeled data [17]. Otherwise, i.e. when no drift is detected, the current classifier is kept unchanged. Note that, our approach is general so that it can be applied with different classification learning algorithms. In this work, we consider two classifiers, namely naive Bayes and logistic regression.

The remainder of this paper is organized as follows. Section 2 defines the concept drift problem and briefly reviews related work. Section 3 introduces our new approach. Section 4 presents the experimental study. Finally, Section 5 rounds the paper off with some conclusions.

2 Concept Drift

2.1 Problem Definition

In dynamic environments, characteristic properties of data streams are often not stable but change over time. This is known as the concept drift problem [18,21].

Formally, concept drift is defined as the change in the joint probability distribution $P(\mathbf{x}, c)$ where c is the class label of a feature vector $\mathbf{x}$. $P(\mathbf{x}, c)$ is the product of the posterior distribution $P(c \mid \mathbf{x})$ and the feature distribution $P(\mathbf{x})$. According to [9], three possible sources of concept drifts can be distinguished as follows:

- Conditional change: In this case, a change in $P(c \mid \mathbf{x})$ occurs, that is, the class labels change given the feature vectors. For instance, in information

filtering domain consisting in classifying a stream of documents, denoted by $\mathbf{x}$, into relevant or non relevant, denoted by c, a conditional change may occur if the relevance of some documents changes over time, that is, their class labels change from relevant to non relevant or vice versa.
- Feature change: In this case, a change in $P(\mathbf{x})$ occurs. Intuitively, some previously infrequent feature vectors may become more frequent or vice versa. For instance, in information filtering domain, the relative frequency of some documents changes over time independently of their relevance, which may remain constant over a long period of time.
- Dual change: In this case, both changes in $P(\mathbf{x})$ and $P(c \mid \mathbf{x})$ occur. According to information filtering example, both changes in the relative frequency and the relevance of some documents are observed.

Note that, when any of these three kinds of concept drift occurs, the current classifier becomes out-of-date and a decrease in the accuracy is usually observed.

2.2 Related Work

Different methods have been proposed to handle concept drifting data streams. As pointed out in [8], these methods can be classified into *blind methods* (e.g. weighted examples [11], time window of fixed size [20,21]) that adapt the classifier at regular intervals without considering whether changes have really occurred, and *informed methods* (e.g. time window of adaptive size [21]) that only adapt the classifier whenever a change is detected.

Clearly, informed methods are more interesting since they present a more efficient way to cope with concept drifts and avoid the non controlled updating of the current classifier. The main issue is how to detect concept drifts. Most of existing works monitor, at least, one performance indicator over time [2,8,21]. The classification accuracy is the most used indicator, i.e. if there is a consistent drop in the accuracy, a drift is signaled. Other performance indicators such as the error rate, recall and precision have also been used.

An alternative approach to detect the drift is to monitor the data distribution on two different windows [3,10,19]. It is assumed that as long as the distribution of old instances is similar to the distribution of recent ones, no concept drift occurred. Otherwise, a distribution difference indicates a concept drift.

Contrary to previously proposed methods, our approach differs in the following ways: First, we do not only assert the presence or the absence of drift but we also determine efficiently which kind of drift has occurred (i.e. feature, conditional or dual) and quantify it. Second, we do not assume the availability of entirely labeled data to deal accurately with evolving data streams, but we tackle through this work a more realistic and important problem of mining data streams when only a small amount of labeled data is available.

To the best of our knowledge, only two relevant previous works have addressed the problem of scarceness of labeled instances in data streams. The first one [12] is based on transductive support vector machines (TSVMs) and it maintains two separate adaptive windows on labeled and unlabeled data in order to monitor, respectively, the probabilities $P(c \mid \mathbf{x})$ and $P(\mathbf{x})$ that may change at different rates. The second work was recently proposed by [15]. It is based on an ensemble approach where each model in the ensemble is built as micro-clusters using a semi-supervised clustering technique. To cope with stream evolution, the ensemble is refined via adding a new model and removing the worst one.

3 Mining Data Streams with Partially Labeled Data

Let D denote the data stream that arrives over time in batches. Let D^s denotes the batch at step s, which consists of the union of two disjoint subsets D_u^s and D_l^s. D_u^s denotes a set of N_u^s unlabeled instances $(\mathbf{x})$, while D_l^s denotes a set of N_l^s labeled instances $(\mathbf{x}, c)$, s.t. $\mathbf{x}$ represents an n-dimensional feature vector $(x_1, ..., x_n)$ and $c \in \Omega_C = \{c_1, c_2, ..., c_{|C|}\}$ represents the corresponding class value for labeled instances. $N^s = N_u^s + N_l^s$ denotes the total size of D^s.

3.1 Classifier Learning

Learning a classifier from data D^s corresponds to maximizing the joint log likelihood of D^s given the parameters $\boldsymbol{\theta}^s$, expressed as follows [17]:

$$\begin{aligned} LL(D^s \mid \boldsymbol{\theta}^s) = &\sum_{i=1}^{N_l^s} log(P(c_j \mid \mathbf{x}_i; \boldsymbol{\theta}^s) P(\mathbf{x}_i \mid \boldsymbol{\theta}^s)) \\ &+ \sum_{i=1}^{N_u^s} log \sum_{j=1}^{|C|} P(c_j \mid \mathbf{x}_i; \boldsymbol{\theta}^s) P(\mathbf{x}_i \mid \boldsymbol{\theta}^s) \end{aligned} \quad (1)$$

where the first term is derived from labeled instances and the second one is based on unlabeled data. Notice that this equation contains a log of sums for the unlabeled data, which makes a maximization by partial derivatives with respect to $\boldsymbol{\theta}^s$ analytically intractable.

Consider that we can have access to the class labels of all the instances, represented using a matrix of binary indicator variables $\mathbf{z}$, where rows correspond to different instances and columns to different classes, so that an entry is $z_{ij} = 1$ iff c_j is the class of instance $\mathbf{x}_i$ and $z_{ij} = 0$ otherwise. Thus, (1) can be rewritten as follows:

$$LL(D^s \mid \boldsymbol{\theta}^s; \mathbf{z}) = \sum_{i=1}^{N^s} \sum_{j=1}^{|C|} z_{ij} \; log(P(c_j \mid \mathbf{x}_i; \boldsymbol{\theta}^s) P(\mathbf{x}_i \mid \boldsymbol{\theta}^s)) \quad (2)$$

We use the EM algorithm [4] to find the maximum $\hat{\boldsymbol{\theta}}^s$ of (2). Let $\hat{\mathbf{z}}_t$ and $\hat{\boldsymbol{\theta}}_t^s$ denote the estimates for $\mathbf{z}$ and $\boldsymbol{\theta}^s$ at iteration t. EM starts with an initial estimate of classifier parameters $\hat{\boldsymbol{\theta}}_1^s$ from only the labeled data D_l^s. Then, it iterates over the E- and M-steps:

- The E-step: uses the current classifier parameters to probabilistically label the unlabeled instances in D^s_u. Formally, it computes the expected value of:

$$\hat{\mathbf{z}}_{t+1} = E[\mathbf{z} \mid D^s; \hat{\boldsymbol{\theta}}^s_t] \tag{3}$$

  Clearly, for labeled data z_{ij} is easily determined since classes are already known. For unlabeled data, z_{ij} should be estimated as follows:

$$E[z_{ij} \mid D^s; \hat{\boldsymbol{\theta}}^s_t] = \begin{cases} 1 & \text{if } c_j = argmax_c P(c \mid \mathbf{x}_i; \hat{\boldsymbol{\theta}}^s_t) \\ 0 & \text{otherwise} \end{cases}$$

- The M-step: re-estimates the classifier for the whole data D^s, i.e. using all instances, the originally labeled and the newly labeled by the E-step. In fact, this step consists in computing new parameters $\hat{\boldsymbol{\theta}}^s_{t+1}$ using the current expected value of $\hat{\mathbf{z}}$. Formally, we have:

$$\hat{\boldsymbol{\theta}}^s_{t+1} = argmax_{\boldsymbol{\theta}^s} LL(D^s \mid \boldsymbol{\theta}^s; \hat{\mathbf{z}}_{t+1}) \tag{4}$$

These two steps are iterated until reaching convergence as proved by Dempster et al. [4]. At convergence, EM finds $\hat{\boldsymbol{\theta}}^s$ that locally maximizes the log likelihood with respect to both labeled and unlabeled data.

We consider in this work two different classifiers:

- Naive Bayes (NB) [16]: For NB, parameters $\boldsymbol{\theta}^s$ denote the probability tables of class C and of each feature variable given C.
- Logistic regression (LR) [7]: For LR, the conditional log likelihood is maximized instead of the log likelihood. Hence, the expression $P(\mathbf{x}_i \mid \boldsymbol{\theta}^s)$ is removed from both equations (1) and (2). Moreover, parameters $\boldsymbol{\theta}^s$ are represented by the vector $(\boldsymbol{\theta}^s_{j0}, \boldsymbol{\theta}^s_{j1}, .., \boldsymbol{\theta}^s_{jn})^T$ for $j = 1, ..., |C|$.

3.2 Detecting Concept Drift

Given a new batch of data D^{s+1}, the objective is to detect changes whenever they occur and adapt if necessary the current classifier. It is assumed that as long as the joint probability distribution of D^{s+1} is similar to the distribution of D^s, no concept drift occurs. Otherwise, a concept drift should be indicated.

Thus, in order to detect possible changes, we use the KL divergence [14] to measure differences between the empirical distributions of D^{s+1} and D^s. Note that the KL divergence has two fundamental properties, namely, the non-negativity, being 0 iff the two compared distributions are the same, and its asymmetry. Moreover, a higher KL value indicates a higher dissimilarity between distributions, and so, a pronounced drift.

First, using only the labeled part of D^{s+1} and D^s, we proceed to measure the KL divergence between the conditional distributions of the class given the features, in order to monitor the conditional change:

$$kl_{cc} = KL(\hat{P}_{D_l^{s+1}}(C \mid \mathbf{x}) || \hat{P}_{D_l^s}(C \mid \mathbf{x})) = \sum_{j=1}^{|C|} \hat{P}_{D_l^{s+1}}(c_j \mid \mathbf{x}) log_2 \frac{\hat{P}_{D_l^{s+1}}(c_j \mid \mathbf{x})}{\hat{P}_{D_l^s}(c_j \mid \mathbf{x})} \tag{5}$$

In addition, using all the data (labeled and unlabeled), except the class variable, we measure the KL divergence between the feature distributions, to monitor the feature change:

$$kl_{fc} = KL(\hat{P}_{D^{s+1}}(\mathbf{x})||\hat{P}_{D^s}(\mathbf{x})) = \sum_{\mathbf{x}} \hat{P}_{D^{s+1}}(\mathbf{x}) log_2 \frac{\hat{P}_{D^{s+1}}(\mathbf{x})}{\hat{P}_{D^s}(\mathbf{x})} \tag{6}$$

In order to determine whether the computed KL measures are statistically significant or not, we use the bootstrapping method [5] following the previous work in [3]. Specifically, to decide whether the obtained kl_{cc} value is significant or not, we consider the null hypothesis: $H_{0_{cc}} : P_{D_l^{s+1}}(C \mid \mathbf{x}) = P_{D_l^s}(C \mid \mathbf{x})$ denoting that no conditional change has occurred. So, our objective is to determine the probability of observing the value kl_{cc} if $H_{0_{cc}}$ is true.

To this end, given the empirical distribution $\hat{P}_{D_l^s}(C \mid \mathbf{x})$, we sample k data sets denoted S_b, $b = 1, ..., k$, each of size $2N_l^s$. Then, we consider the first N_l^s elements S_{b1} as coming from the distribution $\hat{P}_{D_l^s}(C \mid \mathbf{x})$, and the remaining N_l^s elements $S_{b2} = S_b \setminus S_{b1}$ as coming from the other distribution $\hat{P}_{D_l^{s+1}}(C \mid \mathbf{x})$; and we compute bootstrap estimates $\hat{kl}_{ccb} = KL(S_{b2}||S_{b1})$ between the two samples S_{b2} and S_{b1}. These estimates form an empirical distribution from which we construct a critical region $[\hat{kl}_{cc}^{\alpha}, \infty)$, where $\hat{kl}_{cc}^{\alpha}$ represents the $(1 - \alpha)$-percentile of the bootstrap estimates, and α is a desired significance level.

Finally, if we observe that kl_{cc} falls into the critical region, i.e. $kl_{cc} > \hat{kl}_{cc}^{\alpha}$, we conclude that it is statistically significant and invalidates $H_{0_{cc}}$. In other words, we conclude that a conditional change is detected.

Similarly, in order to decide whether the obtained kl_{fc} value is significant or not, we consider the null hypothesis: $H_{0_{fc}} : P_{D^{s+1}}(\mathbf{x}) = P_{D^s}(\mathbf{x})$ and apply the same process to determine the critical region $[\hat{kl}_{fc}^{\alpha}; \infty)$ and decide about a feature change. Note that, in case that either a feature or conditional change is detected, we proceed to learn a new classifier. Otherwise, we keep the current classifier unchanged. Algorithm 1 outlines the whole proposed approach.

4 Experimental Study

For experiments, we consider the benchmark synthetic data set on a rotating hyperplane, widely used to simulate the concept drift problem [6,9,18,20]. A synthetic data allows us to carry out experiments with different types of drift as well as different percentages of labeled data and, hence, to investigate the performance of our approach under controlled situations.

A hyperplane in d-dimensional data is denoted by $\sum_{i=1}^{n} w_i x_i = w_0$ where $\mathbf{w} = (w_1, ..., w_d)^T$ is the weight vector. Instances for which $\sum_{i=1}^{n} w_i x_i \geq w_0$ are labeled positive, and those for which $\sum_{i=1}^{n} w_i x_i < w_0$ are labeled negative. Weights w_i are initialized by random values in the range of $[0, 1]$, and w_0 values are determined so that $w_0 = \frac{1}{2} \sum_{i=1}^{n} w_i$. We generated x_i from a Gaussian distribution with mean μ_i and variance σ_i^2. The feature change is simulated

Algorithm 1

```
Input : $D^s, \boldsymbol{\theta}^s, D^{s+1}, k, \alpha$
Output : $\boldsymbol{\theta}^{s+1}$
Compute $kl_{cc}$
Compute the bootstrap estimates $\hat{kl}_{ccb}$, $b = 1, ..., k$, and critical region $[\hat{kl}^{\alpha}_{cc}, \infty)$
Compute $kl_{fc}$
Compute the bootstrap estimates $\hat{kl}_{fcb}$, $b = 1, ..., k$, and critical region $[\hat{kl}^{\alpha}_{fc}, \infty)$
if $kl_{cc} > \hat{kl}^{\alpha}_{cc}$ or $kl_{fc} > \hat{kl}^{\alpha}_{fc}$ then
    A change is detected, learn a new classifier from $D^{s+1}$
    $\hat{\boldsymbol{\theta}}^{s+1}_1 \leftarrow$ initial parameters induced only from labeled data $D^{s+1}_l$
    while not convergence do
        E-step: compute the expected labels for all unlabeled instances using (3)
        M-step: update classifier parameters using (4) obtaining $\hat{\boldsymbol{\theta}}^{s+1}$
    end while
    $\boldsymbol{\theta}^{s+1} \longleftarrow \hat{\boldsymbol{\theta}}^{s+1}$
else
    No change is detected: $\boldsymbol{\theta}^{s+1} \longleftarrow \boldsymbol{\theta}^s$
end if
return $\boldsymbol{\theta}^{s+1}$
```

through the change of the mean, i.e. μ_i is changed to $\mu_i s_i(1+t)$, and the conditional change is simulated by the change of weights w_i to $w_i s_i(1+t)$. Parameter $t \in [0,1]$ represents the magnitude of changes, and parameter $s_i \in \{-1,1\}$ specifies the direction of changes which could be reversed with a probability of 0.1. We generated a data stream of 10 dimensions ($n = 10$) with 80,000 instances, each block of 20,000 using a different magnitude of change t which is respectively set to 0.1, 0.2, 0.5, 1. We consider equal training and testing subsets of size 1000, such that every training set is followed by a testing set. Also, we use $k = 500$ and $\alpha = 0.05$ as bootstrap parameter values.

Table 1 represents the obtained results for the drift detection proposal. The first column represents the block numbers. Then, in columns 2 and 3, the kl_{fc} and average $\hat{kl}^{\alpha}_{fc}$ values, which are the same for all experiments with different percentages of labeled data, since they only use the features values. Finally, in columns 4 to 9, kl_{cc} and $\hat{kl}^{\alpha}_{cc}$ values respectively for 2%, 5% and 10% of labeled data. Due to lack of space, details for kl values computed over sequential blocks having the same magnitude of change t (e.g. from block 1 to block 10) are omitted. Instead, the average of these kl values are included. Note that, neither feature changes nor conditional changes are detected between these blocks.

As expected, a feature change is only detected between blocks 10 and 11 where the magnitude of change t goes from 0.1 to 0.2, blocks 20 and 21 where t goes from 0.2 to 0.5, and blocks 30 and 31 where the t goes from 0.5 to 1. The larger the modification of t values, the higher are the kl_{fc} values, showing a more significant drift in the feature distributions between considered data blocks.

The same is observed for the conditional distributions monitored via kl_{cc} values, for both 5% and 10% of labeled data. However, in the case of 2% of labeled data conditional changes are not detected. This can be explained by the

Table 1. Drift detection results for rotating hyperplane dataset

			2% labeled		5% labeled		10% labeled	
blocks	kl_{fc}	$\hat{kl}^{\alpha}_{fc}$	kl_{cc}	$\hat{kl}^{\alpha}_{cc}$	kl_{cc}	$\hat{kl}^{\alpha}_{cc}$	kl_{cc}	$\hat{kl}^{\alpha}_{cc}$
1 – 10	0.1132	0.1431	0.7193	4.9660	0.4007	1.2400	0.2087	0.5845
10 – 11	**0.1434**	**0.1405**	3.2875	6.4493	**1.3862**	**1.2369**	**0.9812**	**0.6499**
11 – 20	0.1176	0.1449	1.6984	6.5568	0.8258	1.6646	0.1873	0.5800
20 – 21	**0.1544**	**0.1408**	4.1283	6.4512	**1.3821**	**1.2364**	**1.0118**	**0.8162**
21 – 30	0.1063	0.1413	1.2063	6.4298	0.4126	1.3954	0.1544	0.3882
30 – 31	**0.1767**	**0.1466**	4.7233	6.4346	**1.9310**	**1.3660**	**1.3369**	**0.7648**
31 – 40	0.1164	0.1429	0.8804	5.7063	0.4345	1.0492	0.1962	0.5035

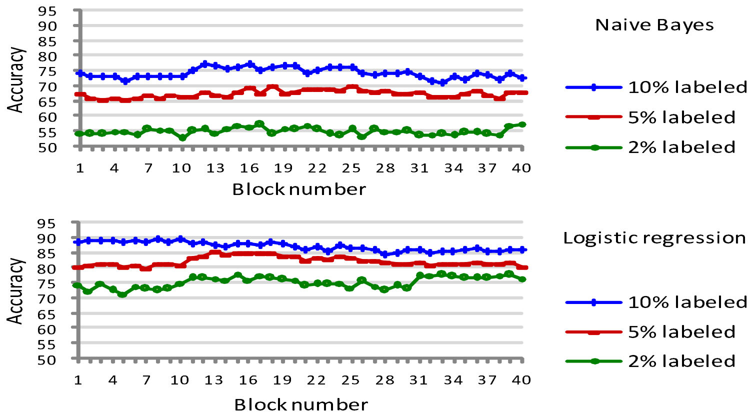


Fig. 1. Classification results of NB and LR for rotating hyperplane dataset

fact that the true conditional distribution can not be accurately approximated with very few labeled instances.

Furthermore, Figure 1 presents accuracy curves for NB and LR. Obviously, the performance of both NB and LR is much better when higher percentages of labeled data are considered. Note also that LR always outperforms NB for all percentages of labeled data.

As additional experiments, we consider the real mushroom dataset from the UCI repository [1] regarded as having virtual concept drift (i.e. feature changes) but no real concept drift (i.e. conditional changes) [13]. Mushroom dataset contains 22 variables and 8124 instances. We split it into 6 blocks, from each block 1000 instances are used for training and 354 instances for testing. As before, we use $k = 500$ and $\alpha = 0.05$ as bootstrap parameter values, and we consider three percentages of randomly selected labeled data: 2%, 5% and 10%.

According to results in Table 2, only feature changes are detected between blocks 1 and 2, and blocks 4 and 5. However, no conditional changes are detected for all percentages of labeled data, which proves that our detection method is resilient to false alarms. Moreover, according to Figure 2, using more labeled

Table 2. Drift detection results for mushroom dataset

			2% labeled		5% labeled		10% labeled	
blocks	kl_{fc}	$\hat{kl}^{\alpha}_{fc}$	kl_{cc}	$\hat{kl}^{\alpha}_{cc}$	kl_{cc}	$\hat{kl}^{\alpha}_{cc}$	kl_{cc}	$\hat{kl}^{\alpha}_{cc}$
$1-2$	**0.2251**	**0.0450**	0.0003	0.1430	0.0001	0.0035	0.0079	0.0176
$2-3$	0.0365	0.0755	0.0019	0.0156	0.0005	0.0030	0.0003	0.0009
$3-4$	0.1184	0.1536	0.0031	0.0049	0.0054	0.0057	0.0002	0.0181
$4-5$	**1.2973**	**0.2373**	0.0043	0.1347	0.0002	0.0063	0.0013	0.0030
$5-6$	0.0007	0.0814	0.0028	0.0105	0.0018	0.0092	0.0001	0.0053

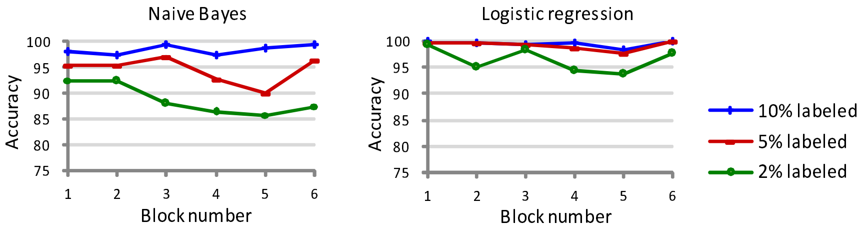


Fig. 2. Classification results of NB and LR for mushroom dataset

data improves the predictive accuracies of both NB and LR. Nevertheless, for LR, from 5% to 10% of labeled data, the improvement is very small, so that the corresponding curves are almost superimposed. Notice also that, LR always outperforms NB for mushroom dataset.

5 Conclusion

We deal with an important problem in data stream mining which has not been addressed by the most existing works assuming that data streams are entirely labeled. In our work, using both labeled and unlabeled instances, we distinguish and detect three possible changes: feature, conditional or dual, using KL divergence and bootstrapping method, then, if required, we update the classifier using EM algorithm. Experiments with naive Bayes and logistic regression show the effectiveness and the good performance of our approach. For the future, we plan to carry out additional experiments especially on more complex data streams.

Acknowledgement

This work has been supported by the Spanish Ministry of Science and Innovation under projects TIN2007-62626, TIN2008-06815-C02-02, Consolider Ingenio 2010-CSD2007-00018, and the Cajal Blue Brain project.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, University of California, Irvine (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html

2. Bifet, A., Gavalda, R.: Learning from time-changing data with adaptive windowing. In: Proc. of the 7th SIAM Int. Conf. on Data Mining, pp. 29–40 (2007)
3. Dasu, T., Krishnan, S., Venkatasubramanian, S., Yi, K.: An information-theoretic approach to detecting changes in multi-dimensional data streams. In: Proc. of the 38th Symposium on the Interface of Statistics, Computing Science, and Applications, pp. 1–24 (2006)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society 39, 1–38 (1977)
5. Efron, B., Tibshirani, R.: Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. Statistical Science 1(1), 54–75 (1986)
6. Hulten, G., Spencer, L., Domingos, P.: Mining time changing data streams. In: Proc. of the 7th Int. Conf. on Knowledge Discovery and Data mining, pp. 97–106 (2001)
7. Hosmer, D.W., Lemeshow, S.: Applied Logistic Regression, 2nd edn. John Wiley & Sons, New York (2000)
8. Gama, J., Castillo, G.: Learning with local drift detection. In: Proc. of the 2nd Int. Conf. on Advanced Data Mining and Applications, pp. 42–55 (2006)
9. Gao, J., Ding, B., Fan, W., Han, J., Yu, P.S.: Classifying data streams with skewed class distributions and concept drifts. IEEE Internet Computing 12(6), 37–49 (2008)
10. Kifer, D., Ben-David, S., Gehrke, J.: Detecting change in data streams. In: Proc. of the 30th Int. Conf. on Very Large Data Bases, vol. 30, pp. 180–191 (2004)
11. Klinkenberg, R.: Learning drifting concepts: Example selection vs. example weighting. Intelligent Data Analysis 8(3), 281–300 (2004)
12. Klinkenberg, R.: Using labeled and unlabeled data to learn drifting concepts. In: Proc. of Int. Joint Conf. on Artificial Intelligence, pp. 16–24 (2001)
13. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: An ensemble method for drifting concepts. Journal of Machine Learning Research 8, 2755–2790 (2007)
14. Kullback, S., Leibler, R.A.: On information and sufficiency. The Annals of Mathematical Statistics 22(1), 79–86 (1951)
15. Masud, M.M., Gao, J., Khan, L., Han, J., Thuraisingham, B.: A practical approach to classify evolving data streams: Training with limited amount of labeled data. In: The 8th IEEE Int. Conf. on Data Mining, pp. 929–934 (2008)
16. Minsky, M.: Steps towards artificial intelligence. Computers and Thought, 406–450 (1961)
17. Nigam, K., McCallum, A., Thrun, S., Mitchell, T.: Text classification from labeled and unlabeled documents using EM. Machine Learning 39(2/3), 103–134 (2000)
18. Tsymbal, A., Pechenizkiy, M., Cunningham, P., Puuronen, S.: Dynamic integration of classifiers for handling concept drift. Information Fusion 9(1), 56–68 (2008)
19. Vorburg, P., Bernstein, A.: Entropy-based concept shift detection. In: Proc. of the 6th Int. Conf. on Data Mining, pp. 1113–1118 (2006)
20. Wang, H., Fan, W., Yu, P., Han, J.: Mining concept drifting data streams using ensemble classifiers. In: Proc. of the 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 226–235 (2003)
21. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23(1), 69–101 (1996)
22. Zhu, X.: Semi-supervised learning literature survey. Technical Report, Computer Sciences, University of Wisconsin-Madison (2008)

Exploring the Performance of Resampling Strategies for the Class Imbalance Problem

Vicente García, José Salvador Sánchez, and Ramón A. Mollineda

Institute of New Imaging Technologies
Dept. Llenguatges i Sistemes Informàtics
Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló de la Plana (Spain)
{jimenezv,sanchez,mollined}@uji.es

Abstract. The present paper studies the influence of two distinct factors on the performance of some resampling strategies for handling imbalanced data sets. In particular, we focus on the nature of the classifier used, along with the ratio between minority and majority classes. Experiments using eight different classifiers show that the most significant differences are for data sets with low or moderate imbalance: over-sampling clearly appears as better than under-sampling for local classifiers, whereas some under-sampling strategies outperform over-sampling when employing classifiers with global learning.

1 Introduction

Class imbalance constitutes one of the problems that has recently received most attention in research areas such as Machine Learning, Pattern Recognition and Data Mining. A two-class data set is said to be imbalanced if one of the classes (the minority one) is represented by a very small number of instances in comparison to the other (the majority) class [1]. Besides, the minority class is the most important one from the point of view of the learning task. It has been observed that class imbalance may cause a significant deterioration in the performance attainable by standard learners because these are often biased towards the majority class [2]. This issue is particular important in real-world applications where it is costly to misclassify examples of the minority class, such as diagnosis of an infrequent diseases [3], detection of fraudulent telephone calls [4], detection of oil spills in radar images [5], text categorization [6], and credit assessment [7]. Because of examples of the minority and majority classes usually represent the presence and absence of rare cases respectively, they are also known as positive and negative examples.

Research on this topic can be categorized into three groups. One has primarily focused on the implementation of solutions for handling the imbalance both at the data and algorithmic levels. Another group has addressed the problem of measuring the classifier performance in imbalanced domains. The third consists of analyzing the relationship between class imbalance and other data complexity characteristics. From these three general topics in class imbalance, data level methods are the most investigated. These methods consist of balancing the original data set, either by over-sampling the

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 541–549, 2010.

minority class [8, 9, 10, 11] and/or by under-sampling the majority class [12, 13, 14], until the problem classes are approximately equally represented.

Conclusions about what is the best data level solution for the class imbalance problem are divergent. In this sense, Hulse et al. [15] suggest that the utility of the resampling methods depends on a number of factors, including the ratio between positive and negative examples, other characteristics of data, and the nature of the classifier.

In the present work, we study the influence of the imbalance ratio (i.e., ratio between minority and majority classes) and the nature of the classifier used on the effectiveness of some popular resampling techniques for handling the class imbalance problem. To this end, we will carry out experiments over real databases with two different levels of imbalance, employing eight classifiers and four performance measures.

The rest of the paper is organized as follows. Section 2 provides some performance measures especially useful for class imbalance problems. Section 3 reviews several well-known resampling strategies. Next, in Sect. 4 the experimental set-up is described. Section 5 reports the results and discusses the most important findings. Finally, Sect. 6 concludes the present study and outlines possible directions for future research.

2 Performance Evaluation in Class Imbalance Problems

Typical metrics for measuring the performance of learning systems are classification accuracy and error rates, which can be easily derived from a 2×2 confusion matrix as that given in Table 1 (for a two-class problem). These measures can be computed as $Acc = (TP+TN)/(TP+FN+TN+FP)$ and $Err = (FP+FN)/(TP+FN+TN+FP)$.

Table 1. Confusion matrix for a two-class problem

	Predicted positive	Predicted negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

However, empirical evidence shows that most of these commonly used measures are biased towards the majority class. Shortcomings of these evaluators have motivated the search for alternative measures, such as the geometric mean of class accuracies [10] and the area under the ROC curve (AUC) [16].

Given the true positive rate, $TPrate = TP/(TP + FN)$, and the true negative rate, $TNrate = TN/(TN + FP)$, the *geometric mean* of $TPrate$ and $TNrate$ is computed as

$$Gm = \sqrt{TPrate \cdot TNrate} \tag{1}$$

This measure can be seen as a sort of correlation between both rates, because a high value occurs when they both are also high, while a low value is related to at least one low rate.

More recently, Garcia et al. proposed a new measure called *index of balanced accuracy* [17], which is computed as

$$IBA = (1 + 0.1 \cdot (TPrate - TNrate)) \cdot TPrate \cdot TNrate \tag{2}$$

The IBA metric quantifies a certain trade-off between an unbiased measure of overall accuracy and an index of how balanced are the two class accuracies. Unlike most performance metrics, the IBA function does not take care of the overall accuracy only, but also intends to favor classifiers with better results on the positive class (generally, the most important class).

3 Resampling

Data level methods for balancing the classes consists of resampling the original data set, either by over-sampling the minority class or by under-sampling the majority class, until the classes are approximately equally represented. Both strategies can be applied in any learning system, since they act as a preprocessing phase, allowing the learning system to receive the training instances as if they belonged to a well-balanced data set. Thus, any bias of the system towards the majority class due to the different proportion of examples per class would be expected to be suppressed.

However, resampling methods have shown important drawbacks. Under-sampling may throw out potentially useful data, while over-sampling artificially increases the size of the data set and consequently, worsens the computational burden of the learning algorithm.

The simplest method to increase the size of the minority class corresponds to random over-sampling, that is, a non-heuristic method that balances the class distribution through the random replication of positive examples. Nevertheless, since this method replicates existing examples in the minority class, overfitting is more likely to occur. Chawla et al. [18] proposed an over-sampling technique that generates new synthetic minority instances by interpolating between several positive examples that lie close together. This method, called SMOTE, allows the classifier to build larger decision regions that contain nearby instances from the minority class. From the original SMOTE algorithm, several modifications have been proposed in the literature. For example, García et al. [19] developed three alternatives based upon the concept of surrounding neighborhood with the aim of taking into account both proximity and spatial distribution of the instances.

Random under-sampling [2, 20] aims at balancing the data set through the random removal of negative examples. Despite its simplicity, it has empirically been shown to be one of the most effective resampling methods. Unlike the random approach, many other proposals are based on a more intelligent selection of the negative examples to eliminate. For example, Kubat and Matwin [10] proposed the one-sided selection technique, which selectively removes only those negative instances that are redundant or that border the minority class examples (they assume that these bordering cases are noise). The border examples were detected using the concept of Tomek links [21]. On the other hand, Barandela et al. [8] introduced a method that eliminates noisy instances of the majority class by means of Wilson's editing [22], as well as redundant examples through the modified selective subset condensing algorithm [23].

4 Experiments

The experiments here carried out are directed to empirically evaluate several resampling strategies, pursuing to determine the influence of the imbalance ratio and the nature of classifier on the performance of over-sampling and under-sampling.

Table 2. Data sets used in the experiments

Data Set	Positive Examples	Negative Examples	Classes	Majority Class	Source
Breast	81	196	2	1	UCI[1]
Ecoli	35	301	8	1,2,3,5,6,7,8	UCI
German	300	700	2	1	UCI
Glass	17	197	9	1,2,4,5,6,7,8,9	UCI
Haberman	81	225	2	1	UCI
Laryngeal$_2$	53	639	2	1	Library[2]
Letter$_a$	789	19211	26	$2,\ldots,26$	UCI
Phoneme	1586	3818	2	1	UCI
Optidigts	554	5066	10	1,2,3,4,5,6,7,8,10	UCI
Pendigits	1055	9937	10	1,2,3,4,5,7,8,9,10	UCI
Pima	268	500	2	1	UCI
Satimage	626	5809	7	1,2,3,5,6,7	UCI
Scrapie	531	2582	2	1	Library
Segmentation	330	1980	6	1,2,3,4,6	UCI
Spambase	1813	2788	2	1	UCI
Vehicle	212	634	4	2,3,4	UCI
Yeast	429	1055	10	1,3,4,5,6,7,8,9,10	UCI

[1] UCI Machine Learning Database Repository `http://archive.ics.uci.edu/ml/`
[2] Library `http://www.vision.uji.es/~sanchez/Databases/`

Experiments were conducted as follows:

Data sets: Seventeen real data sets (summary of whom is given in Table 2) were employed in the experiment. All data sets were transformed into two-class problems by keeping one original class and joining the objects of the remaining classes. The fifth column in Table 2 indicates the original classes that have been joined to shape the majority class. For example, in Vehicle database the objects of classes 2, 3, and 4 were combined to form a unique majority class and the original class 1 was left as the minority class.

Partitions: For each database, a 10-fold cross-validation was repeated 5 times.

Resampling strategies: random under-sampling (RUS), one-sided selection (OSS), Wilson's editing over the negative examples (WE$^-$), the combination of this with the modified selective subset condensing over the negative instances (WE$^-$+MSS$^-$), SMOTE, and the Gabriel-graph-based SMOTE (gg-SMOTE) were employed.

Classifiers: the nearest neighbor ($1, 7, 13$-NN) rule, a multi-layer perceptron (MLP), a support vector classifier (SVC), the naïve Bayes classifier (NBC), a decision tree (J48), and a radial basis function network (RBF) were used, all of them taken from the Weka toolkit [24]. In order to run the NBC on the data sets here considered, the numeric attributes were modeled by a normal distribution.

Performance metrics: TPrate, TNrate, IBA and the geometric mean (Gm) were calculated to measure the classification performance.

Classifiers were applied to each original training set and also to sets that were preprocessed by the different resampling strategies. Results obtained in terms of the four performance metrics were evaluated by a paired $t-$test between each pair of methods, for each data set. Based on these values, we computed an index of performance, which is calculated as the difference between *wins* and *losses*, where *wins* is the total number of times that a method A has been significantly better than another method and *losses* is the total number of times that A has been significantly worse than another method. With the aim of summarizing the values of this index of performance, we ranked the position of the resampling algorithms: as there are 7 competing methods, the ranks were from 1 (best) to 7 (worst).

5 Results

Since we are interested in analyzing the performance of the resampling strategies with respect to the degree of imbalance, we computed an imbalance ratio as

$$IR = \frac{n^+}{n^-} \tag{3}$$

where n^+ and n^- denote the number of positive and negative examples in the data set, respectively.

From the imbalance ratio IR, we divided the data sets into two categories. A first group consists of the databases that can be deemed as strongly imbalanced ($IR \leq 0.27$): Ecoli, Glass, Laryngeal$_2$, Letter$_a$, Optdigits, Pendigits, Satimage, Scrapie, and Segmentation. In the second group, we find the databases with a low/moderate imbalance ($IR > 0.27$): Breast, German, Haberman, Phoneme, Pima, Spambase, Vehicle, and Yeast.

5.1 Results on Data Sets with Severe Imbalance

Regarding to the data sets with a severe imbalance, several interesting conclusions can be drawn from performance rankings plotted in Fig. 1. First, one can observe that all the resampling techniques improve the accuracy on the minority class (TPrate), but entailing a certain reduction of the TNrate (see Fig. 1(a) and 1(b)). However, it is worth noting that the deterioration of the TNrate produced by over-sampling is much less significant than that of under-sampling. This is due to the large number of negative examples removed by the under-sampling algorithms.

When employing the global performance metrics, that is, IBA (Fig. 1(c)) and Gm (Fig. 1(d)), the over-sampling techniques are significantly better than the under-sampling algorithms, except for the NBC and J48 classifiers.

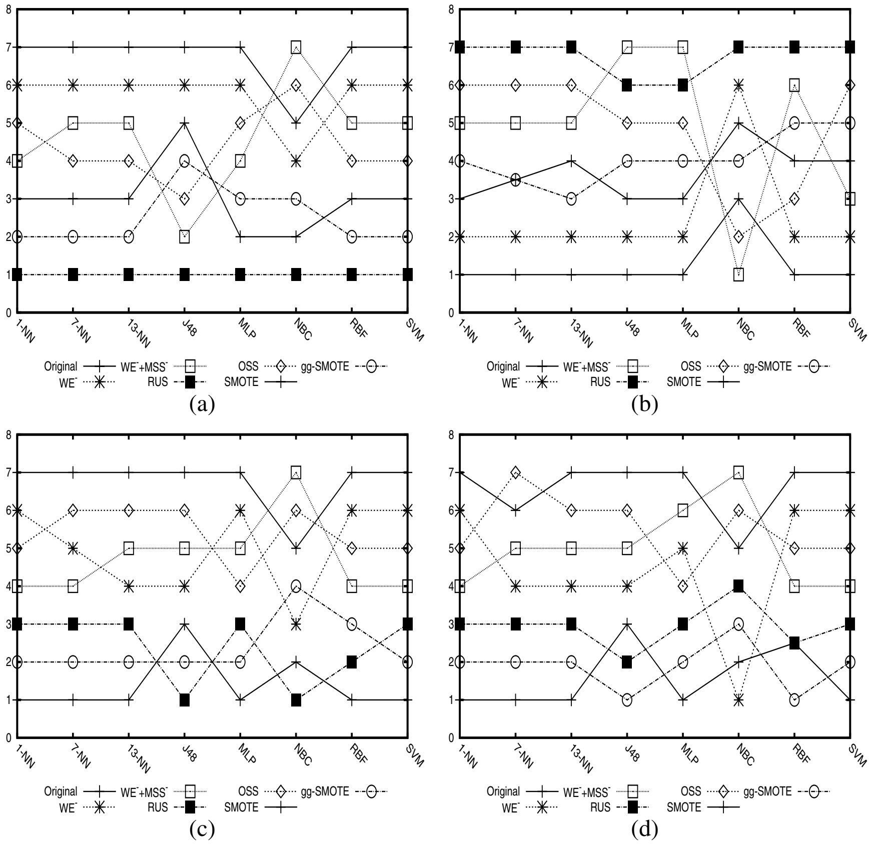
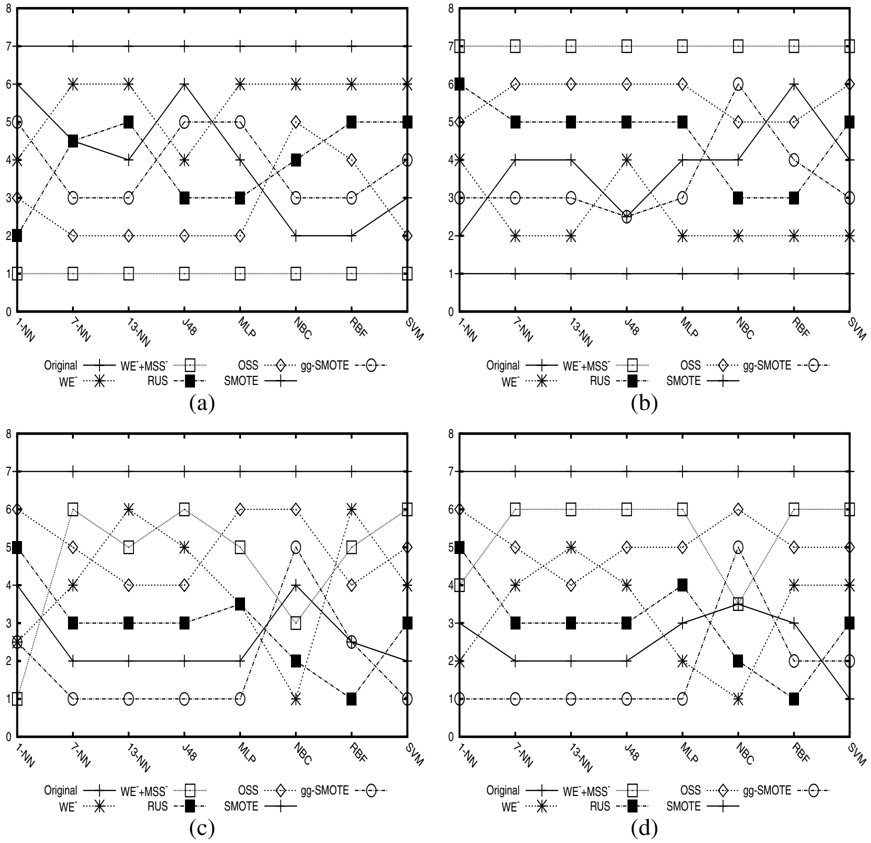


Fig. 1. Rankings for the strongly imbalanced data sets when evaluated with (a) TPrate, (b) TNrate, (c) IBA and, (d) Gm

It is also interesting to remark that the random under-sampling appears as one of the best strategies in terms of IBA and Gm. Paradoxically, the ”intelligent” under-sampling techniques generally show the worst ranks, independently of the classifier used; only the WE$^-$ approach obtains good performance when applied with the NBC classifier.

5.2 Results on Data Sets with Low/Moderate Imbalance

Fig. 2 shows the performance rankings for the data sets with a low/moderate level of imbalance. Like in the case of the strongly imbalanced databases, all the resampling algorithms increase the TPrate (Fig. 2(a)) whereas produce a certain decrease in TNrate (Fig. 2(b)). Once again, this effect is much more obvious for the under-sampling techniques. In fact, the most evident case corresponds to the WE$^-$+MSS$^-$ strategy, which obtains the best rank for the TPrate and the worst one for the TNrate. A more detailed analysis of the results produced by this technique reveals that the classes have



(a)

(b)

(c)

(d)

Fig. 2. Rankings for data sets with low/moderate imbalance when evaluated with (a) TPrate, (b) TNrate, (c) IBA and, (d) Gm

interchanged their respective ”roles”, thus the majority class has now become the minority one and vice versa.

When focused on IBA (Fig. 2(c)) and Gm (Fig. 2(d)) measures, one can observe two different situations depending on the nature of the classifier. For local classifiers such as k-NN, over-sampling is consistently better than under-sampling; in this case, gg-SMOTE stands out from the remaining strategies. Nevertheless, for classifiers with global learning, it seems that random and WE$^-$ under-sampling algorithms obtain good performance.

6 Conclusions and Further Extensions

In this paper, we have studied the effect of the classifier used and the degree of imbalance on the performance of different resampling techniques. The analysis has been based upon a number of experiments over 17 real databases with different levels of imbalance, using 8 distinct classifiers.

Experiments suggest that in fact these two factors have strong influence on the effectiveness of the resampling strategies. More specifically, the most significant differences are for data sets with low or moderate imbalance: over-sampling clearly appears as better than under-sampling for local classifiers, whereas some under-sampling strategies outperform over-sampling when employing classifiers with global learning.

The present work has revealed some interesting research avenues with regards to the resampling strategies for imbalanced data sets, such as: (i) The analysis of the data sets by means of data complexity (or problem difficulty) measures, thus obtaining a better description of data and allowing a more accurate application of specific techniques to tackle the class imbalance problem; (ii) The use of a larger number of resampling strategies to draw more exhaustive, precise conclusions; (iii) The application of several resampling algorithms to real-world imbalanced problems; and (iv) To extend this study to cost-sensitive learning.

Acknowledgment

This work has partially been supported by the Spanish Ministry of Education and Science under grants CSD2007–00018, AYA2008–05965 and TIN2009–14205, and by Fundació Caixa Castelló–Bancaixa under grant P1–1B2009–04.

References

1. He, H., Garcia, E.: Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21(9), 1263–1284 (2009)
2. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis 6(5), 429–449 (2002)
3. Cohen, G., Hilario, M., Sax, H., Hugonnet, S., Geissbuhler, A.: Learning from imbalanced data in surveillance of nosocomial infection. Artificial Intelligence in Medicine 37(1), 7–18 (2006)
4. Fawcett, T., Provost, F.: Adaptive fraud detection. Data Minining and Knowledge Discovery 1(3), 291–316 (1997)
5. Kubat, M., Holte, R., Matwin, S.: Machine learning for the detection of oil spills in satellite radar images. Machine learning 30(2-3), 195–215 (1998)
6. Tan, S.: Neighbor-weighted K-nearest neighbor for unbalanced text corpus. Expert Systems with Applications 28(4), 667–671 (2005)
7. Huang, Y.M., Hung, C.M., Jiau, H.: Evaluation of neural networks and data mining methods on a credit assesssment task for class imbalance problem. Nonlinear Analysis: Real World Applications 7(4), 720–757 (2006)
8. Barandela, R., Sánchez, J., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition 36(3), 849–851 (2003)
9. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005)
10. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: One-sided selection. In: Proceedings of The 14th International Conference on Machine Learning, pp. 179–186 (1997)

11. Yen, S.J., Lee, Y.S., Lin, C.H., Ying, J.C.: Investigating the effect of sampling methods for imbalanced data distributions. In: Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, October 2006, vol. 5, pp. 4163–4168 (2006)
12. Batista, G., Prati, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explorations Newsletter 6(1), 20–29 (2004)
13. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evolutionary Computation 17(3), 275–306 (2009)
14. He, G., Han, H., Wang, W.: An over-sampling expert system for learning from imbalanced data sets. In: International Conference on Neural Networks and Brain, vol. 1, pp. 537–541 (2005)
15. Hulse, J., Khoshgoftaar, T., Napolitano, A.: Experimental perspectives on learning from imbalanced data. In: Proceedings of the 24th International Conference on Machine learning, pp. 935–942 (2007)
16. Daskalaki, S., Kopanas, I., Avouris, N.: Evaluation of classifiers for an uneven class distribution problem. Applied Artificial Intelligence 20(5), 381–417 (2006)
17. García, V., Mollineda, R.A., Sánchez, J.S.: Index of balanced accuracy: A performance measure for skewed class distributions. In: Araujo, H., Mendonça, A.M., Pinho, A.J., Torres, M.I. (eds.) IbPRIA 2009. LNCS, vol. 5524, pp. 441–448. Springer, Heidelberg (2009)
18. Chawla, N., Bowyer, K., Hall, L., Kegelmeyer, W.: SMOTE: Synthetic minority oversampling technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
19. García, V., Sánchez, J.S., Mollineda, R.A.: On the use of surrounding neighbors for synthetic over-sampling of the minority class. In: 8th WSEAS International Conference on Simulation, Modelling and Optimization, Santander (Spain), September 2008, pp. 389–394. WSEAS Press (2008)
20. Zhang, J., Mani, I.: kNN approach to unbalanced data distributions: a case study involving information extraction. In: Proceedings Workshop on Learning from Imbalanced Datasets (2003)
21. Tomek, I.: Two modifications of cnn. IEEE Transactions on Systems, Man and Cybernetics 6(11), 769–772 (1976)
22. Wilson, D.: Asymptotic properties of nearest neighbour rules using edited data. IEEE Transactions on Systems, Man and Cybernetics 2, 408–421 (1972)
23. Barandela, R., Ferri, F., Sánchez, J.: Decision boundary preserving prototype selection for nearest neighbor classification. International Journal of Pattern Recognition and Artificial Intelligence 19(6), 787–806 (2005)
24. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers Inc., San Francisco (2005)

ITSA*: An Effective Iterative Method for Short-Text Clustering Tasks

Marcelo Errecalde[1], Diego Ingaramo[1], and Paolo Rosso[2]

[1] LIDIC, Universidad Nacional de San Luis, Argentina
[2] Natural Language Eng. Lab. ELiRF, DSIC, Universidad Politécnica de Valencia, Spain
{merreca,daingara}@unsl.edu.ar, prosso@dsic.upv.es

Abstract. The current tendency for people to use very *short documents*, e.g. blogs, text-messaging, news and others, has produced an increasing interest in automatic processing techniques which are able to deal with documents with these characteristics. In this context, "short-text clustering" is a very important research field where new clustering algorithms have been recently proposed to deal with this difficult problem. In this work, ITSA*, an iterative method based on the bio-inspired method PAntSA* is proposed for this task. ITSA* takes as input the results obtained by arbitrary clustering algorithms and refines them by iteratively using the PAntSA* algorithm. The proposal shows an interesting improvement in the results obtained with different algorithms on several short-text collections. However, ITSA* can not only be used as an effective improvement method. Using random initial clusterings, ITSA* outperforms well-known clustering algorithms in most of the experimental instances.

1 Introduction

Nowadays, the huge amount of information available on the Web offers an unlimited number of opportunities to use this information in different real life problems. Unfortunately, the automatic analysis tools that are required to make this information useful for the human comprehension, such as clustering, categorization and information extraction systems, have to face many difficulties related to the features of the documents to be processed. For example, most of Web documents like blogs, snippets, chats, FAQs, on-line evaluations of commercial products, e-mails, news, scientific abstracts and others are "*short texts*". This is a central aspect if we consider the well-known problems that short documents usually pose to different natural language processing tasks [1,2].

During the last years, different works have recognized the importance (and complexity) of dealing with short documents, and some interesting results have been reported in *short-document clustering* tasks [1,2,3,4,5,6]. In particular, an approach that has shown an excellent performance is PAntSA* [7]. PAntSA* takes the clusterings generated by arbitrary clustering algorithms and attempts to improve them by using techniques based on the *Silhouette Coefficient* and the idea of *attraction* of a group.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 550–559, 2010.

Despite the significant improvements that PAntSA* has achieved on the results obtained with different algorithms and collections, some aspects of this algorithm deserve a deeper analysis. For example, it is not clear if PAntSA* would be benefitted with an iterative approach where PAntSA* is provided with a clustering generated by the own PAntSA* algorithm in the previous cycle. On the other hand, it would be interesting to analyze the performance that this iterative version of PAntSA* can achieve when the use of an additional clustering algorithm is avoided and it only takes as input randomly generated clusterings.

Our work focusses on the previous aspects by defining in the first place, an iterative version of PAntSA* that we called ITSA*. Then, an exhaustive experimental study is carried out where the performance of ITSA* as a general improvement method is compared with the performance of the original (non-iterative) PAntSA*. Finally, the performance of ITSA* as a complete clustering algorithm is evaluated, by taking as initial groupings in this case, randomly generated clusterings.

The remainder of the paper is organized as follows. Section 2 describes the main ideas of ITSA*. The experimental setup and the analysis of the results is provided in Section 3. Finally, in Section 4 some general conclusions are drawn and possible future work is discussed.

2 The ITSA* Algorithm

The *ITerative PAntSA* (ITSA*)* algorithm, is the iterative version of *PAntSA**, a bio-inspired method intended to improve the results obtained with arbitrary document clustering algorithms. PAntSA* is the partitional version of the AntTree algorithm [8] but it also incorporates information about the *Silhouette Coefficient* [9] and the concept of *attraction* of a cluster. In PAntSA*, data (in this case *documents*) are represented by *ants* which move on a tree structure according to their similarity to the other ants already connected to the tree. Each node in the tree structure represents a single ant and each ant represents a single datum.

The whole collection of ants is initially represented by a $\mathcal{L}$ list of ants waiting to be connected. Starting from an artificial support a_0, all the ants will be incrementally connected either to that support or to other already connected ants. This process continues until all ants are connected to the structure.

Two main components of PAntSA* are: 1) the initial arrangement of the ants in the $\mathcal{L}$ list, and 2) the criterium used by an arbitrary ant a_i on the support to decide which connected ant a_+ should move toward. For the first step, PAntSA* takes as input the clustering obtained with some arbitrary clustering algorithm and uses the Silhouette Coefficient (SC) information of this grouping to determine the initial order of ants in $\mathcal{L}$. For the second process, PAntSA* uses a more informative criterium based on the concept of *attraction*. Here, if $\mathcal{G}_{a^+} = \{a^+\} \cup \mathcal{A}_{a^+}$ is the group formed by an ant a^+ connected to the support and its descendants, this relationship between the group $\mathcal{G}_{a^+}$ and the ant a_i will be referred as the *attraction of* $\mathcal{G}_{a^+}$ *on* a_i and will be denoted as $att(a_i, \mathcal{G}_{a^+})$.

PAntSA* differs from AntTree in another central aspect: it does not build hierarchical structures which have roots (ants) directly connected to the support. In PAntSA*, each ant a_j connected to the support (a_0) and its descendants (the $\mathcal{G}_{a_j}$ group) is considered as a simple set. In that way, when an arbitrary ant a_i has to be incorporated to the group of the ant a^+ that more attraction exerts on a_i, this step is implemented by simply adding a_i to the $\mathcal{G}_{a^+}$ set. The resulting PAntSA* algorithm is given in Figure 1, where it is possible to observe that it takes an arbitrary clustering as input and carries out the following three steps, in order to obtain the new clustering: 1) Connection to the support, 2) Generation of the $\mathcal{L}$ list and 3) Cluster the ants in $\mathcal{L}$.

In the first step, the most representative ant of each group of the clustering received as input is connected to the support a_0. This task involves to select the ant a_i with the highest SC value of each group C_i, and to connect each one of them to the support by generating a singleton set $\mathcal{G}_{a_i}$. The second step consists in generating the $\mathcal{L}$ list with the ants not connected in the previous step. This process also considers the SC-based ordering obtained in the previous step, and merges the remaining (ordered) ants of each group by iteratively taking the first ant of each non-empty queue, until all queues are empty. In the third step, the order in which these ants will be processed is determined by their positions in the $\mathcal{L}$ list. The clustering process of each arbitrary ant a_i simply determines the connected ant a^+ which exerts more attraction on a_i[1] and then includes a_i in the a^+' group ($\mathcal{G}_{a^+}$). The algorithm finally returns a clustering formed by the groups of the ants connected to the support.

Once the PAntSA* is implemented, its iterative version ITSA* can be easily obtained. ITSA* will have to provide an initial clustering to PAntSA*, take the output of PAntSA* as the new PAntSA*'s input for the next iteration, and repeat this process until no change is observed in the clustering generated by PAntSA* with respect to the previous iteration.

3 Experimental Setting and Analysis of Results

For the experimental work, seven collections with different levels of complexity with respect to the size, length of documents and vocabulary overlapping were selected: CICling-2002, EasyAbstracts, Micro4News, SEPLN-CICLing, R4, R8+ and R8-. The last three corpora are subsets of the well known R8-Test corpus, a subcollection of the Reuters-21578 dataset[2].

The documents were represented with the standard (normalized) *tf-idf* codification after a *stop-word* removing process. The popular *cosine measure* was used to estimate the similarity between two documents. The parameter settings for CLUDIPSO and the remainder algorithms used in the comparisons correspond to the parameters empirically derived in [5].

[1] We currently use the average similarity between a_i and all the ants in $\mathcal{G}_{a^+}$ as attraction measure but other alternatives would be also valid.

[2] Space limitations prevent us from giving a description of these corpora but it is possible to obtain in [10,11,12,3,2,4,5] more information about the features of these corpora and some links to access them for research proposes.

```
function PAntSA$^\star$($\mathcal{C}$) returns a clustering $\mathcal{C}^\star$
    input: $\mathcal{C} = \{C_1, \ldots, C_k\}$, an initial grouping
  1. Connection to the support
      1.a. Create a set $\mathcal{Q} = \{q_1, \ldots, q_k\}$ of $k$ data queues (one queue for each
           group $C_j \in \mathcal{C}$).
      1.b. Sort each queue $q_j \in \mathcal{Q}$ in decreasing order according to the Silhouette
           Coefficient of its elements. Let $\mathcal{Q}' = \{q'_1, \ldots, q'_k\}$ be the resulting set of
           ordered queues.
      1.c. Let $\mathcal{G}_\mathcal{F} = \{a_1, \ldots, a_k\}$ be the set formed by the first ant $a_i$ of each
           queue $q'_i \in \mathcal{Q}'$. For each ant $a_i \in \mathcal{G}_\mathcal{F}$, remove $a_i$ from $q'_i$ and set
           $\mathcal{G}_{a_i} = \{a_i\}$ (connect $a_i$ to the support $a_0$).
   2. Generation of the $\mathcal{L}$ list
      2.a. Let $\mathcal{Q}'' = \{q''_1, \ldots, q''_k\}$ the set of queues resulting from the previous
           process of removing the first ant of each queue in $\mathcal{Q}'$.
           Generate the $\mathcal{L}$ list by merging the queues in $\mathcal{Q}''$.
  3. Clustering process
      3.a. Repeat
               3.a.1 Select the first ant $a_i$ from the list $\mathcal{L}$.
               3.a.2 Let $a^+$ the ant with the highest $att(a_i, \mathcal{G}_{a^+})$ value.
                         $\mathcal{G}_{a^+} \leftarrow \mathcal{G}_{a^+} \cup \{a_i\}$
           Until $\mathcal{L}$ is empty
  return $\mathcal{C}^\star = \{\mathcal{G}_{a_1}, \ldots, \mathcal{G}_{a_k}\}$
```

Fig. 1. The PAntSA* algorithm

3.1 Experimental Results

The results of ITSA* were compared with the results of PAntSA* and other four clustering algorithms: K-means, K-MajorClust [13], CHAMELEON [14] and CLUDIPSO [4,5]. All these algorithms have been used in similar studies and in particular, CLUDIPSO has obtained in previous works [4,5] the best results in experiments with the four small size short-text collections presented in Section 3 (CICling-2002, EasyAbstracts, Micro4News and SEPLN-CICLing).

The quality of the results was evaluated by using the classical (external) F-measure on the clusterings that each algorithm generated in 50 independent runs per collection. The reported results correspond to the minimum (F_{min}), maximum (F_{max}) and average (F_{avg}) F-measure values. The values highlighted in bold in the different tables indicate the best obtained results.

Tables 1 and 2 show the F_{min}, F_{max} and F_{avg} values that K-means, K-MajorClust, CHAMELEON and CLUDIPSO obtained with the seven collections. These tables also include the results obtained with PAntSA* and ITSA* taking as input the groupings generated by these algorithms. They will be denoted with a "$\star$" superscript for the PAntSA* algorithm and a "$\star\star$" superscript for the ITSA* algorithm. Thus, for example, the results obtained with PAntSA* taking as input the groupings generated by K-Means, will be denoted as K-Means*, and those obtained with ITSA* with the same groupings as K-Means**.

Table 1. Best F-measures values per collection

	Micro4News			EasyAbstracts			SEPLN-CICLing			CICling-2002		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.67	0.41	0.96	0.54	0.31	0.71	0.49	0.36	0.69	0.45	0.35	0.6
K-Means*	0.84	0.67	**1**	0.76	0.46	0.96	0.63	0.44	0.83	0.54	0.41	0.7
K-Means**	0.9	0.7	**1**	0.94	0.71	**1**	0.73	0.65	0.83	0.6	0.47	0.73
K-MajorClust	0.95	0.94	0.96	0.71	0.48	0.98	0.63	0.52	0.75	0.39	0.36	0.48
K-MajorClust*	0.97	0.96	**1**	0.82	0.71	0.98	0.68	0.61	0.83	0.48	0.41	0.57
K-MajorClust**	**0.98**	**0.97**	**1**	0.92	0.81	**1**	0.72	**0.71**	**0.88**	0.61	0.46	0.73
CHAMELEON	0.76	0.46	0.96	0.74	0.39	0.96	0.64	0.4	0.76	0.46	0.38	0.52
CHAMELEON*	0.85	0.71	0.96	0.91	0.62	0.98	0.69	0.53	0.77	0.51	0.42	0.62
CHAMELEON**	0.93	0.74	0.96	0.92	0.67	0.98	0.69	0.56	0.79	0.59	0.48	0.71
CLUDIPSO	0.93	0.85	**1**	0.92	0.85	0.98	0.72	0.58	0.85	0.6	0.47	0.73
CLUDIPSO*	0.96	0.88	**1**	**0.96**	0.92	0.98	**0.75**	0.63	0.85	0.61	0.47	**0.75**
CLUDIPSO**	0.96	0.89	**1**	**0.96**	**0.94**	0.98	**0.75**	0.65	0.85	**0.63**	**0.51**	0.7

Table 2. Best F-measures values per collection

	R4			R8-			R8+		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means	0.73	0.57	0.91	0.64	0.55	0.72	0.60	0.46	0.72
K-Means*	0.77	0.58	**0.95**	0.67	0.52	0.78	0.65	0.56	0.73
K-Means**	**0.78**	0.6	**0.95**	0.68	0.61	0.78	0.65	0.56	0.73
K-MajorClust	0.70	0.45	0.79	0.61	0.49	0.7	0.57	0.45	0.69
K-MajorClust*	0.7	0.46	0.84	0.61	0.5	0.71	0.63	0.55	0.72
K-MajorClust**	**0.78**	**0.7**	0.94	**0.7**	0.58	0.75	0.64	0.57	0.68
CHAMELEON	0.61	0.47	0.83	0.57	0.41	0.75	0.48	0.4	0.6
CHAMELEON*	0.69	0.6	0.87	0.67	0.6	0.77	0.61	0.55	0.67
CHAMELEON**	0.76	0.65	0.89	0.68	**0.63**	0.75	0.65	0.6	0.69
CLUDIPSO	0.64	0.48	0.75	0.62	0.49	0.72	0.57	0.45	0.65
CLUDIPSO*	0.71	0.53	0.85	0.69	0.54	0.79	0.66	0.57	0.72
CLUDIPSO**	0.74	0.53	0.89	**0.7**	**0.63**	**0.8**	**0.68**	**0.63**	**0.74**

These results are conclusive with respect to the good performance that ITSA* can obtain with short-text collections with very different characteristics. With the exception of the F_{max} value obtained with CICling-2002, it achieves the highest F_{min}, F_{avg} and F_{max} values for all the collections considered in our study, by improving the clusterings obtained with different algorithms. Thus, for instance, ITSA* obtains the highest F_{avg} value for CICling-2002 by improving the clusterings obtained by CLUDIPSO and the highest F_{avg} value for R4 by improving the clusterings obtained by K-Means and K-MajorClust. We can also appreciate in these tables that both improvement algorithms, PAntSA* and ITSA*, obtain in most of the considered cases considerable improvements on the original clusterings. Thus, for example, the F_{avg} value corresponding to the clusterings generated by K-Means with the Micro4News collection ($F_{avg} = 0.67$), is considerably improved by PAntSA* ($F_{avg} = 0.84$) and ITSA* ($F_{avg} = 0.9$).

Table 3. Results of PAntSA* vs. ITSA*

	Micro4News			EasyAbstracts			SEPLN-CICLing			CICling-2002		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means*	0.84	0.67	**1**	0.76	0.46	0.96	0.63	0.44	**0.83**	0.54	0.41	0.7
K-Means**	**0.9**	**0.7**	**1**	**0.94**	**0.71**	**1**	**0.73**	**0.65**	**0.83**	**0.6**	**0.47**	**0.73**
K-MajorClust*	0.97	0.96	**1**	0.82	0.71	0.98	0.68	0.61	0.83	0.48	0.41	0.57
K-MajorClust**	**0.98**	**0.97**	**1**	**0.92**	**0.81**	**1**	**0.72**	**0.71**	**0.88**	**0.61**	**0.46**	**0.73**
CHAMELEON*	0.85	0.71	**0.96**	0.91	0.62	**0.98**	**0.69**	0.53	0.77	0.51	0.42	0.62
CHAMELEON**	**0.93**	**0.74**	**0.96**	**0.92**	**0.67**	**0.98**	**0.69**	**0.56**	**0.79**	**0.59**	**0.48**	**0.71**
CLUDIPSO*	**0.96**	0.88	**1**	**0.96**	0.92	**0.98**	**0.75**	0.63	**0.85**	0.61	0.47	**0.75**
CLUDIPSO**	**0.96**	**0.89**	**1**	**0.96**	**0.94**	**0.98**	**0.75**	**0.65**	**0.85**	**0.63**	**0.51**	0.7

Table 4. Results of PAntSA* vs. ITSA*

	R4			R8-			R8+		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
K-Means*	0.77	0.58	**0.95**	0.67	0.52	**0.78**	**0.65**	**0.56**	**0.73**
K-Means**	**0.78**	**0.6**	**0.95**	**0.68**	**0.61**	**0.78**	**0.65**	**0.56**	**0.73**
K-MajorClust*	0.70	0.46	0.84	0.61	0.5	0.71	0.63	0.55	**0.72**
K-MajorClust**	**0.78**	**0.7**	**0.94**	**0.7**	**0.58**	**0.75**	**0.64**	**0.57**	0.68
CHAMELEON*	0.69	0.6	0.87	0.67	0.6	**0.77**	0.61	0.55	0.67
CHAMELEON**	**0.76**	**0.65**	**0.89**	**0.68**	**0.63**	0.75	**0.65**	**0.6**	**0.69**
CLUDIPSO*	0.71	**0.53**	0.85	0.69	0.54	0.79	0.66	0.57	0.72
CLUDIPSO**	**0.74**	**0.53**	**0.89**	**0.7**	**0.63**	**0.8**	**0.68**	**0.63**	**0.74**

Tables 3 and 4 compare the results obtained with ITSA* with respect to PAntSA*. Here, the benefits of the iterative approach on the original improvement method are evident, with a better performance of ITSA* on PAntSA* in most of the considered experimental instances. As un example, when ITSA* took as input the clusterings generated by K-Means, its results were consistently better than (or as good as) those obtained by PAntSA* with the same clusterings, on the seven considered collections.

Despite the excellent results previously shown by ITSA*, it is important to observe that in a few experimental instances, ITSA* do not improve (and it can even deteriorate) the results obtained with PAntSA* or the initial clusterings generated by the other algorithms. This suggests that, despite the *average* improvements that ITSA* achieves on all the considered algorithms, and the highest F_{min} and F_{max} value obtained on most of the considered collections, a deeper analysis is required in order to also consider the *improvements* (or the *deteriorations*) that ITSA* carries out on each clustering that it receives as input.

The previous observations pose some questions about *how often* (and in *what extent*) we can expect to observe an improvement in the quality of the clusterings provided to ITSA*. Tables 5 and 6 give some insights on this subject, by presenting in Table 5 the *improvement percentage* (IP) and the *improvement magnitude* (IM) obtained with ITSA*, whereas Table 6 gives the *deterioration percentage* (DP) and the *deterioration magnitude* (DM) that ITSA* produced on the original clusterings. The *percentage* of cases where ITSA* produces clusterings with the *same quality* as the clusterings received as input (SQP) can be directly estimated from the two previous percentages. Thus, for example, ITSA* produced an improvement in the 92% of the cases when received the clusterings generated by CHAMELEON on the R4 collection, giving F-measures values which are (on average) a 0.17 higher than the F-measures values obtained with CHAMELEON. In this case, $DP = 7\%$ and $DM = 0.02$ meaning that in 1% of the experiments with this algorithm and this collection, ITSA* gave results of the same quality ($SQP = 1\%$).

Table 5. IP and MP values of ITSA* with respect to the original algorithms

	4MNG		Easy		SEPLN-CIC		CIC-2002		R4		R8-		R8+	
Algorithms	IP	MP	IP	MP	IP	MP	IP	MP	IP	MP	IP	MP	IP	MP
K-Means	100	0.3	100	0.44	100	0.27	100	0.18	96	0.06	90	0.06	88	0.07
K-MajorClust	100	0.48	100	0.49	100	0.32	100	0.22	100	0.39	100	0.29	100	0.27
CHAMELEON	100	0.18	83	0.15	50	0.16	92	0.13	92	0.17	71	0.14	100	0.18
CLUDIPSO	100	0.03	100	0.05	55	0.03	62	0.04	98	0.1	100	0.39	100	0.43

Table 6. DP and DM values of ITSA* with respect to the original algorithms

	4MNG		Easy		SEPLN-CIC		CIC-2002		R4		R8-		R8+	
Algorithms	DP	DM	DP	DM	DP	DM	DP	DM	DP	DM	DP	DM	DP	DM
K-Means	0	0	0	0	0	0	0	0	4	0.03	10	0.03	12	0.02
K-MajorClust	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CHAMELEON	0	0	16	0.05	50	0.04	7	0.03	7	0.02	28	0.001	0	0
CLUDIPSO	0	0	0	0	44	0.03	38	0.01	1	0.01	0	0	0	0

With the exception of the CHAMELEON - SEPLN-CICLing and CLUDIPSO - SEPLN-CICLing combinations, where ITSA* does not obtain significant improvements, the remaining experimental instances are conclusive about the advantages of using ITSA* as a general improvement method. Thus, for example, in 17 experimental instances (algorithm-collection combinations) ITSA* obtained an improvement in the 100% of the experiments. This excellent performance of ITSA* can be easily appreciated in Figure 2, where the IP (white bar)), DP (black bar) and SQP (gray bar) values are compared but considering in this case the improvements/deteriorations obtained in each one of the seven collections. Space limitations prevent us from showing a similar comparison of the IP, DP and SQP values of ITSA* with respect to PAntSA*, but in the last section some summarized data will be given about this aspect.

Finally, an important aspect to consider is to what extent the effectiveness of ITSA* depends on the quality of the clusterings received as input. In order

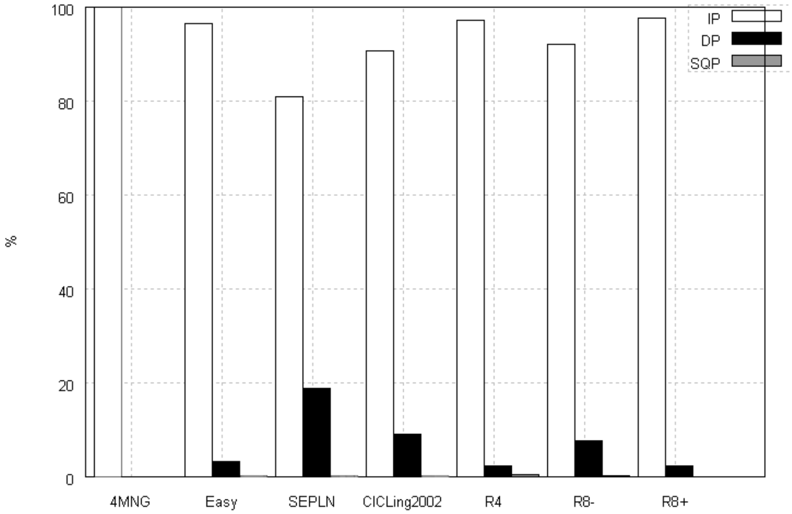


Fig. 2. *IP*, *DP* and *SQP* values per collection of ITSA⋆ with respect to the original algorithms

Table 7. Best *F*-measures values per collection (subsets of R8-Test corpus)

	R4			R8-			R8+		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
R-Clustering	0.32	0.29	0.35	0.21	0.19	0.24	0.21	0.2	0.24
R-Clustering⋆	0.68	0.48	0.87	0.63	0.52	0.73	0.64	0.54	0.7
R-Clustering⋆⋆	**0.75**	0.54	**0.94**	**0.66**	**0.57**	0.74	**0.65**	**0.57**	**0.72**
K-Means	0.73	**0.57**	0.91	0.64	0.55	0.72	0.60	0.46	**0.72**
K-MajorClust	0.70	0.45	0.79	0.61	0.49	0.7	0.57	0.45	0.69
CHAMELEON	0.61	0.47	0.83	0.57	0.41	**0.75**	0.48	0.4	0.6
CLUDIPSO	0.64	0.48	0.75	0.62	0.49	0.72	0.57	0.45	0.65

Table 8. Best *F*-measures values per collection (other corpora)

	Micro4News			EasyAbstracts			SEPLN-CICLing			CICling-2002		
Algorithms	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}	F_{avg}	F_{min}	F_{max}
R-Clustering	0.38	0.31	0.5	0.38	0.32	0.45	0.38	0.3	0.47	0.39	0.31	0.52
R-Clustering⋆	0.87	0.73	**1**	0.76	0.54	0.96	0.63	0.48	0.77	0.54	0.42	0.71
R-Clustering⋆⋆	0.9	0.73	**1**	**0.92**	0.67	**1**	**0.73**	**0.65**	0.84	**0.6**	0.46	**0.75**
K-Means	0.67	0.41	0.96	0.54	0.31	0.71	0.49	0.36	0.69	0.45	0.35	0.6
K-MajorClust	**0.95**	**0.94**	0.96	0.71	0.48	0.98	0.63	0.52	0.75	0.39	0.36	0.48
CHAMELEON	0.76	0.46	0.96	0.74	0.39	0.96	0.64	0.4	0.76	0.46	0.38	0.52
CLUDIPSO	0.93	0.85	**1**	**0.92**	**0.85**	0.98	0.72	0.58	**0.85**	**0.6**	**0.47**	0.73

to analyze this dependency, 50 clusterings generated by a simple process that randomly determines the group of each document (denoted as R-Clustering) were considered. These clusterings were given as input to PAntSA* and ITSA* as initial orderings and the results were compared with the remaining algorithms considered in our experimental work. The results of these experiments are shown in Tables 8 and 7 where it is possible to appreciate that ITSA* is very robust to low quality initial clusterings. In fact, ITSA* obtains in several collections the best F_{min}, F_{max} or F_{avg} values and, in the remaining cases, it achieves results comparable to the best results obtained with other algorithms.

4 Conclusions and Future Work

In this work we presented ITSA*, an effective iterative method for the short-text clustering task. ITSA* achieved the best F-measure values on most of the considered collections. These results were obtained by improving the clusterings generated by different clustering algorithms.

PAntSA* does not guarantee an improvement of all the clusterings received as input. However, some data about the improvements obtained with ITSA* are conclusive with respect to its strengths as a general improvement method: on a total of 1750 experiments (1750 = 7 collections × 5 algorithms × 50 runs per algorithm) ITSA* obtained 1639 improvements, 5 results with the same quality and only 106 lower quality results with respect to the initial clusterings provided to the algorithm. On the other hand, if ITSA* is compared with the results obtained with PAntSA*, it can be seen that ITSA* obtained 1266 improvements, 219 results with the same quality and only 265 lower quality results.

An interesting result of our study is the robustness of ITSA* to low quality initial clusterings received as input. In that sense, the results obtained with random initial clusterings shown similar (or better) F-measure values than those obtained by well-known algorithms used in this area. This last aspect deserves a deeper analysis which will be carried out in future works.

Acknowledgments. The first and third authors have collaborated in the framework of the TEXT-ENTERPRISE 2.0 TIN2009-13391-C04-03 research project. We also thank the ANPCyT and UNSL (Argentina) for funding the work of the first and second authors.

References

1. Pinto, D., Rosso, P.: On the relative hardness of clustering corpora. In: Matoušek, V., Mautner, P. (eds.) TSD 2007. LNCS (LNAI), vol. 4629, pp. 155–161. Springer, Heidelberg (2007)
2. Errecalde, M., Ingaramo, D., Rosso, P.: Proximity estimation and hardness of short-text corpora. In: Proceedings of TIR-2008, pp. 15–19. IEEE CS, Los Alamitos (2008)

3. Ingaramo, D., Pinto, D., Rosso, P., Errecalde, M.: Evaluation of internal validity measures in short-text corpora. In: Gelbukh, A. (ed.) CICLing 2008. LNCS, vol. 4919, pp. 555–567. Springer, Heidelberg (2008)
4. Cagnina, L., Errecalde, M., Ingaramo, D., Rosso, P.: A discrete particle swarm optimizer for clustering short-text corpora. In: BIOMA08, pp. 93–103 (2008)
5. Ingaramo, D., Errecalde, M., Cagnina, L., Rosso, P.: Particle Swarm Optimization for clustering short-text corpora. In: Computational Intelligence and Bioengineering, pp. 3–19. IOS press, Amsterdam (2009)
6. Ingaramo, D., Errecalde, M., Rosso, P.: A new anttree-based algorithm for clustering short-text corpora. Journal of CS&T (in press, 2010)
7. Ingaramo, D., Errecalde, M., Pinto, D.: A general bio-inspired method to improve the short-text clustering task. In: Proc. of CICLing 2010. LNCS. Springer, Heidelberg (in press 2010)
8. Azzag, H., Monmarche, N., Slimane, M., Venturini, G., Guinot, C.: AntTree: A new model for clustering with artificial ants. In: Proc. of the CEC 2003, Canberra, pp. 2642–2647. IEEE Press, Los Alamitos (2003)
9. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65 (1987)
10. Makagonov, P., Alexandrov, M., Gelbukh, A.: Clustering abstracts instead of full texts. In: Sojka, P., Kopeček, I., Pala, K. (eds.) TSD 2004. LNCS (LNAI), vol. 3206, pp. 129–135. Springer, Heidelberg (2004)
11. Alexandrov, M., Gelbukh, A., Rosso, P.: An approach to clustering abstracts. In: Montoyo, A., Muńoz, R., Métais, E. (eds.) NLDB 2005. LNCS, vol. 3513, pp. 8–13. Springer, Heidelberg (2005)
12. Pinto, D., Benedí, J.M., Rosso, P.: Clustering narrow-domain short texts by using the Kullback-Leibler distance. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 611–622. Springer, Heidelberg (2007)
13. Stein, B., Meyer zu Eißen, S.: Document Categorization with MajorClust. In: Proc. WITS 02, pp. 91–96. Technical University of Barcelona (2002)
14. Karypis, G., Han, E.H., Vipin, K.: Chameleon: Hierarchical clustering using dynamic modeling. Computer 32, 68–75 (1999)

Mining Interestingness Measures for String Pattern Mining[⋆]

Manuel Baena-García and Rafael Morales-Bueno

Departamento de Lenguajes y Ciencias de la Computación,
Universidad de Málaga, 29071 Málaga, Spain
{mbaena,morales}@lcc.uma.es
http://www.lcc.uma.es

Abstract. In this paper we present a novel method to detect interesting patterns in strings. A common way to refine results of pattern mining algorithms is using interestingness measures. But the set of appropiate measures is different in each domain and problem. The aim of our research is to obtain a model that classify patterns by interest. The method is based on the application of machine learning algorithms to a generated dataset from factors features. Each dataset row is associated to a factor of a string and contains values of different interestingness measures and contextual information. We also propose a new interestingness measure based on an entropy principle which improves obtained classification results. The proposed method avoids the experts having to configure parameters in order to obtain interesting patterns. We demonstrated the utility of the method by giving example results on real data. The datasets and scripts to reproduce experiments are available on-line.

Keywords: String mining, association rules, interestingness measures, pattern mining.

1 Introduction

Biological sequences generation technologies require massive data processing methods. Different string mining approaches have been studied in order to manage all these new data sources, e.g., pattern mining in strings, string clustering, string classification, etc. However, the detection of *interesting patterns* in string remains as an important issue. The known methods generate a huge quantity of patterns which is a common problem in knowledge discovery processes [1]; indeed, an expert cannot evaluate this quantity of resulting patterns. Algorithms like Apriori [2], FP-growth [3,4] o SPEXS [5] suffer from this problem. Hence, new pattern mining methods, to reduce and to cluster string patterns, are needed.

In other domain areas, pattern mining [6] has been a popular data mining technique for analyzing large data sets. In processing transactional data, e.g.,

[⋆] This work has been partially supported by the SESAAME project, number TIN2008-06582-C03-03, of the MICINN, Spain.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 560–569, 2010.

market basket analysis, the most valuable results have been obtained. But some of these results have not yet been adapted to string pattern mining.

The general results of pattern mining processes are association rules. These rules link antecedent X and consequent Y sets. This relation indicates that all items in set X appear together with all items in set Y. Different interestingness measures have been defined, or adopted from other areas, to order the rules obtained in pattern mining algorithms. By applying these measures, an expert can obtain a reduced set of patterns: interesting patterns.

But the selection of correct measures is domain dependent. Hence, we have to carefully select interestingness measures and pattern mining algorithms in order to efficiently process string data. For this reason, it is necessary to differentiate string pattern mining from sequence pattern mining. In string pattern mining the data sources are a few long strings but in sequence pattern mining the data sources are formed by a lot of short sequences. Due to the inherent high computational costs in string pattern mining problems, sequence pattern mining algorithms can only process small sets of closely related short strings.

In this paper we define a new method to use interestingness measures in pattern mining. The method is based on the application of machine learning algorithms to learn a model to classify patterns. We also propose a new interestingness measure based on an entropy principle which improves obtained classification results. And our last contribution is the empirical demonstration of the contextual information value in the discovery of interesting patterns.

To finish this introduction we fix some notations and basic concepts. Then, we present the interestingness measures used in this paper as well as a way to obtain association rules and contingency tables from factors. After that, we define a method to obtain datasets from interestingness measures information. Next, we propose a new interestingness measure. And finally, in order to validate our proposals, the experiments and results are presented.

2 Background

Basic Definitions

Let Σ be a finite set of characters (or symbols) called *alphabet*. Let $S = S_1 \dots S_n$ be a string of length n; a *factor* of S is a string $e = S_{i+1} \dots S_{i+|e|}$, where $i \geq 0$ and $i + |e| \leq n$. A factor e has a support of $k \in N$ on S if e is a factor at k positions of S (at least). Let $k \in N$, we say that a factor e is frequent (frequent pattern) in S if it has a minimum support k on the string.

Let $X \rightarrow Y$ be an association rule. A 2x2 contingency table for this rule contains the frequency terms related to sets X and Y as shown in table 1. The contingency tables are used to calculate the relationship between these sets.

2.1 Interestingness Measures

Functions to measure the interest of patterns are usually based on correlation analysis of association rule terms. This correlation is obtained from contingency

Table 1. 2x2 Contingency table

	Y	$\overline{Y}$	
X	f_{11}	f_{10}	f_{1+}
$\overline{X}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	n

tables, e.g. by using table 1 we can compute *support* of $(X \cup Y)$ as f_{11} and the *confidence* of the rule $(X \rightarrow Y)$ as $\frac{f_{11}}{f_{1+}}$.

The problem of interesting patterns and the definition of interestingness measures has been studied by different scientists. The works of Hilderman et al. [7], Tan et al. [8], Huynh et al. [9] Geng and Hamilton [10] and Lenca et al. [1] present a summary and evaluation of different interestingness measures in literature. These authors define which properties are associated to good measures and experimentally compare measures to determine the correlation between different interestingness measures. All the authors transmit a similar message: there is not

Table 2. Interestingness measures

Ref.	**Abbreviation: Name**
[10]	Acc: Accuracy
[8]	AV: Added Value/ Change of Support
[11]	BF: Bayes Factor
[12]	CConf: Causal Confidence
[12]	CConfirm: Causal Confirm
[12]	CCConf: Causal Confir_med-confidence
[12]	CSupp: Causal Support
[1]	CenConf: Centred confidence
[13]	CF: Certainty factor
[14]	CS: Collective strength
[15]	Conf: Confidence
[16]	Conv: Conviction
[8]	Cos: Cosine
[10]	Cov: Coverage
[12]	Dep: Dependency
[12]	DConfirm: Descriptive Confirm
[12]	DCConf: Descriptive Confirmed Confidence
[1]	EC: Examples & counter-examples
[8]	GI: Gini index
[8]	Lambda: Goodman-Kruskal Lambda

Ref.	**Abbreviation: Name**
[1]	-ImpInd: Implication Index
[1]	IG: Information Gain
[17]	Lift: Interest
[8]	J: J Measure
[8]	Jac: Jaccard
[18]	Kappa: Kappa
[19]	Klosgen: Klosgen
[20]	Lap: Laplace
[21]	LC: Least Contradiction
[9]	Ler: Lerman
[10]	Lev: Leverage
[9]	Loe: Loevinger
[8]	MI: Mutual Information
[8]	OddR: Odds ratio
[22]	PS: Piatetsky-Shapiro
[23]	Phi: ϕ-coeficient
[10]	Prev: Prevalence
[12]	Put: Putative Causal Dependency
[10]	Recall: Recall
[22]	RI: Rule Interest
[1]	Seb: Sebag and Schoenauer
[10]	Spec: Specificity
[15]	Supp: Support
[8]	YuleQ: Yule's Q
[8]	YuleY: Yule's Y
[1]	Zhang: Zhang

a “best” measure, each domain and problem has a different better measure set. Then, prior to applying pattern mining algorithms to a determined domain it is necessary to select the correct measure set.

In table 2 we present the selected measures to be used in this paper.

2.2 Association Rules Generation

We have to define association rules from factors in order to apply interestingness measures in a string domain. In this paper, like in other traditional approaches, we use the last symbol of the factor as the consequent of the implication. In this sense, the next definition is raised.

Definition 1. *Let $P = axb$ be a factor, the association rule linked to P has ax as antecedent and b as consequent.*

$$(ax \rightarrow b)$$

Example 1. Let S be the string representing the full-length version of the novel “El ingenioso hidalgo don Quijote de la Mancha”. By analysing the Spanish factor *que* we obtain frequencies shown in table 3.

Table 3. Absolute frequencies of factors *que*, *qu*, *ue* and *e* in string S of example 1

factor	#
que	25568
qu	29535
ue	36548
e	217906

Table 4. Contingency table for $qu \rightarrow e$

	e	$\bar{e}$	
qu	25568	3967	29535
$\overline{qu}$	192338	1840461	2032799
	217906	1844428	2062334

We can generate a contingency table (4) from the association rule ($qu \rightarrow e$) and previous frequency data. Now we can compute the value of different interestingness measures for the factor, e.g. the Lift of the rule is 8.19.

3 From Interestingness Measures to Classification Datasets

In this section, we define a new method for detecting interesting pattern in strings. The proposed method is not a new interestingness measure because it does not order the pattern set. The new method is developed in order to classify patterns in two sets: interesting patterns set and uninteresting patterns set.

Firstly, by analyzing comprehensible strings, e.g. natural language strings, we generate a dataset to be mined with machine learning algorithms. Each dataset row is associated to a factor of a string. Rows contain values of different interestingness measures and a class attribute for the factor’s interest classification. Only frequent factors, with a minimal support, are included in the dataset.

To determine the class of a factor is also an automatic task. First, we get all the words that appear in the factor by stripping separators (e.g. spaces, commas, semicolon, etc.). If all of these words are fully contained in a dictionary then we tag the factor as interesting. In other case the factor will be marked as uninteresting.

Example 2. Let *Quij* and *Quijote* two factors in the string S of example 1. By the previous method we obtain the next two rows:

$$\begin{bmatrix} \textsc{Acc} & \mathrm{AV} & \mathrm{BF} & \textsc{CConf} & \textsc{CConfirm} & \ldots & class \\ 2151 & -10312 & 2557 & 0.96 & 0.996 & \ldots & 0 \\ 1928 & -217905 & 6084 & 0.9993 & 0.895 & \ldots & 1 \end{bmatrix}$$

The first row is associated to *Quij* and tagged with 0 (uninteresting). The second row is associated to *Quijote* and is tagged with 1 (interesting).

Now, by using a machine learning algorithm we can obtain a model to classify patterns by interest.

Note 1. We name dataset D1 to a dataset with the explained configuration.

3.1 Improving Interestingness Measures with Contextual Information

The interestingness measures only take into consideration the information relative to antecedent and consequent frequencies contained in the contingency table. But more information about a pattern is available in the string. Therefore, we have the hypothesis that the more information we include in the dataset the more precision the classification model will have.

Consequently, this new information could also be important in order to detect interesting patterns. To experiment with this hypothesis, we include in the row representing a factor all the interestingness measures values for its proper prefix and proper suffix. We triplicate the number of attributes with these new data attributes.

After that, we can test if the contextual information improves the detection of interesting patterns by comparing results of this new dataset and the previous dataset. Then, the variation on the accuracy of the model inducted by the machine learning algorithm will inform us about how important this new information is.

Note 2. We name dataset D2 to a dataset with D1 attributes and contextual information.

4 Entropic Information Ratio: A New Interestingness Measure

The entropy measure has been applied to different domains like information theory or machine learning. In this section we propose using entropy as base of a new interestingness measure. This new interestingness measure is also based on a proper prefix relation. We want to measure the disorder of factors with a common proper prefix. Hence, for a factor p we obtain factors in which p is a proper prefix and compute the entropy from the frequency of these factors.

Definition 2. *Let p be a factor, let W be the factors set such as w is in W iff p is a proper prefix of w and let $\{f_1, f_2, \ldots, f_n\}$ be the relative frequencies of factors in W, normalized by the absolute frequency of p. The entropic information ratio (*EIR*) of p is defined as:*

$$\mathrm{EIR}(p) = \begin{cases} -\sum_{i=1}^{n} f_i \, log_n f_i & \text{if } |W| = n > 1 \\ 0 & \text{in other case} \end{cases}$$

Example 3. Let "resp" be a factor in the string S of example 1. For factor "resp" we obtain $W = \{respe, respi, respl, respo, respu\}$ set of factors with frequencies represented in table 5. Then, the value of entropic information ratio of "resp" is 0.2375.

Table 5. Frequencies of factors with "resp" as common proper prefix in "El Quijote"

factor	**#**	**frequency**
respe	42	0.0293
respi	2	0.0014
respl	19	0.0132
respo	1314	0.915
respu	59	0.0411

Note 3. We name dataset D3 to a dataset with D1 attributes and entropic information ratio of the factor. We name dataset D4 to a dataset with D2 attributes and entropic information ratio of the factor, of factor proper prefix and of factor proper suffix.

5 Experimental Evaluation

We use datasets obtained from different idiom natural language strings to test our proposals. These string represent the novels *El Ingenioso Hidalgo Don Quijote de la Mancha* ("The Ingenious Hidalgo Don Quixote of La Mancha") written by Miguel de Cervantes, *Le Comte de Monte-Cristo* ("The Count of Monte Cristo") written by Alexandre Dumas, *Ulysses* by James Joyce and *Les Misérables* by Victor Hugo. All these novels have been obtained from Project Gutenberg[1].

In table 6 we present the different strings and related datasets used in our experiments. The "ID" will be used to identify a string along this section. The "size" is length of each string. For each string we generate datasets as outlined in section 3 and 4. The column "Instances" represents the number of rows a dataset associated to the string. The column "I.P." represents the number of interesting patterns in the imbalanced dataset.

[1] http://www.gutenberg.org/

Table 6. String and datasets used in experiments

ID	String	Size	Instances	I.P.	Idiom
#1	The Count of Monte Cristo	2618786	118425	52610	English
#2	Don Quijote	2062751	90918	28063	Spanish
#3	Ulysses	1525781	52721	28269	English
#4	Les Misérables	3087797	136537	60339	French

The D1 datasets contain 46 attributes corresponding to each measure in table 2. The D2 datasets contain 138 attributes corresponding to each measure for the factor, factor proper prefix (tagged with "p_" + "measure name") and factor proper suffix (tagged with "s_" + "measure name"). D3 and D4 datasets add entropic information ratio measure to D1 and D2. Hence the number of attributes for D3 and D4 datasets is 47 and 141 respectively.

All our experiments were done on a dual-processor Pentium Xeon server running Ubuntu Linux and with 64GB RAM. WEKA [24] is used as machine learning framework. J48 is used as machine learning algorithm to generate classification models.

The datasets and scripts to reproduce experiments are available on http://iaia.lcc.uma.es/~baena/papers/measures_mining_datasets.tgz

5.1 Cross Validation Results

We now show how to incorporate contextual information and how the entropic information ratio measure improves the precision of generated models.

The table 7 summarizes the results of this experimental evaluation. To evaluate each dataset we use 10 Cross Validation and repeat 20 times to ensure a normal distribution of results. We obtain that each result in table is statistically different from others. The machine learning algorithm parameters stay constants in each experiment. The True Positive Ratio (TP Ratio) columns represent how many interesting tagged patterns the model detects over the total number of interesting patterns. The Precision columns represent the number of correct detected interesting patterns over the number of interesting classified patterns. We represent in bold text the better results.

Table 7. Interesting pattern true positive ratio and precision results

	D1		D3		D2		D4	
ID	TP Ratio	Precision	TP Ratio	Precision	TP Ratio	Precision	TP Ratio	Precision
#1	0.595	0.611	0.614	0.622	0.759	0.727	0.764	**0.739**
#2	0.198	0.55	0.268	0.583	0.59	0.594	0.585	**0.6**
#3	0.752	0.634	0.755	0.644	0.791	**0.78**	0.79	**0.782**
#4	0.6	0.584	0.604	0.593	0.764	0.724	0.758	**0.732**

These results support our hypothesis of why more information as well as interestingness measures is needed in order to detect interesting patterns. The relation between a factor and its prefix and suffix give to the classification model decisive information to make precise interesting pattern identification. Furthermore, as an alternative of interestingness measures, the proposed classification model based method improves the detection of interesting patterns.

5.2 Cross-Domain Results

We next show the effectiveness of the generated models to detect interesting pattern in cross-domain datasets.

To test proposed attributes in other domains we have to use a different evaluation methodology. We have to generate new datasets: a training dataset to train a classification model and a test datasets to evaluate it. Particularly, we append three of four previous D4 datasets instances in order to generate a training dataset. Then, the test dataset is the excluded one. The table 8 present results of each combination from previous D4 datasets.

Table 8. Cross-domain true positive ratio and precision results with dataset type 4

Training datasets	Test set	TP Ratio	Precision
#1 ∪ #2 ∪ #3	#4	0.538	0.592
#1 ∪ #2 ∪ #4	#3	0.064	0.721
#1 ∪ #3 ∪ #4	#2	0.567	0.37
#2 ∪ #3 ∪ #4	#1	0.397	0.653

By comparing with previous results we conclude that a training dataset from the same domain let us obtain a more precise classification model. These are promising results but new attributes could be mined to improve the classification precision.

6 Conclusions

We have defined a new method to detect interesting patterns in strings. The method is based on the generation of datasets from factor information in strings. The datasets are filled with interestingness measures and contextual information. A classification model is generated from the obtained dataset by using a machine learning algorithm. In addition, we have defined a new interestingness measure that improves classification model performance. The proposed method avoids the experts having to configure parameters. We demonstrated the utility of the method by giving example results on real data.

More promising attributes remain to be mined. Ongoing work is focused on analyzing new attributes or features based on other contextual information and calculated attributes. Then a subset of interesting attributes will be studied with feature selection algorithms.

References

1. Lenca, P., Vaillant, B., Meyer, P., Lallich, S.: Association rule interestingness measures: Experimental and theoretical studies. In: [25], pp. 51–76
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proc. 20th Int. Conf. Very Large Data Bases, VLDB, 12-15, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
3. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min. Knowl. Discov. 8(1), 53–87 (2004)
4. Borgelt, C.: An implementation of the fp-growth algorithm. In: OSDM '05: Proceedings of the 1st international workshop on open source data mining, pp. 1–5. ACM, New York (2005)
5. Vilo, J.: Discovering frequent patterns from strings. Technical report, Department of Computer Science, University of Helsinki, Finland (1998)
6. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Min. Knowl. Discov. 15(1), 55–86 (2007)
7. Hilderman, R.J., Hamilton, H.J.: Knowledge Discovery and Measures of Interest. Kluwer Academic Publishers, Norwell (2001)
8. Tan, P.N., Kumar, V., Srivastava, J.: Selecting the right objective measure for association analysis. Information Systems 29(4), 293–313 (2004)
9. Huynh, X.H., Guillet, F., Blanchard, J., Kuntz, P., Briand, H., Gras, R.: A graph-based clustering approach to evaluate interestingness measures: A tool and a comparative study. In: [25], pp. 25–50
10. Geng, L., Hamilton, H.J.: Choosing the right lens: Finding what is interesting in data mining. In: [25], pp. 3–24 (2007)
11. Jeffreys, H.: Some tests of significance, treated by the theory of probability. Proceedings of the Cambridge Philosophical Society 31, 203–222 (1935)
12. Kodratoff, Y.: Comparing machine learning and knowledge discovery in databases: an application to knowledge discovery in texts. Machine Learning and Its Applications: advanced lectures, 1–21 (2001)
13. Galiano, F.B., Blanco, I.J., Sánchez, D., Miranda, M.A.V.: Measuring the accuracy and interest of association rules: A new framework. Intell. Data Anal. 6(3), 221–235 (2002)
14. Aggarwal, C.C., Yu, P.S.: A new framework for itemset generation. In: PODS 98, Symposium on Principles of Database Systems, Seattle, WA, USA, pp. 18–24 (1998)
15. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data, pp. 207–216. ACM, New York (1993)
16. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. In: SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, Tucson, Arizona, USA, May 1997, pp. 255–264 (1997)
17. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: generalizing association rules to correlations. SIGMOD Rec. 26(2), 265–276 (1997)
18. Cohen, J.: A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20(1), 37–46 (1960)

19. Klösgen, W.: Explora: a multipattern and multistrategy discovery assistant. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in knowledge discovery and data mining, pp. 249–271. American Association for Artificial Intelligence, Menlo Park (1996)
20. Good, I.: The estimation of probabilities, Research monograph. M.I.T. Press, Cambridge (1965)
21. Az, J., Kodratoff, Y.: A study of the effect of noisy data in rule extraction systems. In: Proceedings of the Sixteenth European Meeting on Cybernetics and Systems Research (EMCSR'02), vol. 2, pp. 781–786 (2002)
22. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Piatetsky-Shapiro, G., Frawley, W. (eds.) Knowledge Discovery in Databases. AAAI/MIT Press, Cambridge (1991)
23. Yule, U.G.: On the methods of measuring association between two attributes. Journal of the Royal Statistical Society 75(6), 579–652 (1912)
24. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. SIGKDD Explor. Newsl. 11(1), 10–18 (2009)
25. Guillet, F., Hamilton, H.J. (eds.): Quality Measures in Data Mining. Studies in Computational Intelligence, vol. 43. Springer, Heidelberg (2007)

Analyzing the Impact of the Discretization Method When Comparing Bayesian Classifiers

M. Julia Flores, José A. Gámez, Ana M. Martínez, and José M. Puerta

Computer Systems Department, Intelligent Systems & Data Mining - SIMD, I3A.
University of Castilla-La Mancha, Albacete, Spain
{julia.flores,jose.gamez,anamaria.martinez,jose.puerta}@uclm.es

Abstract. Most of the methods designed within the framework of Bayesian networks (BNs) assume that the involved variables are of discrete nature, but this assumption rarely holds in real problems. The Bayesian classifier AODE (Aggregating One-Dependence Estimators) e.g. can only work directly with discrete variables. The HAODE (from Hybrid AODE) classifier is proposed as an appealing alternative to AODE which is less affected by the discretization process. In this paper, we study if this behavior holds when applying different discretization methods. More importantly, we include other Bayesian classifiers in the comparison to find out to what extent the type of discretization affects their results in terms of accuracy and bias-variance discretization. If the type of discretization applied is not decisive, then future experiments can be k times faster, k being the number of discretization methods considered.

Keywords: Discretization, Bayesian Classifiers, AODE, Naive Bayes.

1 Introduction

Discretization is probably one of the pre-processing techniques most broadly used in machine learning and data mining. Strictly speaking, by means of a discretization process the real distribution of the data is replaced with a mixture of uniform distributions. In practice, discretization can be viewed as a method for reducing data dimensionality since the input data go from a huge spectrum of numeric values to a much more reduced subset of discrete values.

The necessity of applying discretization on the input data can be caused by different reasons. Firstly, some methods can only deal with discrete variables, therefore the discretization is a requisite if we wish to apply those algorithms (e.g. certain Bayesian methods). In other cases, discretization improves the run time for the given algorithms such as decision trees, as the number of possible partitions to be evaluated is drastically decreased. Finally, there are cases where discretization is simply a pre-processing step with the aim of obtaining a reduction of the values set and, then, a reduction of the noise which is quite possibly present in the data.

Many distinct techniques for discretization can be found in literature: unsupervised (equal frequency or width, k-means) and supervised (based on entropy,

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 570–579, 2010.

1R algorithm). There is no doubt that, when dealing with a concrete problem (dataset), choosing a certain discretization method can have a direct consequence in the success (accuracy, AUC, etc.) of the posterior classification task. However, in this paper we do not focus on the effect of the chosen discretization method for a concrete application domain (dataset), but its impact when studying a BN classifier over a significant range of application domains (datasets). That is to say, should we worry about the discretization method applied when designing the set of experiments in order to study if the analyzed method is better or not than other BN classifiers? If the answer is yes, then we must add a new parameter (the discretization method) to our experimental study, and so the number of experiments will be multiplied by k, k being the number of tested discretizations. Otherwise, if the answer is no, then we can avoid introducing this parameter in the experimental study and therefore we will save a considerable amount of time in our experiments (concretely our experiments will be k times faster).

In order to answer the question posed above, in this work, we intend to perform an empiric analysis of this problem using as a base a subset of classifiers based on BNs: NB, TAN [6], AODE [9] and HAODE [5]. In [5] we show that HAODE is better than AODE for continuous attributes when applying Fayyad and Irani's discretization method, but does it hold for other kind of discretization techniques? We have also considered NB and TAN because they represent two of the most used approaches in Bayesian networks classifiers. NB is probably the most popular and classical method, still being a standard technique and its performance being quite competitive in many fields. TAN is one of the main approaches to relax the independence hypothesis imposed by NB but maintaining the complexity under control (quadratic). AODE is a relatively recent classifier which pursues the same goal as TAN, but it works by using a mixture of classifiers which results in one of the most efficient BN classifiers nowadays in terms of accuracy while keeping under control the computational complexity (quadratic) [14]. As above indicated, HAODE is a proposal from AODE structure in which the use of discretization is kept to the minimum so that numeric variables are only discretized when they act as superparents and are treated as numeric in the other cases.

Our main contribution with this study is the conclusion that when using a group of datasets sufficiently large, the ranking of the studied classifier is maintained with respect to the averaged accuracy independently from the discretization method chosen. The same pattern can be seen when the discretization bias-variance is analyzed. Furthermore, we corroborate that HAODE is less sensitive to the discretization method than AODE and obtains better performance in all cases. Of course, as a side conclusion we can corroborate that for some concrete domains the discretization method matters.

This paper is organized as follows: in Section 2, we present the distinct BN classifiers we have used in this study, whereas in Section 3 the corresponding discretization methods are described. Section 4 presents both the design of the performed experiments and their results are presented. We finally provide with a brief section that summarizes the main conclusions out of this work.

2 Bayesian Networks Classifiers

The classification task consists of assigning one category c_i or value of the class variable $C = \{c_1, \ldots, c_k\}$ to a new object $\boldsymbol{e}$, which is defined by the assignment of a set of values, $\boldsymbol{e} = (a_1, \ldots, a_n)$, to the attributes $A_1, \ldots, A_n$. In the probabilistic case, this task can be accomplished in an exact way by the application of Bayes theorem. However, since working with joint probability distributions is usually unmanageable, simpler models based on factorizations are normally used for this problem. In this work we apply three computationally efficient paradigms NB ($\mathcal{O}(n)$), TAN and AODE ($\mathcal{O}(n^2)$).

2.1 Naive Bayes

NB assumes that all attributes are conditionally independent given the value of the class, this implies the following factorization: $\forall c \in \Omega_C \; p(\boldsymbol{e}|c) = \prod_{i=1}^{n} p(a_i|c)$. Here, the maximum a posteriori (MAP) hypothesis is used to classify:

$$c_{MAP} = argmax_{c \in \Omega_C} \; p(c|\boldsymbol{e}) = argmax_{c \in \Omega_C} \; \left(p(c) \prod_{i=1}^{n} p(a_i|c) \right). \qquad (1)$$

2.2 Tree Augmented Naive Bayes (TAN)

The TAN model [6] releases the conditional independence restriction without a large increase in the complexity of the construction process. The idea behind TAN entails learning a maximum weighted spanning tree based on the conditional mutual information between two attributes given the class label, choosing a variable as root and completing the model by adding a link from the class to each attribute.

Hence, TAN becomes a structural augmentation of NB where every attribute has the class variable and at most one other attribute as its parents. It is considered a fair trade-off between model complexity and model accuracy.

2.3 Aggregating One-Dependence Estimators (AODE)

AODE [9] is considered an improvement of NB and an interesting alternative to other attempts such as (Lazy Bayesian Rules (LBR) and Super-Parent TAN (SP-TAN), since they offer similar accuracy values, but AODE is significantly more efficient at classification time comparing to the first one and at training time comparing to the second. In order to maintain efficiency, AODE is restricted to exclusively use 1-dependence estimators. Specifically, AODE can be considered as an ensemble of SPODEs (Superparent One-Dependence Estimators), because every attribute depends on the class and another shared attribute, designated as superparent.

Graphically, every classifier (BN) used in AODE will be a structure as the one depicted in Figure 1, AODE combines all possible classifiers with this pattern structure. Hence, AODE computes the average of the n possible SPODE classifiers (one for each attribute in the data base):

$$c_{MAP} = argmax_{c \in \Omega_C} \left(\sum_{j=1, N(a_j)>m}^{n} p(c, a_j) \prod_{i=1, i \neq j}^{n} p(a_i | c, a_j) \right). \qquad (2)$$

where the condition $N(a_j) > m$ is used as a threshold to avoid making predictions from attributes with few observations.

AODE is quadratic in the number of variables for the training process and also for prediction.

2.4 Hybrid Aggregating One-Dependence Estimators (HAODE)

NB can deal with hybrid (discrete and numeric variables) datasets by means of Gaussian and multinomial distributions. On the contrary, this is not possible for AODE as a numeric variable (super-parent) can not be parent of a discrete variable. This is the reason why AODE can only be applied after discretizing numeric variables. HAODE [5] restricts the use of discretization only to the variable which acts as super-parent in every model, keeping it numeric when it is playing the role of *child*. So, multinomial distributions are estimated for discrete variables and the superparent, together with one Univariate Gaussian Distribution (one per each configuration in the Cartesian product between the class and the super-parent) for each numeric variable which acts as child.

Classification is performed according to the following equation:

$$argmax_c \left(\sum_{j=1, N(a_j)>m}^{n} p(a_j, c) \prod_{i=1 \wedge i \neq j}^{n} \mathcal{N}(a_i : \mu_i(c, a_j), \sigma_i^2(c, a_j)) \right). \qquad (3)$$

HAODE presents the same time complexity as AODE producing a slight reduction in spatial complexity because HAODE requires only two parameters (mean and variance) for Gaussian distributions, independently of the number of states in which this variable has been discretized when it acts as super-parent.

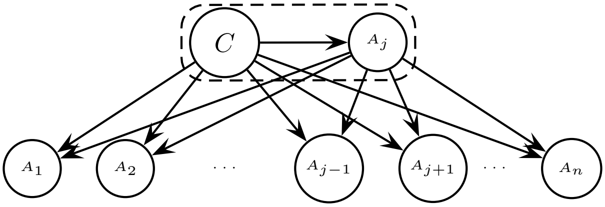


Fig. 1. General structure for a SPODE classifier

3 Evaluated Discretization Methods

Equal-width discretization [3]. This a technique for unsupervised discretization, since class value is not considered when the interval limits are selected.

Equal-width divides the range of the attribute in b bins with the same width. Its time complexity is $\mathcal{O}(t)$, t being the number of instances.

It is quite usual to set this value to 5 or 10 bins although the optimum value for b depends on, among other factors, the dataset size. The software tool Weka [12] provides the utility of searching the most appropriate value to b by means of a filter method which minimizes the partition entropy.

Equal-depth (or frequency) discretization [3]. In this unsupervised technique the values are ordered and divided into b bins so that each one contains approximately the same number of training instances. Therefore, every bin contains t/b instances with adjacent values.

Time complexity for for this technique is $\mathcal{O}(t \log t)$ as it is necessary to perform an ordering of the data.

Minimum entropy-based discretization by Fayyad & Irani [4]. It refers to a supervised technique which evaluates for cut-off point candidates those mean points between every pair of values contiguous in the ordered data. When evaluating every candidate data are divided into two intervals and the entropy for each class is computed. A binary discretization is performed in that candidate cut-off point which minimizes the entropy. This process is repeated in a recursive way by applying the Minimum Description Length (MDL) criterion to decide when to stop.

Its time complexity is $\mathcal{O}(kt \log t)$, where k stands for the number of classes.

4 Experimental Methodology and Results

4.1 Experimental Frame

The set of experiments has been carried out over 26 numeric datasets downloaded from the University of Waikato homepage [11]. We gathered together all the datasets on this web page, originally from the UCI repository [1], which are aimed at classification problems and exclusively contain numeric attributes according to Weka [12]. Table 1 displays these datasets and their main characteristics.

4.2 Analysis and Results

The experiments have been performed over the four BN classifiers above described: NB, TAN, AODE and HAODE. We have considered 6 different techniques for discretizing the datasets: equal width discretization with 5 (EW5) and 10 bins (EW10) and optimizing the number of bins through entropy minimization (EWE); equal frequency discretization with 5 (EF5) and 10 bins (EF10); and Fayyad and Irani's supervised discretization method (FeI). In all cases, the filters included in Weka [12] have been used.

For all the experiments 5x2 cross validation has been used. The Bias-Variance decomposition has been performed using the sub-sampled cross-validation procedure as specified in [10].

We analyze the results obtained in terms of accuracy and error on different blocks, the latter divided in the bias and variance components according to [8].

Study in terms of accuracy. The accuracy is an important factor when evaluating a classifier or even when studying more general data mining techniques. It is defined as the percentage of correctly predicted instances in the test dataset, i.e. it indicates how error-free these predictions are made by a specific model.

The Y-axis on Figure 2 represents the average accuracy for the 26 datasets from Table 1. The different lines correspond to the behavior of the four classifiers for the 6 discretization methods tested, which are represented on the X-axis. Although the tendency followed by the four classifiers is similar, and more importantly, the ranking among classifiers is maintained in all cases except for NB and TAN when discretizing with EF10, we can see that equal frequency discretization is specially good for AODE whereas FeI performs worse for HAODE. As we pointed out, we can also observe how HAODE among the 4 classifiers tested, is the least sensitive to the discretization method applied.

Figure 3 shows on each circular graph the individual accuracy obtained over each dataset for the four classifiers. We can see from the circular visualitation that HAODE almost always encloses the other methods, indicating that it dominates them in terms of accuracy. Similarly, the line corresponding to AODE covers TAN's and NB's; and TAN's encloses NB's in most of the cases.

Comparisons between discretization methods: Friedman's tests have been applied to perform the multiple comparison of the different discretization methods for each classifier, as well as two post-hoc tests: Nemenyi and Holm, following [2,7] guidelines. The summary results are shown on Table 2.

For NB, even though Friedman's claims statistical difference, these differences are not found by the post-hoc tests. As for TAN is concerned, both Nemenyi and Holm's find differences between FEI vs FEW and EF10. AODE and HAODE seem to be the most robust to the discretization method, as the null hypothesis is rejected by Friedman. Some differences are found by Holm's though.

Table 1. Main characteristics of the datasets: number of predictive variables (n), number of classes (k), and number of instances (t)

Id	Dataset	n	k	t	Id	Dataset	n	k	t
1	balance-scale	4	3	625	14	mfeat-fourier	76	10	2000
2	breast-w	9	2	699	15	mfeat-karh	64	10	2000
3	diabetes	8	2	768	16	mfeat-morph	6	10	2000
4	ecoli	7	8	336	17	mfeat-zernike	47	10	2000
5	glass	9	7	214	18	optdigits	64	9	5620
6	hayes-roth	4	4	160	19	page-blocks	10	5	5473
7	heart-statlog	13	2	270	20	pendigits	16	9	10992
8	ionosphere	34	2	351	21	segment	19	7	2310
9	iris	4	3	150	22	sonar	60	2	208
10	kdd-JapanV	14	9	9961	23	spambase	57	2	4601
11	letter	16	26	20000	24	vehicle	18	4	946
12	liver-disorders	6	2	345	25	waveform-5000	40	3	5000
13	mfeat-factors	216	10	2000	26	wine	13	3	178

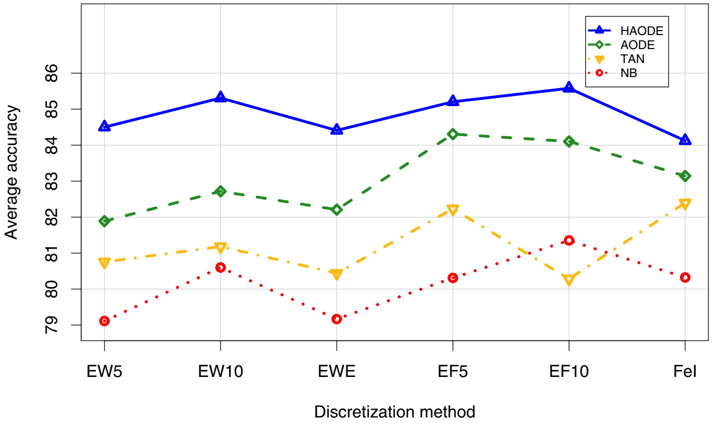


Fig. 2. Comparison of average accuracy for NB, TAN, AODE and HAODE when using different discretization methods

Comparisons between classifiers: If we change the point of view and perform the same tests comparing the different classifiers for a specific discretization method we obtain evidence of statistical difference in all cases when applying the Friedman test. According to both Holm and Nemenyi tests, HAODE is significantly better than NB and TAN in all cases. HAODE is also found to be better than AODE in all cases except when EF10 is used. The Nemenyi test also states that AODE is better than NB when EW5 is used and TAN for EF10.

It is worth noting that in all cases HAODE is placed in first position and AODE in the second one by the ranking performed by the Friedman test. NB gets the third position when equal frequency division or EW10 are used, forcing TAN to occupy the forth position in these cases.

Study in terms of bias and variance. In [13] the behavior of NB in terms of bias and variance is studied, in which they call *discretization bias and variance.* Similar analysis is intended to be carried out here.

Table 2. Test results when comparing the discretization methods over each classifier (in brackets the p-value obtained). The null hypothesis (H_0) states that there is not difference between the algorithms. In the third column, the method taken as control by the Holm test is displayed, and marked with • the methods where statistical difference is found. $\alpha = 0.05$ for all the cases.

	FRIEDMAN	**IMAN-DAV.**	**NEMENYI**	**HOLM**
NB	Reject H_0 (0.034)	Not necessary	• None	Control = EW10 •None
TAN	Reject H_0 (0.006)	Not necessary	• FEI vs (FEW&EF10)	Control = FEI • FEW,EF10,EW10
AODE	Accept H_0 (0.069)	Accept H_0 (0.065)	• None	Control = EF5 • EW10
HAODE	Accept H_0 (0.052)	Reject H_0 (0.049)	• None	Control = EF10 •FEI

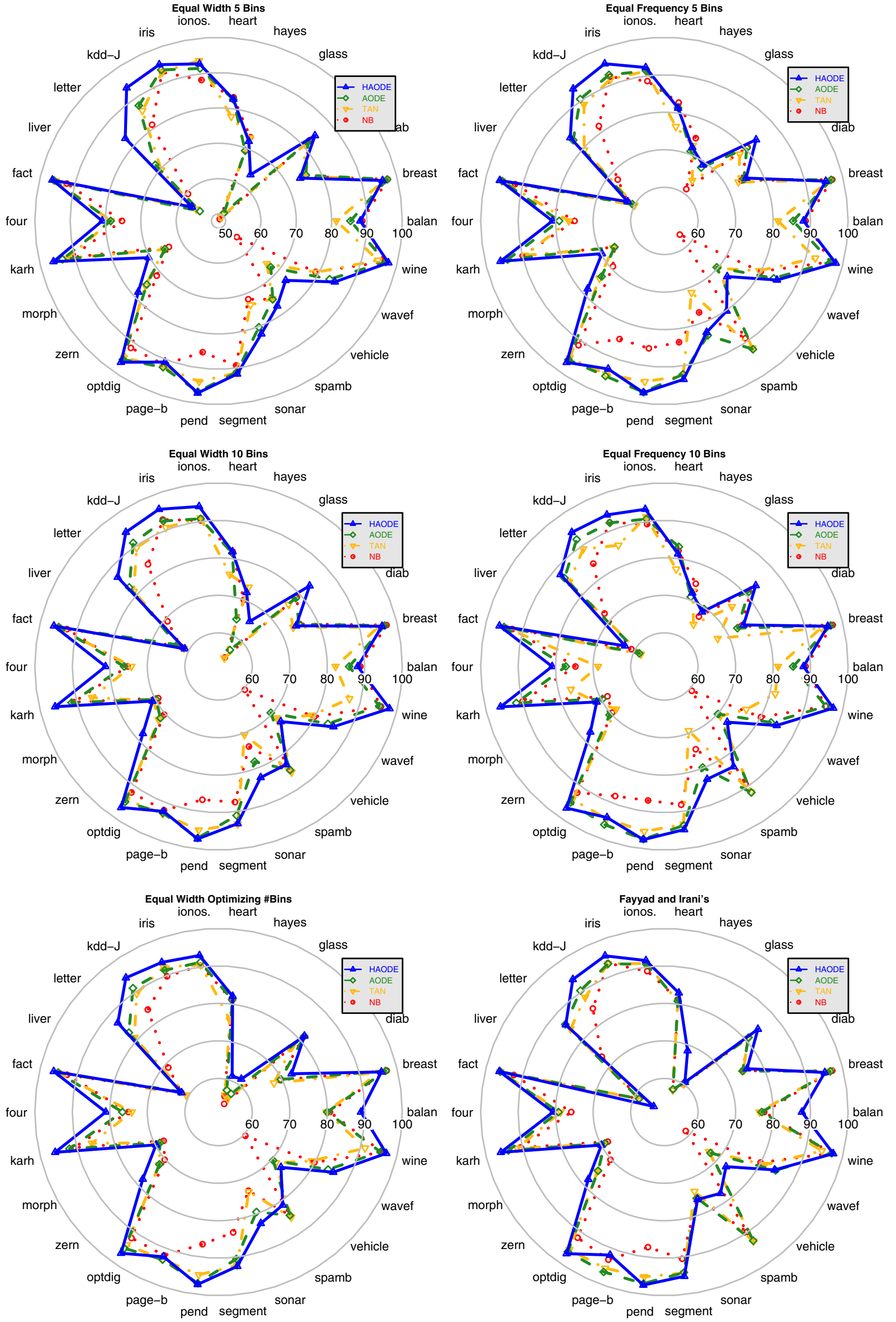
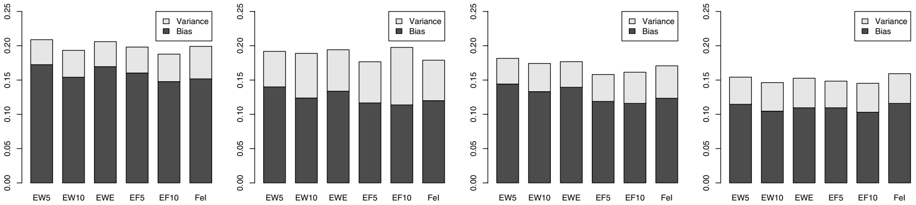


Fig. 3. Comparison of accuracy obtained on each of the 26 datasets with NB, TAN, AODE and HAODE when applying the different discretization methods. From up to down and left to right: EW5, EW10, EWE, EF5, EW10, FeI (following previous notation).



(a) NB (b) TAN (c) AODE (d) HAODE

Fig. 4. Mean error divided into bias and variance for NB, TAN, AODE and HAODE. The discretization methods are presented in the X-axis, Y-axis projects the error rate.

The error component can be divided into three terms: bias, variance and an irreducible term [8]. The **bias** describes the part from the error component that results from the systematic error when learning the algorithm, whereas the **variance** describes the random variation existing on the training data and from the random behavior when learning the algorithm. The more sensitive the algorithm is, the higher the variance becomes. The irreducible error describes the error existing in an optimal algorithm (noise level in data).

On Figure 4 the average error for every discretization method over all the dataset divided into bias and variance is shown. Now again HAODE obtains on average the lowest error rates. Similar reasoning is obtained when just the bias is considered, followed by AODE and TAN which obtain similar values. In terms of variance the analysis is slightly different. It is now NB the one that obtains the lowest variance almost in all cases, whereas TAN obtains the highest rates for all discretization methods. Note that TAN's results are similar in terms of bias than AODE's but it is when considering the variance for the former that the final error is dramatically increased.

5 Conclusions and Future Work

In this paper, we have studied the effect in terms of accuracy, bias and variance obtained when applying some of the most common discretization methods to NB, TAN, AODE and HAODE.

One of our first goals was the comparison between AODE and HAODE in order to try to see if the results in [5] could be extendable to other discretization methods. The results obtained reveal that in all cases HAODE's average accuracy is higher than AODE's, being the former significantly better than the latter in all cases except when equal frequency division with 10 bins is applied.

Nevertheless, it was of a major interest to investigate whether the application of a particular discretization technique could alter the ranking of classifiers according to the accuracy or error obtained for the four BN classifiers taken into consideration. The results indicate that no matter what discretization method we use the ranking is the same for HAODE and AODE with the rest of classifiers as their performance is sufficiently different. However, as NB and TAN obtain very similar results, its position on the ranking can be altered in a particular case. Even so, in the light of the results, we believe, that if the set of datasets is large enough, the discretization method applied becomes irrelevant in the comparison of the BN classifiers.

In the future, we pretend to extend this work by adding more sophisticated discretization techniques. They are not so well-known, but they have shown to provide promising results, e.g. proportional discretization or equal frequency division without fixing the number of bins in advance, both included in [13].

Acknowledgments. This work has been partially supported by the Consejería de Educación y Ciencia (JCCM) under Project PCI08-0048-8577574, the Spanish Ministerio de Educación y Tecnología under Project TIN2007-67418-C03-01 and the FPU grant with reference number AP2007-02736.

References

1. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, University of California, Irvine, School of Information and Computer Sciences (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Demšar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7, 1–30 (2006)
3. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and Unsupervised Discretization of Continuous Features. In: Proc. of the 12th Int. Conf. on Mach. Learn., pp. 194–202 (1995)
4. Fayyad, U.M., Irani, K.B.: Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning. In: Proc. of the 13th Int. Joint Conf. on AI, pp. 1022–1027 (1993)
5. Flores, M.J., Gámez, J.A., Martínez, A.M., Puerta, J.M.: GAODE and HAODE: Two Proposals based on AODE to deal with Continuous Variables. In: ICML. ACM Int. Conf. Proc. Series, vol. 382, p. 40 (2009)
6. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning 29, 131–163 (1997)
7. García, S., Herrera, F.: An Extension on Statistical Comparisons of Classifiers over Multiple Data Sets for all Pairwise Comparisons. J. Mach. Learn. Res. 9, 2677–2694 (2009)
8. Webb, G.I.: Multiboosting: A Technique for Combining Boosting and Wagging. Mach. Learn. 40(2), 159–196 (2000)
9. Webb, G.I., Boughton, J.R., Wang, Z.: Not So Naive Bayes: Aggregating One-Dependence Estimators. Mach. Learn. 58(1), 5–24 (2005)
10. Webb, G.I., Conilione, P.: Estimating bias and variance from data (2002)
11. Collection of Datasets avalaibles from the Weka Official Homepage (2008), http://www.cs.waikato.ac.nz/ml/weka/
12. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
13. Yang, Y., Webb, G.I.: Discretization for Naive-Bayes Learning: Managing Discretization Bias and Variance. Mach. Learn. 74(1), 39–74 (2009)
14. Zheng, F., Webb, G.I.: A Comparative Study of Semi-naive Bayes Methods in Classification Learning. In: Proc. of the 4th Australasian Data Mining Conf. (AusDM05), Sydney, pp. 141–156. University of Technology (2005)

Improving Incremental Wrapper-Based Feature Subset Selection by Using Re-ranking

Pablo Bermejo, José A. Gámez, and José M. Puerta

Intelligent Systems and Data Mining group (Computing Systems Department)
Universidad de Castilla-La Mancha, Spain
{Pablo.Bermejo,Jose.Gamez,Jose.Puerta}@uclm.es

Abstract. This paper deals with the problem of supervised wrapper-based feature subset selection in datasets with a very large number of attributes. In such datasets sophisticated search algorithms like beam search, branch and bound, best first, genetic algorithms, etc., become intractable in the wrapper approach due to the high number of wrapper evaluations to be carried out. Thus, recently we can find in the literature the use of hybrid selection algorithms: based on a filter ranking, they perform an incremental wrapper selection over that ranking. Though working fine, these methods still have their own problems: (1) depending on the complexity of the wrapper search method, the number of wrapper evaluations can still be too large; and (2) they rely in an univariate ranking that does not take into account interaction between the variables already included in the selected subset and the remaining ones. In this paper we propose to work incrementally in two levels (block-level and attribute-level) in order to use a filter re-ranking method based on conditional mutual information, and the results show that we drastically reduce the number of wrapper evaluations without degrading the quality of the obtained subset (in fact we get the same accuracy but reducing the number of selected attributes).

1 Introduction

Feature (or variable, or attribute) Subset Selection (FSS) is the process of identifying the input variables which are relevant to a particular learning (or data mining) problem [LM98, GE03], and is a key process in supervised classification. FSS helps to improve classification performance (accuracy, AUC, etc.) and also to obtain more interpretable classifiers. In the case of high-dimensional datasets, e.g., datasets with thousands of variables, FSS is even more important because otherwise the number of instances needed to obtain reliable models will be enormous (impracticable for many real applications as microarray domains).

Most algorithms for supervised FSS can be classified as *filter* or *wrapper* approaches. In the filter approach an attribute (or attribute subset) is evaluated by only using intrinsic properties of the data (e.g. statistical or information-based measures). Filter techniques have the advantage of being fast and general, in the sense that the resultant subset is not biased in favor of a concrete classifier. On

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 580–589, 2010.

the other hand wrapper algorithms are those that use a classifier (usually the one to be used later) in order to asses the quality of a given attribute subset. Wrapper algorithms have the advantage of achieving a greater accuracy than filters but with the disadvantage of being (by far) more time consuming and obtaining an attribute subset that is biased toward the used classifier. During the last decade wrapper-based FSS has been an active area of research. Different search algorithms (greedy sequential, best-first search, evolutionary algorithms, etc.) have been used to guide the search process while some classifier (e.g. Naive Bayes, KNN, etc.) is used as a surrogate in order to evaluate the goodness of the subset proposed by the search algorithm. There is no doubt that the results provided by wrapper methods are better than those obtained by using filter algorithms, but the main problem is that they do not scale well. Thus, while datasets up to 100 or 500 variables were the standard in the last decade, with the venue of 2000's decade new datasets which involve thousands of variables appeared (e.g. genetics or information retrieval based datasets), and the result is that the use of *pure* wrapper algorithms is intractable in many cases [RRAR06]. Because of this, hybrid filter-wrapper algorithms have became the focus of attention in the last few years. The idea is to use a filter algorithm whose output guides the wrapper one. In this way the advantages of the wrapper approach is retained but (strongly) reducing the number of wrapper evaluations. Examples of these algorithms are [RRAR06, FG05] that incrementally explore the attributes by following the ranking obtained by a filter measure, [GFHK09] that applies a wrapper sequential forward search but only over the first k (e.g. 100) attributes in the filter ranking, or [BGP09, RAR09] that use the filter-based ranking for a better organization of the search process (although [RRAR06] and [RAR09] offer the possibility to make ranking and search both filter or bot wrapper).

Our idea in this paper is to improve the efficiency of these so-called hybrid filter-wrapper FSS algorithms. To do this, our aim is to *drastically* reduce the number of wrapper evaluations by increasing the number of the filter evaluations carried out. Our proposal is based on working incrementally not only at the attribute level, but also at the *block* or *set* of attributes, taking into account the selected subset $\mathcal{S}$) in the previous blocks. Thus, we start by using a filter measure to rank the attributes, then an incremental filter-wrapper algorithm $\mathcal{A}$ is applied but only over the first *block*, that is, over the first B ranked attributes. Let $\mathcal{S}$ be the subset of attributes selected from this first block. Then, a new ranking is computed over the remaining attributes but taking into account the already selected ones ($\mathcal{S}$). Then, algorithm $\mathcal{A}$ is run again over the first block in this new ranking but initializing the selected subset to $\mathcal{S}$ instead of $\emptyset$. This process is iterated until no modification in the selected subset is obtained. As we show, in our experiments, the number of *re-ranks* carried out is very small, and so only a small percentage of attributes is explored, which provokes a great reduction in wrapper evaluations (and so in CPU time) but without decreasing the accuracy of the obtained output and even reducing the size of the selected subset.

Besides this introduction, this paper is organized as follows: the following section briefly presents incremental selection algorithms; next, in Section 3 we introduce our contribution/improvement based on re-ranking. Then, Section 4 contains the experimental evaluation carried out, and finally in Section 5 we state a summary of main conclusions and future work.

2 Incremental Wrapper Subset Selection Algorithms

In this section we briefly describe the Incremental Wrapper Subset Selection (IWSS) algorithms used in our re-rank algorithm.

2.1 Filter Step: Creating the Ranking

As we are in a supervised problem, in order to create the ranking a measure $m(A_i, C)$ is computed for each predictive attribute A_i. Therefore, this stage requires $\mathcal{O}(n)$ filter evaluations. As filter measure $m(A_i, C)$ is very common to use correlation and information-based metrics. In our case, we follow [RRAR06, FG05, BGP09] and Symmetrical Uncertainty (SU) is used to evaluate the *individual* merit with respect to the class for each attributes. SU is a nonlinear information theory-based measure that can be interpreted as a sort of Mutual Information normalized to interval [0,1]:

$$SU(A_i, C) = 2\left(\frac{H(C) - H(C|A_i)}{H(C) + H(A_i)}\right),$$

C being the class and $H()$ being the Shannon entropy. Attributes are ranked in decreasing SU order; that is, more informative attributes are placed first.

2.2 Incremental Wrapper Subset Selection (IWSS)

Once the ranking has been computed, the simpler incremental filter-wrapper algorithm consists of starting with $\mathcal{S} = \emptyset$ and run over the ranking by iteratively testing $\mathcal{S} \cup A_{r_i}$ in a wrapper way. Then, if the wrapper evaluation obtained is better than the current one (corresponding to $\mathcal{S}$) A_{r_i} is added to $\mathcal{S}$ and in other case is discarded. This basic behaviour [FG05] is improved in [RRAR06] by including a relevance criterion to decide when to include a new attribute in $\mathcal{S}$. The criterion used by BIRS [RRAR06] is to based on performing an inner 5 fold cross-validation, and using a t-test over the results of the 5 (tests) folds to decide if the improving (in accuracy) is relevant or not. Later, Bermejo et al. [BGP08] study different relevance criteria and conclude that a purely heuristic one like accepting the inclusion of a new attribute when the new subset has better accuracy on average (over the 5 folds) and at least in k of the folds, works fine and avoids he criticism of using a statistical test with such a small sample (5) and a non-standard high α (0.1). Here we use this acceptance criterion, *MinFoldersBetter* (mf) and requiring improvement in at least 2 folds (mf=2).

These algorithms have linear complexity order in the number of wrapper evaluations, that is, $O(n)$, n being the number of used variables.

2.3 Incremental Wrapper Subset Selection with Replacement

In [BGP09] a more sophisticated incremental wrapper algorithm is presented. Now, when an attribute ranked in position i is analyzed, then not only its inclusion is studied but also its interchange with any of the variables already included in $\mathcal{S}$. Thus, the algorithm can retract from some of its previous decisions, that is, a previously selected variable can become useless after adding some others. As shown in [BGP09] this new algorithms behaves similar to the simpler (IWSS) incremental approach with respect to accuracy but obtains more compact subsets. Of course, this search is more complex and has worst-case complexity $\mathcal{O}(n^2)$ although its actual (experimental) complexity reduces to $\mathcal{O}(n^{1.3})$ [BGP09].

2.4 Related Literature

Other hybrid filter-wrapper FSS algorithms, although not incremental, are BARS (*Best Agglomerative Ranked Subset*) [RAR09] and LinearForward subset selection (LFS) [GFHK09]. In both cases the filter ranking is used only in the initial step, and not as the order in which the attributes are studied for its inclusion in the selected subset. BARS is based on a heuristic rule that obtains good subsets by iterating between two phases: ranking of (promising) subsets and generation of new candidate subsets by combining those previously ranked. The heuristic nature of the algorithm lets it evaluate a reduced number of candidate subsets and its outstanding characteristic is that it obtains very compact subsets. On the other hand, LFS algorithm is simpler, the ranking is just used to select the first k attributes and then a wrapper algorithm is used by taking this k attributes as input but without tanking advantage of the ranking. In [GFHK09] sequential forward selection (SFS) [GE03] is used as wrapper algorithm.

3 Re-ranking Based Incremental Wrapper Subset Selection

Some of the algorithms described in the previous section use the ranking just to get the first k variables and then launch a more sophisticated method over a smaller search space. LinearForward [GFHK09] and BARS [RAR09] take this decision at the initial stage and in a static way, while IWSS and IWSS$_r$ [BGP09] take this decision in a dynamic way, adjusting the number of attributes to study each time $\mathcal{S}$ is modified. The main criticism to this behaviour is that the ranking is based on the *individual* merit of each variable with respect to the class, but do not take into account possible interactions between the attributes. That is, if we have a subset of attributes $X_1, \ldots, X_n$ individually highly correlated with the class but also correlated among them, then *all* these variables will be in the first positions of the ranking, although only one (or few) of them will be likely selected by the wrapper stage. In order to alleviate this problem, the usual choice is to select a large number of variables (e.g. 20% - 50%), however, this decision implies that a great number of wrapper evaluations have to be carried out. On the other

```
In   $\mathbf{T}$ training set, $M$ filter measure, $\mathcal{C}$ classifier, $B$ block size
Out  $\mathcal{S}$   // The selected subset
list $R = \{\}$   // The ranking, best attributes first
for each predictive attribute $A_i$ in $\mathbf{T}$
   $Score = M_{\mathbf{T}}(A_i$, class)
   insert $A_i$ in $R$ according to $Score$
$sol.\mathcal{S} = \emptyset$   // selected variables
$sol.eval = null$   // data about the wrapper evaluation of $sol.\mathcal{S}$
$\mathcal{B}$ = first block of size $B$ in $R$   // $\mathcal{B}$ is ordered
Remove first $B$ variables from $R$
$sol$ = IncrementalSelection($\mathbf{T}$,$\mathcal{B}$,$\mathcal{C}$,$\mathcal{S}$)
$continue = true$
while $continue$ do
   $R' = \{\}$
   for each predictive attribute $A_i$ in $R$
      $Score = M_{\mathbf{T}}(A_i$, class $|sol.\mathcal{S})$
      insert $A_i$ in $R'$ according to $Score$
   $R = R'$
   $\mathcal{B}$ = first block of size $B$ in $R$   // $\mathcal{B}$ is ordered
   Remove first $B$ variables from $R$
   $sol'$ = IncrementalSelection($\mathbf{T}$,$\mathcal{B}$,$\mathcal{C}$,$\mathcal{S}$)
   if($sol.\mathcal{S} == sol'.\mathcal{S}$) //no new feature selected
      then $continue = false$
      else $sol = sol'$
return ($sol.\mathcal{S}$)
```

Fig. 1. Re-Ranking Canonical Algorithm

hand, there could be attributes that are marginally uncorrelated with the class, but conditionally correlated with the class given some other attribute(s). In this case, those variables will be ranked in the last positions and so the only way to explore them is to use the full ranking.

In this paper we propose to use *re-ranking* as a way to overcome the two problems identified above. Our idea is to work by using *blocks* (subsets) of variables computed from the ranking, but instead of using always the initial (univariate) ranking we propose to re-rank the remaining attributes by taking into account the current selected subset $\mathcal{S}$. In this way, (1) correlated attributes with $\mathcal{S}$ will be placed at the end of the new ranking because they add nothing to the class once we know the value of variables in $\mathcal{S}$; and (2) conditionally correlated variables with the class will be placed early in the ranking if the conditional correlation is due to variables included in $\mathcal{S}$.

The algorithm for *re-ranking* based incremental selection is shown in Figure 1. The following points must be detailed:

- *Selection algorithm.* As selection algorithm any incremental one can be used. In this paper we try with IWSS and IWSS$_r$. The only needed modifications are: (1) the algorithm is seeded with an initial selected subset; and (2) there is no need to compute the ranking because we receive it as a parameter $\mathcal{B}$.

- *Stop criterion.* As can be observed the algorithm stops when analyzing a new block does not produce a modification in the selected subset, that is, it returns the same subset received as seed. This is an interesting point because we do not have to decide in advance the number of attributes to explore.
- *Block size.* The block size is a key parameter in this approach. This value must be large enough to give some freedom to the wrapper algorithm, but not so large to explore a great deal of useless attributes, vanishing in this way the advantages of using re-ranking. As we will see in the experiments, only a few re-ranks are carried out in most datasets, so we think that values about 50 are adequate to get the aforementioned balance.
- *Re-ranking algorithm.* In order to build the ranking of the remaining attributes, $\{A_1, \ldots, A_r\}$, but considering the current selected subset, we need to score $M(A_i, C|\mathcal{S})$ for each $i = 1, \ldots, r$. As we know, exact computation of this term is not feasible even for moderate sizes of $\mathcal{S}$ because we will need very large (#instances) training sets and too much time. Of course, if the size of $\mathcal{S}$ grows, then this expression is directly computationally intractable. In the literature we can find different ways to approximate this score. In this paper we use CMIM, an approach based on using conditional mutual information [FG04]. CMIM tries to balance the amount of information present for each candidate attribute A_i and class C, and the fact that this information might have been already caught by some feature $A_j \in \mathcal{S}$. Thus, this method selects features maximizing their mutual information with the class but minimizing their pair-to-pair dependency. In our case, given that we have a selected subset $\mathcal{S}$ and a set of attributes to rank $\{A_1, \ldots, A_r\}$, the merit $M(A_i, C|\mathcal{S})$, $i = 1, \ldots, k$ is computed as:

  $$M(A_i, C|\mathcal{S}) = min_{A_j \in \mathcal{S}} I(A_i; C|A_j)$$

  Besides, we replace mutual information $I()$ by Symmetrical Uncertainty $SU()$. As we can see, the number of calls to the filter measure each time we have to re-rank attributes is $r \cdot |\mathcal{S}|$. However these computations are by far compensated by the extreme reduction in the number of calls to the wrapper evaluator.

4 Experiments

4.1 Test Suite

As we state in the introduction of this paper our research in this study is motivated by high dimensional datasets, thus we consider 12 publicly obtained datasets: seven microarrays related to cancer prediction (Colon, Leukemia, Lymphoma, GCM, DLBCL-Stanford, ProstateCancer and LungCancer-Harvard2) and five datasets used in the NIPS 2003 feature selection challenge (Arcene, Madelon, Dorothea, Dexter and Gisette). The first four microarrays datasets can be downloaded from *http://www.upo.es/eps/aguilar/datasets.html* while the last three are available at *http://sdmc.i2r.a-star.edu.sg/rp/*. Datasets from the feature selection challenge can be obtained from the web page of NIPS 2003.

Table 1 shows the number of features and records/instances each dataset contains. As we can observe from this table, specially for the case of microarrays datasets, the number of instances is really small.

Table 1. Number of attributes and instances in the used datasets

Dataset	**#Features**	**#Instances**	**Dataset**	**#Features**	**#Instances**
Colon	2000	62	Leukemia	7129	72
Lymphoma	4026	96	DLBCL	4026	47
Prostate	12600	136	Lung	12533	181
GCM	16063	190	Arcene	10000	100
Madelon	500	2000	Dorothea	100000	800
Dexter	20000	300	Gisette	5000	6000

4.2 Algorithms

The goal of our experiments is to test the goodness of introducing re-ranking in incremental wrapper-based subset selection. To do this, and as a preliminary study we focus on the use of re-ranking in combination with IWSS and IWSS$_r$ algorithms, using min folders better equals to 2 in both cases. Furthermore, because of the lack of space we only test with Naive Bayes as classifier, whose performance is known to be quite sensible to irrelevant and redundant attributes. For comparison in the number of evaluations, we use IWSS, IWSS$_r$, BARS (ϵ=100 and k=3), FSS and LFS algorithms.

In all the cases we use a 10 fold cross validation and the average over the test fold is reported.

4.3 Experiments Results

First, we study the case of IWSS, thus we run the IWSS and re-ranking based IWSS with block size of 5, 10, 20, 30, 40 and 50 over the twelve datasets. The results related to the quality of the obtained output (accuracy and number of selected attributes) are shown in Table 2.

The last three rows of the table show the average value (arithmetic and geometric mean) over the 12 datasets and the result of carrying out the non-parametrical Wilcoxon Matched-Pairs Signed-Ranks Test [Wil45, Dem06] between original IWSS (second column) and IWSS with re-ranking for each block size. Statistical tests are performed with a confidence level of $\alpha = 0.05$, and the result of a test is indicated in the corresponding cell as: = if there is no statistical difference, $\Uparrow$ if the re-rank algorithm returns a value significantly greater, and $\Downarrow$ if the re-rank algorithm returns a value significantly smaller. As we can observe, taking as input the averaged results over the twelve classifiers, there is no difference in accuracy between using or not re-ranking, while a significant reduction is obtained with respect to the size of the selected subset. This analysis holds for all the tested block sizes and the result is also the same for IWSS$_r$ algorithm (Table 3). Thus, any block size seems to be optimal, so sizes $B = 10$ could be suggested as standard use.

Table 2. Results using Naive Bayes classifier, IWSS^2 selection algorithm and re-ranking with block sizes B

DataSet	**IWSS^2**		**B=5**		**B=10**		**B=20**		**B=30**		**B=40**		**B=50**	
	Acc	#atts	Acc	#atts	Acc	#atts	Acc	#atts	Acc	#atts	Acc	#atts	Acc	#atts
Colon	80.6	3.8	80.6	2.8	83.9	3.0	82.3	3.2	82.3	3.3	82.3	3.3	82.3	3.3
Leukemia	87.5	2.5	87.5	2.0	87.5	2.4	87.5	2.4	87.5	2.4	87.5	2.4	87.5	2.5
Lymphoma	76.0	8.8	66.7	6.3	75.0	7.6	76.0	7.9	77.1	8.0	75.0	8.1	77.1	8.2
DLBCL	85.1	1.9	89.4	1.5	87.2	1.6	85.1	1.7	85.1	1.7	85.1	1.7	85.1	1.7
Prostate	77.9	11.1	72.1	4.1	74.3	4.0	77.9	5.6	74.3	7.3	72.1	7.8	74.3	8.0
Lung	97.2	2.7	96.7	2.2	96.7	2.4	97.2	2.7	97.2	2.7	97.2	2.7	97.2	2.7
GCM	64.2	36.6	54.2	12.3	60.0	19.8	62.1	21.4	65.3	22.5	64.2	24.4	64.7	27.4
Arcene	70.0	13.4	70.0	3.5	68.0	5.1	70.0	6.8	70.0	7.0	70.0	7.8	69.0	7.8
Madelon	59.9	13.3	61.3	2.7	60.9	4.8	60.3	7.1	59.8	8.0	60.0	10.1	59.6	11.4
Dorothea	93.5	7.4	93.9	2.8	94.1	3.6	94.4	3.8	94.0	4.3	93.9	4.3	93.8	4.5
Dexter	81.0	19.6	81.7	11.9	83.7	13.1	83.7	14.8	81.3	15.7	83.0	15.2	80.7	14.9
Gisette	94.7	112.6	88.7	18.3	92.3	41.3	93.7	62.6	93.9	69.5	94.4	82.0	94.1	77.2
Arith. Mean	*80.6*	*19.5*	*78.5*	*5.9*	*80.3*	*9.1*	*80.9*	*11.7*	*80.6*	*12.7*	*80.4*	*14.2*	*80.4*	*14.1*
Geom. Mean	*79.81*	*9.45*	*77.41*	*4.24*	*79.34*	*5.52*	*79.97*	*6.50*	*79.81*	*6.94*	*79.51*	*7.32*	*79.59*	*7.49*
Test			=	⇓	=	⇓	=	⇓	=	⇓	=	⇓	=	⇓

Table 3. Results using Naive Bayes classifier, IWSS^2_r selection algorithm and re-ranking with block sizes B

DataSet	**IWSS^2_r**		**B=5**		**B=10**		**B=20**		**B=30**		**B=40**		**B=50**	
	Acc	#atts	Acc	#atts	Acc	#atts	Acc	#atts	Acc	#atts	Acc	#atts	Acc	#atts
Colon	83.9	2.8	82.3	2.2	85.5	2.2	83.9	2.4	83.9	2.4	83.9	2.3	83.9	2.3
Leukemia	87.5	2.0	87.5	1.9	87.5	1.9	87.5	1.9	87.5	1.9	87.5	1.9	87.5	2.0
Lymphoma	80.2	5.9	72.9	4.7	75.0	5.6	80.2	5.6	81.3	5.7	78.1	5.9	76.0	5.9
DLBCL	80.9	1.8	87.2	1.5	83.0	1.6	80.9	1.7	80.9	1.7	80.9	1.7	80.9	1.7
Prostate	78.7	7.0	73.5	3.2	75.7	3.4	77.2	4.2	80.9	4.7	80.1	4.7	83.1	4.8
Lung	97.2	2.4	96.7	2.2	96.7	2.2	97.2	2.4	97.2	2.4	97.2	2.4	97.2	2.4
GCM	59.5	19.9	51.6	7.4	53.7	10.3	57.9	10.9	57.9	11.8	60.0	13.4	62.1	14.2
Arcene	72.0	6.2	71.0	2.6	70.0	3.7	72.0	3.8	73.0	4.3	71.0	4.3	69.0	4.3
Madelon	60.5	8.0	61.7	2.0	60.8	3.4	61.3	4.8	60.3	5.9	60.1	5.6	60.5	6.2
Dorothea	92.9	6.3	94.3	3.0	93.3	4.0	93.4	5.0	93.0	5.3	92.9	5.3	92.9	5.3
Dexter	83.0	12.9	82.7	8.5	81.0	10.1	83.0	9.8	82.7	9.7	82.7	9.8	82.7	9.4
Gisette	94.1	30.7	86.2	2.7	88.5	6.4	90.8	10.8	91.6	15.9	92.3	16.5	92.6	17.1
Arith. Mean	*80.9*	*8.8*	*79.0*	*3.5*	*79.2*	*4.6*	*80.4*	*5.3*	*80.8*	*6.0*	*80.6*	*6.2*	*80.7*	*6.3*
Geom. Mean	*79.97*	*6.06*	*77.82*	*3.01*	*78.16*	*3.81*	*79.54*	*4.35*	*79.93*	*4.72*	*79.68*	*4.76*	*79.85*	*4.85*
Test			=	⇓	=	⇓	=	⇓	=	⇓	=	⇓	=	⇓

Once we have stated that the use of re-ranking does not degrade the quality of the obtained output (on the contrary, it gets more compact subsets), we study the behaviour of the re-ranking algorithm in terms of wrapper evaluations (by far, the most expensive ones). Table 4 shows the number of evaluations and re-ranks carried out by the algorithm when using IWSS (IWSS$_r$) and Naive Bayes as classifier. The results shown are for the 6 block sizes considered and averaged (arithmetic and geometric mean) over the 12 datasets. As we can observe the reduction with respect to the number of evaluations carried out is really impressive and the number of re-ranks very small. A global comparison including also Sequential Forward Selection, BARS and LinearForward can be viewed in Figure 2 (note the log-scale in Y-axe), which shows the great reduction in number of evaluations that the re-ranking method provides (besides achieving more compact subsets).

Table 4. Mean number of evaluations and re-ranks performed over the 12 datasets

	#evals (arith.)	**#evals (geom.)**	**%arith.**	**%geom.**	**#re-ranks**
B=5	18.33	15.97	99.90	99.82	2.47
B=10	40.83	33.60	99.79	99.63	2.98
B=20	81.67	66.62	99.57	99.26	3.03
B=30	113.75	95.10	99.41	98.95	2.76
B=40	152.00	126.61	99.21	98.60	2.78
B=50	185.17	157.43	99.03	98.26	2.68
IWSS^2 (19175.92 evals.)					
B=5	55.93	45.43	99.95	99.91	1.53
B=10	156.24	120.03	99.87	99.76	1.84
B=20	338.38	264.76	99.71	99.47	1.84
B=30	621.43	442.74	99.47	99.11	1.97
B=40	821.40	565.14	99.31	98.87	1.78
B=50	959.81	699.00	99.19	98.60	1.63
IWSS$_r^2$ (118242.93 evals.)					

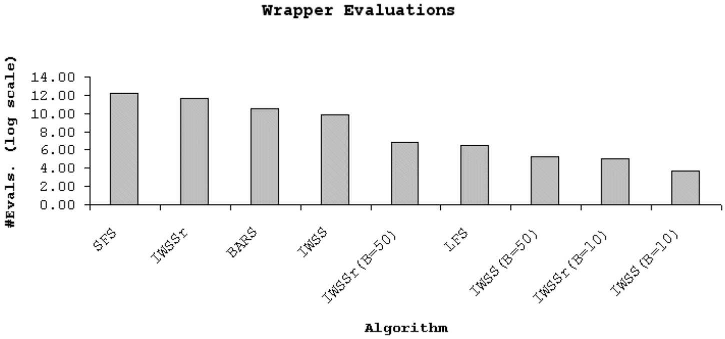


Fig. 2. Number of Wrapper Evaluations for each selection algorithm and re-ranking

5 Conclusions and Future Work

In this paper we have presented a significant improvement for incremental feature subset selection algorithms. The idea is based on working by blocks/sets of attributes that are taking at the header of the current ranking. The novelty is that the ranking changes each time we analyze a block. Thus, after analyzing the current block we perform a re-rank based on conditional mutual information that takes into account the variables already included in the selected subset. The advantage of this approach is that attributes that become redundant because of the current content of the selected subset will go to the end of the new ranking, while attributes that become relevant because interactions with the content of the current selected subset will be placed at the beginning of the new ranking. The results show that this way of proceed leads to a reduced number of re-ranks, which means that only a few blocks (and so attributes) are analyzed and so the method is (by far) more efficient. On the other hand the results also show that the quality of the selected subset does not degrade, on the contrary, the same (classification) accuracy is obtained but reducing the number of selected attributes.

In the future we plan to analyze the impact of different re-ranking algorithms and to extend the study to non-incremental feature subset selection algorithms, like BARS and sequential forward selection.

Acknowledgments

This work has been partially supported by the JCCM under project PCI08-0048-8577 and the *Secretara de Estado de Universidades e Investigacin* under project ProGraMo (TIN2007-67418-C03-01).

References

[BGP08] Bermejo, P., Gámez, J.A., Puerta, J.M.: On incremental wrapper-based attribute selection: experimental analysis of the relevance criteria. In: IPMU'08: Proceedings of the 12th Intl. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (2008)

[BGP09] Bermejo, P., Gámez, J.A., Puerta, J.M.: Incremental wrapper-based subset selection with replacement: An advantageous alternative to sequential forward selection. In: Proceedings of the IEEE Symposium Series on Computational Intelligence and Data Mining (SSCI CIDM-2009), pp. 367–374 (2009)

[Dem06] Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)

[FG04] Fleuret, F., Guyon, I.: Fast binary feature selection with conditional mutual information. Journal of Machine Learning Research 5, 1531–1555 (2004)

[FG05] Flores, J., Gámez, J.A.: Breeding value classification in manchego sheep: a study of attribute selection and construction. In: Khosla, R., Howlett, R.J., Jain, L.C. (eds.) KES 2005. LNCS (LNAI), vol. 3682, pp. 1338–1346. Springer, Heidelberg (2005)

[GE03] Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)

[GFHK09] Gutlein, M., Frank, E., Hall, M., Karwath, A.: Large-scale attribute selection using wrappers. In: CIDM, pp. 332–339 (2009)

[LM98] Liu, H., Motoda, H.: Feature Extraction Construction and Selection: a data mining perspective. Kluwer Academic Publishers, Dordrecht (1998)

[RAR09] Ruiz, R., Aguilar, J.S., Riquelme, J.: Best agglomerative ranked subset for feature selection. In: JMLR: Workshop and Conference Proceedings New Challenges for feature selection, vol. 4, pp. 148–162 (2009)

[RRAR06] Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Incremental wrapper-based gene selection from microarray data for cancer classification. Pattern Recogn. 39, 2383–2392 (2006)

[Wil45] Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin 1, 80–83 (1945)

Information Extraction from Helicopter Maintenance Records as a Springboard for the Future of Maintenance Text Analysis

Amber McKenzie[1,3], Manton Matthews[1], Nicholas Goodman[2,3], and Abdel Bayoumi[2,3]

[1] Department of Computer Science
University of South Carolina, Columbia, South Carolina, U.S.A.
[2] Department of Mechanical Engineering
University of South Carolina, Columbia, South Carolina, U.S.A.
[3] Condition-Based Maintenance Center
University of South Carolina, Columbia, South Carolina, U.S.A.

Abstract. This paper introduces a novel application of information extraction techniques to extract data from helicopter maintenance records to populate a database. The goals of the research are to preprocess the text-based data for further use in data mining efforts and to develop a system to provide a rough analysis of generic maintenance records to facilitate in the development of training corpora for use in machine-learning for more refined information extraction system design. The Natural Language Toolkit was used to implement partial parsing of text by way of hierarchical chunking of the text. The system was targeted towards inspection descriptions and succeeded in extracting the inspection code, description of the part/action, and date/time information with 80.7% recall and 89.9% precision.

Keywords: Natural language processing, Information extraction, Data preprocessing.

1 Introduction

Condition-based maintenance (CBM) is the practice of relying on mechanical indicators, such as vibration and oil debris, to detect and characterize faults and wear on machinery and basing maintenance practices on such indicators rather than on a given timeframe determined by the manufacturer-designated life of a part. CBM is a rapidly growing research field that relies mainly on vibration data and historical maintenance records to improve CBM techniques, identify new condition indicators, and refine algorithms. For several years, the Condition-Based Maintenance Center at the University of South Carolina (USC) has been involved with the South Carolina Army National Guard (SCARNG) and funded by the U.S. Department of Defense to conduct research to both enhance CBM practices and provide a cost benefit analysis. The data provided includes vibration data from fleet helicopters and historical maintenance reports which include maintenance test flight, part requisition, and fault and

N. García -Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 590–600, 2010.

action reports. While much research and data analysis has been conducted on vibration data and select fields in helicopter maintenance records in order to improve and perfect CBM techniques, the wealth of information contained in the text-based portions of maintenance records (e.g. fault and action descriptions from TAMMS-A 2408-13-1 and -2 Army maintenance reports) has largely gone untapped due to the significant amounts of time and human effort required to analyze them by hand.

Around 2000, SCARNG migrated to an electronic system for maintenance record-keeping from their previous paper-based system. Though the records are now stored in a database, the same basic record format was retained, including the text-based fields. This text would be much more suitable for the data mining necessary for CBM research if important key pieces of information were extracted out to populate queriable database fields. The value of this data for CBM research is significant and motivates this research to develop an efficient method for analyzing the data. Advances in the computer science/linguistics field of natural language processing (NLP), specifically information extraction (IE), have made it possible to employ IE techniques to allow a computer to extract usable information from these text fields in a form suitable to populate fields in a database, thus significantly augmenting the collection of CBM data available for computer-based analysis and facilitating the use of data mining techniques. Another goal of the research efforts at the CBM Center is data fusion, or to detect patterns and draw conclusions based on the integration of the different types of CBM data available.

2 Information Extraction

While many IE systems have already been developed for use in analyzing a variety of different types of texts, the majority of these systems are developed to analyze documents written in Standard English, such as news articles and literature. These systems are not suitable for use with this data, given the informal and often ambiguous nature of the language used in these reports. A tailored system must be developed that takes into account the unique and specialized vocabulary and syntax used by maintenance personnel in these reports, which is more similar to shorthand notation than Standard English usage. Research into information extraction in shorthand-type text for specialized domains mainly focuses around medical records [1-4]. Specific to a data set with a shorthand-type notation, Kraus et al. did work on extracting drugs and dosages from medical notes written by a physician [1]. For their approach, they used both supervised learning and a knowledge-based approach, which were well suited for implementation in their domain because of the consistent and predictable nature of the information they were trying to extract (i.e. drug names and numbers related to dosages). Also, a significant amount of work has been done on extracting temporal information from shorthand triage notes, medical records, and e-mails [3-5]. Little research has been done on information extraction for the maintenance record domain.

For the majority of these systems, a training set and gold standard dataset were necessary to train a system for use on the specialized set of data. In a domain where no such training set is available, much time and effort would have to be expended to produce a dataset sufficiently large enough to adequately train such as system. In addition, there is much room for human error during a supervised learning process that will result in more time to find and fix these problems later [6]. For this reason, basic

NLP techniques were used to develop a new system capable of a basic analysis of helicopter maintenance records. It is the goal of this research not only to extract usable information to populate a research database, but also to utilize the system to assist in the creation of a sufficiently large tagged corpus that could be used for future research into automated information extraction from maintenance records in general.

2.1 Data

The dataset for text-analysis comprises historical TAMMS-A 2408-13-1 and -2 Army helicopter maintenance records, which contain information regarding reported faults or scheduled inspections/checks and the corresponding maintenance actions. Specifically, the text field in the report containing the fault description or a description of the inspection or check performed is being targeted for the IE process. Certain notable characteristics of the dataset are easily computed [7]. It contains approximately 100,000 individual records with a lexicon of over 34,000 words or acronyms. The most interesting characteristic is that 80% of the records can be expressed using approximately 20% of the lexicon.

A number of challenges for the data processing system were identified after an initial examination of the text. In reference to the lexicon, the majority of the words are domain-specific, particularly regarding helicopter parts and maintenance actions, and consist of abbreviations or acronyms, some of which are specific to individual, or a group of specific, maintainers. Acronyms are not always consistent, and abbreviations are inconsistently marked with periods. Most of the fault descriptions do not constitute complete English sentences, and most are ungrammatical. Apostrophes are used to contract words and are occasionally omitted. Spaces are often omitted, and misspellings are common. Many decisions had to be made about how many and which dataset problems to resolve and which were not critical detriments to system performance.

Three main types of fault descriptions are able to be differentiated: inspections, checks, and fault/actions. Because of the highly variable nature of these reports, inspection descriptions were chosen as input for the system based on the fact that they are easily distinguishable from other types of descriptions, and the system output is more easily validated by cross checking with inspection code information. Inspection descriptions comprise roughly forty percent of the records in the entire dataset and contain information regarding the inspection code, a description of the inspection or the part involved, and when the inspection was due or completed.

2.2 Tools and Techniques

The IE domain encompasses a broad range of researched techniques, as well as open-source toolkits, available for implementation in the development of an IE system. For this research, tools and techniques were used which were flexible enough to be tailored to suit the demands and requirements for analyzing the given text. The Natural Language Toolkit (NLTK) is an open-source package of programs and modules geared towards language analysis and computational linguistic tasks [8]. The advantages of this over other toolkits or IE systems are that it is open-source, provides basic, easy-to-use NLP functionality, and is completely self-contained [9]. The available programs and functions are able to be fine-tuned and manipulated to tailor them for specific needs. The Python programming language it is written in is well-suited for NLP tasks [9]. In this way, NLTK was seamlessly applied to developing an IE system from the bottom up and was adaptable enough so as not to constrain creative development.

A partial parsing approach was implemented because a full text analysis was not necessary in order to extract the desired information. Abney's technique of finite-state cascades was implemented to handle the nested syntactic structure needed to capture meaningful chunks of data from the text descriptions [10]. These cascades obviate the need for a complete grammar of ungrammatical input and allow a parsing of only relevant, desired information from the text.

3 Methodology

For this research, a sequence of NLP tasks were individually tailored to process text for this specific engineering domain. Given the wide range of NLP and IE tools available, it was necessary to begin with basic, individual NLP processing techniques in order to facilitate the identification of potentially useful tools that would be uniquely applicable to the maintenance domain. An individually tailored IE system will also enable the faster processing of text to develop a gold standard and training corpus with which to implement machine learning techniques to discover extraction patterns, templates and rules, thus allowing for the automation of IE system development in the future for records in the same domain. The developed IE process involves four main steps: text preprocessing, part-of-speech (POS) tagging, chunking/parsing, and relation extraction. These steps combine to produce a template that is filled out for each record and is easily migrated into the database. Templates also play a critical role in allowing for the evaluation of the system [11]. The main goal of the system is to produce extracted information that, once entered into the database, will greatly facilitate data mining CBM research and data fusion efforts.

3.1 Text Preprocessing

The historical helicopter maintenance data is stored in a database and currently accessed through Microsoft Access, though database renovations and improvements are currently being implemented. The final form and interface for the database have yet to be determined. For this reason and for the ease of system development, the relevant text fields were simply exported as a text document for system input. In the future, a process will be put in place to automatically extract the relevant data for processing, including data from other fields in the records which are currently not integrated in the text-analysis process.

Once the target data is saved into a text file, it must be preprocessed before being considered for tagging. All of the reports are recorded with all of the text in capital letters, so case differentiation does not factor into the analysis. Each line of text represents one fault description and is first split by white space. Then each token is processed individually. It is broken around symbols (except '/', which is left in the middle of a token to handle instances such as "W/IN"), with periods between or before numbers left attached to that token. Also, apostrophes before "S", "D", and "T" remain in the token. This handles cases such as “REQ’D”, “PILOT’S”, and “WON’T”. Other apostrophes are used as quotation marks and should be broken from the token. The output of this process is a list of tokens for each line, and thus a list of lists.

3.2 Part-of-Speech Tagging

The nonstandard and often ambiguous nature of the maintenance descriptions posed significant challenges for POS tagging. To address this challenge, several different types of taggers were trained and implemented in a ‘back-off’ system, in which any word not tagged by a given tagger will be sent to the back-off tagger, as illustrated in Figure 1. Any words that are not tagged by any of the taggers are ultimately tagged with the default tag. In this case, the default tag is NN (noun), as nouns are the most common part of speech for the dataset. For this tagging process, the tag set used is a modified version of the Penn Treebank Tag Set [12]. Several tags were added to tag specific, high-frequency, semantically-significant words, e.g. INSP for any form of inspection and DUE for “due”.

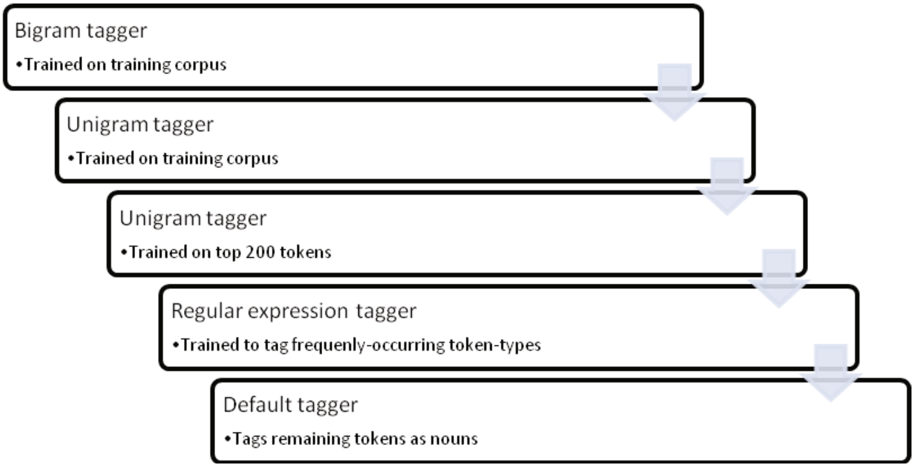

Bigram tagger
•Trained on training corpus
Unigram tagger
•Trained on training corpus
Unigram tagger
•Trained on top 200 tokens
Regular expression tagger
•Trained to tag frequenly-occurring token-types
Default tagger
•Tags remaining tokens as nouns

Fig. 1. Progression of POS tagging

A small training set of approximately 175 descriptions was manually tagged and used to train both a bigram and unigram tagger. The training set consisted of a mixture of different description types to ensure that the system was not tailored for one specific type. The bigram tagger achieved limited effectiveness given the limited training set it was provided. The training set was also used to train a Brill tagger, which did not perform as well as the step-down tagging system and was therefore not used as part of the IE program. (It is likely that both the bigram and Brill taggers would achieve significantly improved performance if trained on a much larger tagged corpus.) Those tokens not tagged with the bigram tagger are run through the unigram tagger trained on the same training set. The back-off tagger for this unigram tagger is another unigram tagger trained on a tagged set of the top 200 most frequently occurring tokens in the dataset. Because the training set and top 200 words comprise a relatively small portion of the lexicon, a regular expression tagger was installed to catch frequently-occurring types of tokens. Regular expressions were written to catch the following tokens:

- CODE: a sequence of a letter followed by numbers
- D: an eight-digit date, e.g. 20040327
- CD: all types of number formats
- LE: a single letter
- SYM: symbol other than '/', '#' or '.'
- VBN: verbs ending in '-ED'
- VBG: verbs ending in '-ING'
- CK: the word 'check' or any variation/abbreviation of it

The process of tagger development involved much trial and error, as is the case in any system development. Frequently-occurring or semantically-significant words which were incorrectly tagged were simply added to the tagged set of top words. For future research as the system is further refined, tests will be run to determine if adding a few extra POS tags will be sufficient to enhance the performance of the chunker or if it is necessary to examine the word content in order to correctly identify more detailed semantic chunks.

3.3 Chunking

Because of the specialized nature of the text domain, a layered regular expression chunking technique was chosen to provide a partial parse of the text-based maintenance entries. Again, trial and error was necessary to determine the effectiveness and success of chunking certain tag patterns. The inspection descriptions generally contain three main parts: the inspection code, a description of the inspection action or the part inspected, and some form of time indication as to when the inspection was due or took place (either date or hours or both). For this research, these were the goal chunks to be extracted. The inspection code and time information easily translates into database entries. The description text is left in its original form to ensure the generic nature of the system and make it usable for a variety of maintenance records. Future research is geared towards incorporating an ontology and knowledge base that are domain specific and will allow for the extraction of parts and actions specific to helicopter maintenance. The generic nature of the program promotes a broad application of the system in a variety of fields and provides the capability of making the system more domain-specific with the use of a domain-oriented ontology.

For the regular expression chunker, the following chunks were specified for identification:

- DATE: includes many forms that a date appears as
- CODES: includes both words tagged as codes and codes consisting of certain sequences of tags
- INSPECT: different syntactic formats that denote an inspection
- RANGE: text specifying a range of numbers

- HRS: text specifying a specific time or range of hours
- VP: any of a variety of verbs or verb phrases
- DUEPP: denotes a phrase identifying when something is due
- PP: prepositional phrase
- NNP: noun phrase
- PAREN: set of parentheses and the text inside those parentheses
- TIME: date or hours
- DESC: part of record detailing the inspection

The regular expressions written for these chunks are ordered specifically to allow the inclusion of smaller chunks in the specifications for later chunks, thus resulting in the cascading effect. Parentheses were excluded (chinked, rather than chunked) and were not considered to be part of any other chunks. Generally, information included in parentheses is not significant and need not be considered when analyzing content. While this list represents the majority of chunks needed for IE from all types of descriptions, some other chunks were included, which are necessary to adequately analyze text descriptions other than for just inspections.

3.4 Relation Extraction

Once the desired information has been chunked, relations between the entities are established to provide some context for the extracted information. For inspection descriptions, this process is rather basic and almost unnecessary, but it is modeled in the system because it will become necessary with the inclusion of other description types. For inspection descriptions, relationships exist among all three of the extracted parts of the description. Thus, three relation triples were formed:

- ([<CODE>], ‘at’, [<TIME>]) : code occurred at given time
- ([<DESC>], ‘at’, [<TIME>]) : action/part inspection occurred at given time
- ([<DESC>], ‘describes’, [<CODE>]) : action/part description describes given code

These relations represent the first step in the process of analyzing the semantic structure of the records in order to extract not only flat pieces of information, but also information that is meaningful and content related.

4 Results

4.1 Template

For this research, the IE process was based on an extraction template that represents the information that is targeted for extraction from an individual inspection description. The template consists of four fields: code, action/part, date, and time (as represented in the bottom portion of Figure 2). While this is the information generally included in an inspection description, all four pieces of information are not always included for a given report. Some records do not contain a code; others may contain a date or a time, or may lack a timestamp altogether. Not only was this template critical

Sample inspection description:

INSP A822-(PMS) INSP FOR APU ANTI-FLAIL AND #7 DRIVESHDUE AT: 20031105 OR 1590.0 HOURS FOR RHA

⇩

Code:	A822
Action/Part:	APU ANTI - FLAIL
Action/Part:	# 7 DRIVESH
Date:	20031105
Time:	1590.0

Fig. 2. Extraction template for inspection descriptions

to determine what information was to be extracted, but it is also used to structure the extracted information for insertion into the database.

4.2 Performance

The assessment of the performance of the system was broken into stages, and the POS tagger and chunker/parser were analyzed separately. The POS tagger achieved a correct tagging rate of 93.29% when run on a test set of 64 individual maintenance descriptions, which included a variety of description types. Table 1 presents a confusion matrix representing the types of mistakes the tagger made. Due to space considerations based upon the number of POS tags used, the matrix only includes the parts of the matrix that represent errors. Tags for which all instances in the test set were tagged correctly, and which were therefore not represented in the matrix, include: CC, CD, CODE, D (date), DASH (hyphen), DUE, HR, MD, NP, PAR (parentheses), PER (period), SYM, TO, and WRB. When provided with a test set comprising only inspection descriptions, the tagger achieved a performance of 96.20%, which gives evidence to the more predictable nature of these record types as compared with other types. The largest numbers of errors for both test sets occurred when a token was not tagged by any of the taggers and was given an incorrect NN tag. Many of these incorrectly-tagged tokens represent misspellings and words that require more refined preprocessing.

For comparison, the gold standard was altered to include only tags from the Penn Treebank tagset and was compared to the output of the NLTK's currently recommended pos tagger, which is a classifier-based MaxEnt tagger [13]. This type of tagger does not require training data, but rather relies on a prebuilt, feature classification model to determine part of speech of a given word [14]. The MaxEnt tagger recorded a performance of 73.18%, which gives credence to the assertion of performance improvement achieved by the developed pos-tagging system. For further proof of improvement, the test set was run through both a unigram and bigram tagger trained on the training set. By themselves, without the support of the other parts of the system, the unigram and bigram taggers achieved accuracy of 48.71% and 13.14%, respectively, which represents a significant decrease in performance from the hybrid system. The extremely poor performance of the bigram tagger is due to the small size of the

Table 1. Confusion matrix for POS tagger, displaying only those categories where errors occurred

	CK	DT	IN	INSP	JJ	LE	NN	NUM	RB	VB	VBD	VBG	VBN	VBP	VBZ
CK	[9]						2								
DT		[3]					1								
IN			[66]												
INSP				[19]			2								
JJ			1		[35]		17								
LE						[5]	1								
NN					1		[254]								
NUM								[4]							
RB					1		2	1	[6]						
VB				1			3			[5]					
VBD							1				[2]		3		
VBG							2					[4]			
VBN							1						[12]		
VBP							1						1	[1]	
VBZ							4								[2]

(rows: reference; columns: test).

training set, and the fact that it did not have a backoff tagger to resort to when it could not determine the pos of a certain word.

The chunker was evaluated using the standard metrics of precision, recall, and F-measure. Each text description in the test set was manually reviewed to determine how many chunks were identified by the system, how many actual chunks were in the record, and how many chunks were correctly identified. The system achieved a precision of 89.93% and a recall of 80.72%. The F-measure was 85.08%. These numbers demonstrate that the chunker was rather successful at identifying information needed to be extracted, but also that there remains room for improvement using other NLP techniques.

The success of the chunking technique chosen for use in this work is wholly dependent on the chunking patterns built into the system. Because of this dependency, it is not possible to compare it to another similar system without simply providing a different set of extraction patterns for the same chunker. However, this comparison would be unrealistic, as these patterns would not represent an efficient parse of the input data and would certainly perform poorly. Once other chunking systems are implemented, a comparison can be made between the different parsing techniques, but this implementation is a task that is scheduled for future work on this system.

5 Conclusions and Future Work

A critical need was identified in the engineering maintenance research domain for a system to reliably analyze text-based maintenance records in order to populate a database to facilitate data mining research efforts. To address this need, an IE tool was developed to extract meaningful chunks of information and identify the relations between extracted entities. Specifically, this research targets Army helicopter maintenance records, particularly inspection text descriptions, to support research efforts to improve CBM techniques and practices. The developed IE system was successful at identifying the inspection code, description, and time and/or date information from individual descriptions. Further research will focus on refining the system to handle other types of maintenance action descriptions, which contain more variability and ambiguity. Also, if data from other fields in the records is incorporated, the validity of the system could be more firmly established, and that data might be able to be incorporated in the IE process to improve performance.

The IE system described in this paper is just the beginning step in an endeavor to provide a reliable system for analyzing text-based engineering-domain maintenance records. A variety of future research tasks are anticipated to address specific needs of such a system. The system described in this paper will be used to tag a much larger training corpus of records to greatly reduce the manual labor required to produce such a dataset. This corpus will be used to train both Brill and bigram (and possibly trigram) POS taggers to determine if these types of taggers could prove to be more effective and efficient. A larger available training corpus will also facilitate the use of machine learning techniques to determine more detailed and refined extraction patterns. The inclusion of a domain-specific knowledge base/ontology will provide semantic content identification capabilities for the system, which, in turn, will allow for more in-depth analysis of the maintenance record content.

References

[1] Kraus, S., Blake, C., West, S.L.: Information extraction from medical notes. In: Proceedings of the 12th World Congress on Health (Medical) Informatics – Building Sustainable Health Systems (MedInfo), Brisbane, Australia, pp. 1662–1664 (in press)

[2] Hripcsak, G., Griedman, C., Alderson, P.O., DuMouchel, W., Johnson, S.B., Clayton, P.D.: Unlocking clinical data from narrative reports: A study of natural language processing. Annals of Internal Medicine 122, 681–688 (1995)

[3] Gaizauskas, R., Harkema, H., Hepple, M., Setzer, A.: Task-oriented extraction of temporal information: The case of clinical narratives. In: Thirteenth International Symposium on Termporal Representation and Reasoning (TIME'06), pp. 188–195 (2006)

[4] Irvine, A.K.: Natural language processing and temporal information extraction in emergency department triage notes. A Master's Paper (2008)

[5] Han, B., Gates, D., Levin, L.: Understanding temporal expressions in emails. In: Proceedings of the Human Language Technology Conference of the North American Chapter of the ACL, pp. 136–143 (2006)

[6] Cardie, C.: Empirical methods in information extraction. AI Magazine 18(4), 65–79 (1997)

[7] Bayoumi, A., Goodman, N., Shah, R., Eisner, L., Grant, L., Keller, J.: Conditioned-based maintenance at USC – part II: Implementation of CBM through the application of data source integration. Presented at the American Helicopter Society Specialists' Meeting on Condition Based Maintenance. Huntsville, AL (2008)

[8] Bird, S.G., Loper, E.: NLTK: The Natural Language Toolkit. In: Proceedings, 42nd Meeting of the Association for Computational Linguistics (Demonstration Track), Barcelona, Spain (2004)

[9] Madnani, N.: Getting started on natural language processing with Python. ACM Crossroads 13(4) (2007)

[10] Abney, S.: Partial parsing via finite-state cascades. Journal of Natural Language Engineering 2(4), 337–344 (1996)

[11] Cowie, J., Lehnert, W.: Information extraction. Communications of the ACM 39(1), 80–91 (1996)

[12] Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: the Penn Treebank. Computational Linguistics 19(2), 313–330 (1993)

[13] Bird, S., Klein, E., Loper, E.: Natural language processing with python: analyzing text with the Natural Language Toolkit. O'Reilly Media, Sebastopol (2009)

[14] Jurafsky, D., Martin, J.H.: Speech and language processing: an introduction to natural language processing, computational linguistics, and speech recognition. Pearson Education, Inc., Upper Saddle River (2009)

A Preliminary Study on the Selection of Generalized Instances for Imbalanced Classification

Salvador García[1], Joaquín Derrac[2], Isaac Triguero[2],
Cristóbal Carmona[1], and Francisco Herrera[2]

[1] University of Jaén, Department of Computer Science, 23071 Jaén, Spain
sglopez@ujaen.es, ccarmona@ujaen.es

[2] University of Granada, Department of Computer Science and Artificial Intelligence,
18071 Granada, Spain
jderrac@decsai.ugr.es, isaaktriguero@gmail.com, herrera@decsai.ugr.es

Abstract. Learning in imbalanced domains is one of the recent challenges in machine learning and data mining. In imbalanced classification, data sets present many examples from one class and few from the other class, and the latter class is the one which receives more interest from the point of view of learning. One of the most used techniques to deal with this problem consists in preprocessing the data previously to the learning process.

This contribution proposes a method belonging to the family of the nested generalized exemplar that accomplishes learning by storing objects in Euclidean n-space. Classification of new data is performed by computing their distance to the nearest generalized exemplar. The method is optimized by the selection of the most suitable generalized exemplars based on evolutionary algorithms. The proposal is compared with the most representative nested generalized exemplar learning approaches and the results obtained show that our evolutionary proposal outperforms them in accuracy and requires to store a lower number of generalized examples.

1 Introduction

In the last years, the class imbalance problem is one of the emergent challenges in data mining [20]. The problem appears when the data presents a class imbalance, which consists in containing many more examples of one class than the other one, being the less representative class the most interesting one [4]. Imbalance in class distribution is pervasive in a variety of real-world applications, including but not limited to telecommunications, WWW, finance, biology and medicine.

Usually, the instances are grouped into two type of classes: the majority or negative class, and the minority or positive class. The minority or positive class is often of interest and also accompanied with a higher cost of making errors. A standard classifier might ignore the importance of the minority class because its representation inside the data set is not strong enough. As a classical example, if

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 601–610, 2010.

the ratio of imbalance presented in the data is 1:100 (that is, there is one positive instance in one hundred instances), the error of ignoring this class is only 1%.

A main process in data mining is the one known as data reduction [16]. In classification, it aims to reduce the size of the training set mainly to increase the efficiency of the training phase (by removing redundant data) and even to reduce the classification error rate (by removing noisy data). Instance Selection (IS) is one of the most known data reduction techniques in data mining.

The Nested Generalized Exemplar (NGE) theory was introduced in [17] and makes several significant modifications to the exemplar-based learning model. The most important one is that it retains the notion of storing verbatim examples in memory but, it also allows examples to be generalized. In NGE theory, generalizations take the form of hyperrectangles or rules in a Euclidean n-space. The generalized examples may be nested one inside another and inner generalized examples serve as exceptions to surroundings generalizations.

Several works argue the benefits of using generalized instances together with instances to form the classification rule [19,6,15]. With respect to instance-based classification [1], the use of generalizations increases the comprehension of the data stored to perform classification of unseen data and the achievement of a substantial compression of the data, reducing the storage requirements. Considering rule induction [10], the ability of modeling decision surfaces by hybridizations between distance-based methods (Voronoi diagrams) and parallel axis separators could improve the performance of the models in domains with clusters of exemplars or exemplars strung out along a curve. In addition, NGE learning allows capture generalizations with exceptions.

Evolutionary Algorithms (EAs) [7] are general purpose search algorithms that use principles inspired by nature to evolve solutions to problems. EAs have been successfully used in data mining problems [9,14]. Their capacity of tackling IS as a combinatorial problem is especially useful [3].

In this contribution, we propose the use of EAs for generalized instances selection in imbalanced classification tasks. Our objective is to increase the accuracy of this type of representation by means of selecting the best suitable set of generalized examples to improve the classification performance for imbalanced domains. We compare our approach with the most representative models of NGE learning: BNGE [19], RISE [6] and INNER [15]. The empirical study has been contrasted via non-parametrical statistical testing [5,11,12], and the results show an improvement of accuracy whereas the number of generalized examples stored in the final subset is much lower.

The rest of this contribution is organized as follow: Section 2 reviews the preliminary theoretical study. Section 3 explains the evolutionary selection of generalized examples. Section 4 describes the experimental framework used and presents the analysis of results. Finally, in Section 5, we point out the conclusions achieved.

2 Background and Related Work

This section shows the main topics of the background in which our contribution is based. Section 2.1 describes the evaluation framework of imbalanced classification.

Section 2.2 highlights the main characteristics of NGE theory and finally, Section 2.3 shows the EAs in which our model is based.

2.1 Evaluation in Imbalanced Classification

The measures of the quality of classification are built from a confusion matrix (shown in Table 1) which records correctly and incorrectly recognized examples for each class.

Table 1. Confusion matrix for a two-class problem

	Positive Prediction	Negative Prediction
Positive Class	True Positive (TP)	False Negative (FN)
Negative Class	False Positive (FP)	True Negative (TN)

The most used empirical measure, accuracy, does not distinguish between the number of correct labels of different classes, which in the ambit of imbalanced problems may lead to erroneous conclusions. Because of this, more correct metrics are considered in imbalanced learning. Specifically, from Table 1 it is possible to obtain four metrics of performance that measure the classification quality for the positive and negative classes independently:

- **True positive rate** $TP_{rate} = \frac{TP}{TP+FN}$ is the percentage of positive cases correctly classified as belonging to the positive class.
- **True negative rate** $TN_{rate} = \frac{TN}{FP+TN}$ is the percentage of negative cases correctly classified as belonging to the negative class.
- **False positive rate** $FP_{rate} = \frac{FP}{FP+TN}$ is the percentage of negative cases misclassified as belonging to the positive class.
- **False negative rate** $FN_{rate} = \frac{FN}{TP+FN}$ is the percentage of positive cases misclassified as belonging to the negative class.

One appropriate metric that could be used to measure the performance of classification over imbalanced data sets is the Receiver Operating Characteristic (ROC) graphics [2]. In these graphics, the tradeoff between the benefits (TP_{rate}) and costs (FP_{rate}) can be visualized, and acknowledges the fact that the capacity of any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC) corresponds to the probability of correctly identifying which of the two stimuli is noise and which is signal plus noise. AUC provides a single-number summary for the performance of learning algorithms.

The way to build the ROC space is to plot on a two-dimensional chart the true positive rate (Y axis) against the false positive rate (X axis) as shown in Figure 1. The points (0, 0) and (1,1) are trivial classifiers in which the output class is always predicted as negative and positive respectively, while the point (0, 1) represents perfect classification. To compute the AUC we just need to obtain the area of the graphic as:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \tag{1}$$

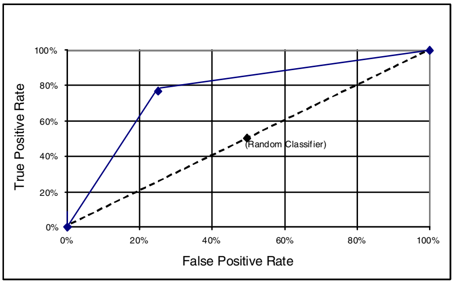


Fig. 1. Example of an ROC plot. Two classifiers are represented: the solid line is a good performing classifier whereas the dashed line represents a random classifier.

2.2 Nested Generalized Exemplar Theory

This subsection provides an overview on learning with generalized instances. First, we explain the needed concepts to understand the classification rule followed by this type of methods. After this, the three main proposals of NGE learning will be briefly described.

Matching and Classification. The matching process in one of the central features in NGE learning and it allows some customization, if desired. This process computes the distance between a new example and an generalized exemplar memory object. For remainder of this contribution, we will refer to the example to be classified as E and the generalized example stored as G, independently of G is formed by an unique instance or it has some volume.

The model computes a match score between E and G by measuring the Euclidean distance between two objects. The Euclidean distance is well-known when G is a single point. In case contrary, the distance is computed as follows (numeric attributes):

$$D_{EG} = \sqrt{\sum_{i=1}^{M} \left(\frac{dif_i}{max_i - min_i} \right)^2}$$

where

$$dif_i = \begin{cases} E_{f_i} - G_{upper} \text{ when } E_{f_i} > G_{upper} \\ G_{lower} - E_{f_i} \text{ when } E_{f_i} < G_{lower} \\ 0 \text{ otherwise} \end{cases}$$

M is the number of attributes of the data, E_{f_i} is the value of the ith feature of the example, G_{upper} and G_{lower} are the upper and lower values of G for a specific attribute and max_i and min_i are the maximum and minimum values for ith feature in training data, respectively.

The distance measured by this formula is equivalent to the length of a line dropped perpendicularly from the point E_{f_i} to the nearest surface, edge or corner of G. Note that points internal to a generalized instance have distance 0 to it. In the case of overlapping generalized examples, a point falling in the area of overlap belongs to the smaller instance, in terms of volume. The size of a hyperrectangle is defined in terms of volume. In nominal attributes, if the features are equal, the distance is zero, else it is one.

BNGE: Batch Nested Generalized Exemplar. BNGE is a batch version of the first model of NGE (also known as EACH [17]) and it is proposed to alleviate some drawbacks presented in it [19]. The generalization of examples is done by expanding its frontiers just to cover the desired example and it only merges generalized instances if the new generalized example dos not cover (or overlap with) any other stored example from any other classes. It does not permit overlapping or nesting.

RISE: Unifying Instance-Based and Rule-Based Induction. RISE [6] is an approach proposed to overcome some of the limitations of instance-based learning and rule induction by unifying the two. It follows similar guidelines explained above, but it furthermore introduces some improvements regarding distance computations and selection of the best rule using the Laplace correction used by many existing rule-induction techniques [10]).

INNER: Inflating Examples to Obtain Rules. INNER [15] starts by selecting a small random subset of examples, which are iteratively inflated in order to cover the surroundings with examples of the same class. Then, it applies a set of elastic transformations over the rules, to finally obtain a concise and accurate rule set to classify.

2.3 CHC Algorithm

We have studied its main characteristics to select it as the baseline EA which will guide the search process of our model. During each generation, the CHC algorithm [8] develops the following steps:

1. It uses a parent population of size R to generate an intermediate population of R individuals, which are randomly paired and used to generate R potential offspring.
2. Then, a survival competition is held where the best R chromosomes from the parent and offspring populations are selected to form the next generation.

CHC also implements HUX recombination operator. HUX exchanges half of the bits that differ between parents, where the bit position to be exchanged is randomly determined. It also employs a method of incest prevention: Before applying HUX to two parents, the Hamming distance between them is measured. Only those parents who differ from each other by some number of bits (mating threshold) are mated. If no offspring is inserted into the new population then the threshold is reduced.

No mutation is applied during the recombination phase. Instead, when the search stops making progress the population is reinitialized to introduce new diversity. The chromosome representing the best solution found is used as a template to re-seed the population, randomly changing 35% of the bits in the template chromosome to form each of the other chromosomes in the population.

We have selected CHC because it has been widely studied, being now a well-known algorithm on evolutionary computation. Furthermore, previous studies like [3,13] support the fact that it can perform well on data reduction problems.

3 Selection of Generalized Examples Using the Evolutionary Model CHC

The approach proposed in this contribution, named Evolutionary Generalized Instance Selection by CHC (EGIS-CHC), is explained in this section. The specific issues regarding representation and fitness function complete the description of the proposal.

Let us assume that there is a training set TR with P instances and each one of them has M input attributes. Let us also assume that there is a set of generalized instances GS with N generalized instances and each one of the N generalized instances has M conditions which can be numeric conditions, expressed in terms of minimum and maximum values in interval $[0, 1]$; or they can be categorical conditions, assuming that there are v different values for each attribute. Let $S \subseteq GS$ be the subset of selected generalized instances resulted in the run of a generalized instances selection algorithm.

Generalized instance selection can be considered as a search problem in which EAs can be applied. We take into account two important issues: the specification of the representation of the solutions and the definition of the fitness function.

- *Representation:* The search space associated is constituted by all the subsets of GS. This is accomplished by using a binary representation. A chromosome consists of N genes (one for each sample in GS) with two possible states: 0 and 1. If the gene is 1, its associated generalized example is included in the subset of GS represented by the chromosome. If it is 0, this does not occur.
- *Fitness Function:* Let S be a subset of samples of GS and be coded by a chromosome. We define a fitness function based on AUC evaluated over TR through the rule described in Section 2.2.

$$Fitness(S) = \alpha \cdot AUC + (1 - \alpha) \cdot red\_rate.$$

  AUC denotes the computation of the AUC measure from TR using S. red_rate denotes the ratio of generalized examples selected.
  The objective of the EAs is to maximize the fitness function defined. We preserve the value of $\alpha = 0.5$ used in previous works related to instance selection [3].

  The same mechanisms to perform a classification of a unseen example exposed in [17] are used in our approach. In short, they are:

  - If no rule covers the example, the class of the nearest generalized instance defines the prediction.
  - If various rules cover the example, the one with lowest volume is the chosen to predict the class, allowing exceptions within generalizations. The volume is computed following the indications given in [19].

- There is a detail not specified yet. It refers to the building of the initial set of generalized instances. In this first approach, we have used a heuristic which is fast and obtain acceptable results. The heuristic yields a generalization from each example in the training set. For each one, it finds the $K - 1$ nearest neighbours being the Kth neighbour an example of different class. Then each generalization is built getting the minimal an maximal values (in case of numerical attributes) to represent the interval in such attribute or getting all the different categorical values (in case of nominal attributes) of all the examples belonging to its set of $K - 1$ neighbours. Once all the generalizations are obtained, the duplicated ones are removed (keeping one representant in each case), hence $|GS| \leq |TR|$.

4 Experimental Framework and Results

This section describes the methodology followed in the experimental study of the generalized examples based learning approaches. We will explain the configuration of the experiment: used imbalanced data sets and parameters for the algorithms.

4.1 Experimental Framework

Performance of the algorithms is analyzed by using 18 data sets taken from the UCI Machine Learning Database Repository [18]. Multi-class data sets are modified to obtain two-class non-balanced problems, defining one class as positive and one or more classes as negative.

The data sets are sorted by their Imbalance Ratio (IR) values in an incremental way. IR is defined as the ratio between number of instances of the negative class divided by the number of instances of the positive class. Data sets considered have an IR lower than 9. The main characteristics of these data sets are summarized in Table 2. For each data set, it shows the number of examples (#Examples), number of attributes (#Attributes) and class name (minority and majority).

The data sets considered are partitioned using the *ten fold cross-validation (10-fcv)* procedure. The parameters of the used algorithms are presented in Table 3.

4.2 Results and Analysis

Table 4 shows the results in test data obtained by the algorithms compared by means of the *AUC* evaluation measure. It also depicts the number of generalized

Table 2. Summary Description for Imbalanced Data-Sets

Data-set	#Ex.	#Atts.	Class (min., maj.)	%Class(min.; maj.)	IR
Glass1	214	9	(build-win-non_float-proc; remainder)	(35.51, 64.49)	1.82
Ecoli0vs1	220	7	(im; cp)	(35.00, 65.00)	1.86
Wisconsin	683	9	(malignant; benign)	(35.00, 65.00)	1.86
Pima	768	8	(tested-positive; tested-negative)	(34.84, 66.16)	1.90
Glass0	214	9	(build-win-float-proc; remainder)	(32.71, 67.29)	2.06
Yeast1	1484	8	(nuc; remainder)	(28.91, 71.09)	2.46
Vehicle1	846	18	(Saab; remainder)	(28.37, 71.63)	2.52
Vehicle2	846	18	(Bus; remainder)	(28.37, 71.63)	2.52
Vehicle3	846	18	(Opel; remainder)	(28.37, 71.63)	2.52
Haberman	306	3	(Die; Survive)	(27.42, 73.58)	2.68
Glass0123vs456	214	9	(non-window glass; remainder)	(23.83, 76.17)	3.19
Vehicle0	846	18	(Van; remainder)	(23.64, 76.36)	3.23
Ecoli1	336	7	(im; remainder)	(22.92, 77.08)	3.36
New-thyroid2	215	5	(hypo; remainder)	(16.89, 83.11)	4.92
New-thyroid1	215	5	(hyper; remainder)	(16.28, 83.72)	5.14
Ecoli2	336	7	(pp; remainder)	(15.48, 84.52)	5.46
Glass6	214	9	(headlamps; remainder)	(13.55, 86.45)	6.38
Yeast3	1484	8	(me3; remainder)	(10.98, 89.02)	8.11

Table 3. Parameters considered for the algorithms

Algorithm	Parameters
BNGE	It has not parameters to be fixed
RISE	$Q = 1,\ S = 2$
EGIS-CHC	$Pop = 50,\ Eval = 10000,\ \alpha = 0.5$
INNER	Initial Instances= 10, MaxCycles= 5, Min Coverage= 0.95, Min Presentations= 3000, Iterations to Regularize= 50, Select Threshold= -50.0

instances maintained for each approach across all the data sets. The best case in each data set is remarked in bold.

Observing Table 4, we can make the following analysis:

- EGIS-CHC proposal obtains the best average result in AUC measure. It clearly outperforms the other techniques use in learning from generalized examples and 1NN.
- The number of generalized instances needed by EGIS-CHC to achieve such AUC rates is much lower than the needed by BNGE and RISE. In average, it also needs less generalized instances than INNER.

We have included a second type of table accomplishing a statistical comparison of methods over multiple data sets. Specifically, we have used the Wilcoxon Signed-Ranks test [5,11,12]. Table 5 collects results of applying Wilcoxon's test between our proposed methods and the rest of generalized instance learning algorithms studied in this paper over the 18 data sets considered. This table is divided into two parts: In the first part, the measure of performance used is the accuracy classification in test set through AUC. In the second part, we accomplish Wilcoxon's test by using as performance measure the number of generalized instances resulted for each approach. Each part of this table contains one column, representing our proposed methods, and N_a rows where N_a is the number of algorithms considered in this study. In each one of the cells can appear three symbols: +, = or -. They represent that the proposal outperforms (+), is similar (=) or is worse (-) in performance than the algorithm which appears in the row (Table 5). The value in brackets is the p-value obtained in the comparison and the level of significance considered is $\alpha = 0.10$.

Table 4. *AUC* in test data and number of generalized instances resulted from the run of the approaches used in this study

	AUC					number of generalized instances			
dataset	**1NN**	**BNGE**	**INNER**	**RISE**	**EGIS-CHC**	**BNGE**	**RISE**	**INNER**	**EGIS-CHC**
glass1	0.7873	0.6420	0.6659	0.6808	**0.7870**	74.40	63.80	17.30	**9.40**
ecoli0vs1	0.9630	0.9663	**0.9764**	0.9283	0.9708	10.40	53.60	5.00	**3.00**
wisconsin	0.9550	**0.9705**	0.9176	0.9351	0.9668	61.90	153.10	7.00	**2.90**
pima	0.6627	0.7099	0.6329	0.6499	**0.7223**	329.80	436.90	15.90	**14.80**
glass0	**0.8345**	0.7698	0.6579	0.7752	0.7579	67.50	69.80	34.00	**9.50**
yeast1	0.6262	0.6303	0.6467	0.6187	**0.7054**	817.00	780.30	**14.90**	16.00
vehicle1	0.6513	0.6143	0.5275	0.6499	**0.6872**	319.50	293.70	20.00	**17.20**
vehicle2	**0.9521**	0.8503	0.5293	0.9132	0.9122	155.30	133.90	26.40	**16.80**
vehicle3	0.6591	0.5559	0.5127	0.6414	**0.7119**	311.40	291.30	22.60	**17.10**
haberman	0.5618	0.5762	**0.5962**	0.5222	0.5924	211.30	132.40	17.20	**7.80**
glass0123vs456	0.9235	0.9175	0.8358	0.9199	**0.9410**	21.80	23.00	6.20	**4.60**
vehicle0	**0.9214**	0.6819	0.5342	0.8298	0.8962	214.90	166.70	37.70	**12.40**
ecoli1	0.7951	0.7983	0.8366	0.8440	**0.8759**	71.10	109.00	11.60	**5.10**
new-thyroid2	0.9819	0.9750	0.8861	0.9500	**0.9917**	12.50	26.60	3.50	**2.20**
new-thyroid1	**0.9778**	0.9208	0.9278	0.9583	**0.9778**	12.40	23.00	5.00	**2.20**
ecoli2	**0.9023**	0.8681	0.8528	0.8461	0.8821	66.60	100.20	7.00	**5.90**
glass6	0.9113	0.8613	0.7835	0.9086	**0.9534**	17.30	30.20	7.70	**3.30**
yeast3	0.8201	0.7613	0.8510	0.7588	**0.8725**	280.80	663.80	14.60	**12.10**
Average	0.8270	0.7817	0.7317	0.7961	**0.8447**	169.77	197.29	15.20	**9.02**

Table 5. Wilcoxon's test results over *AUC* and number of generalized instances resulted

	EGIS-CHC	EGIS-CHC
algorithm	*AUC*	num. gen. instances
1NN	+ (.064)	
BNGE	+ (.000)	+ (.000)
RISE	+ (.000)	+ (.000)
INNER	+ (.000)	+ (.000)

We can see that the Wilcoxon test confirms the analysis carried out above.

5 Concluding Remarks

The purpose of this contribution is to present an evolutionary model developed to tackle data reduction tasks to improve imbalanced classification based on the nested generalized example learning. The proposal performs an optimized selection of previously defined generalized examples.

The results show that the use of generalized exemplar selection based on evolutionary algorithms can obtain promising results to optimize the performance in imbalanced domains.

Acknowledgement. This work was supported by TIN2008-06681-C06-01 and TIN2008-06681-C06-02.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6(1), 37–66 (1991)
2. Bradley, A.P.: The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. Pattern Recognition 30(7), 1145–1159 (1997)

3. Cano, J.R., Herrera, F., Lozano, M.: Using evolutionary algorithms as instance selection for data reduction in KDD: An experimental study. IEEE Transactions on Evolutionary Computation 7, 561–575 (2003)
4. Chawla, N.V., Japkowicz, N., Kotcz, A.: Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations 6(1), 1–6 (2004)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
6. Domingos, P.: Unifying instance-based and rule-based induction. Machine Learning 24, 141–168 (1996)
7. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2003)
8. Eshelman, L.J.: The CHC adaptative search algorithm: How to safe search when engaging in nontraditional genetic recombination. Foundations of Genetic Algorithms, 265–283 (1991)
9. Freitas, A.A.: Data Mining and Knowledge Discovery with Evolutionary Algorithms. Springer, New York (2002)
10. Fürnkranz, J.: Separate-and-conquer rule learning. Artificial Intelligence Review 13(1), 3–54 (1999)
11. García, S., Herrera, F.: An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. Journal of Machine Learning Research 9, 2677–2694 (2008)
12. García, S., Fernandez, A., Luengo, J., Herrera, F.: A Study of Statistical Techniques and Performance Measures for Genetics-Based Machine Learning: Accuracy and Interpretability. Soft Computing 13(10), 959–977 (2009)
13. García, S., Herrera, F.: Evolutionary Under-Sampling for Classification with Imbalanced Data Sets: Proposals and Taxonomy. Evolutionary Computation 17(3), 275–306 (2009)
14. Ghosh, A., Jain, L.C.: Evolutionary Computation in Data Mining. Springer, Berlin (2005)
15. Luaces, O., Bahamonde, A.: Inflating examples to obtain rules. International Journal of Intelligent Systems 18(11), 1113–1143 (2003)
16. Pyle, D.: Data Preparation for Data Mining. The Kaufmann Series in DMS (1999)
17. Salzberg, S.: A nearest hyperrectangle learning method. Machine Learning 6, 151–276 (1991)
18. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository Irvine, CA (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
19. Wettschereck, D., Dietterich, T.G.: An experimental comparison of the nearest-neighbor and nearest-hyperrectangle algorithms. Machine Learning 19, 5–27 (1995)
20. Yang, A., Wu, X.: 10 challenging problems in data mining research. International Journal of Information Technology & Decision Making 5(4), 597–604 (2006)

Feature Selection Applied to Data from the Sloan Digital Sky Survey

Miguel Á. Montero[1], Roberto Ruíz[1],
Miguel García-Torres[1], and Luis M. Sarro[2]

[1] Área de Lenguajes y Sistemas Informáticos, Universidad Pablo de Olavide,
Ctra. de Utrera, km. 1, 41013, Sevilla, Spain
{mmontero,robertoruiz,mgarciat}@upo.es
[2] Dpt. de Inteligencia Artificial, UNED, Juan del Rosal, 16, 28040, Madrid, Spain
lsb@dia.uned.es

Abstract. In recent years there has been an explosion in the rate of acquisition of astronomical data. The analysis of astronomical data presents unprecedented opportunities and challenges for data mining in tasks, such as clustering, object discovery and classification. In this work, we address the feature selection problem in classification of photometric and spectroscopic data collected from the SDSS survey. We present a comparison of five feature selection algoritms: best first (BF), scatter search (SS), genetic algorithm (GA), best incremental ranked subset (BI) and best agglomerative ranked subset (BA). Up to now all these strategies were first applied to this paper to study relevant features in SDSS data.

Keywords: Classification, feature selection, astronomy, SDSS.

1 Introduction

The Sloan Digital Sky Survey (SDSS) [1,3,26,30] is an ambitious cosmological study of the Universe that began in 2000. Its aim is to create the greater three-dimensional cosmic map obtained so far using the SDSS 2.5 meter telescope located at the Apache Point Observatory, Sunspot, New Mexico (USA). This telescope will observe in detail a quarter of the sky and it will measure the positions and absolute brightness of hundreds of million of celestial sources, as well as distances to more than a million galaxies, and quasars. The SDSS catalogue will include both images and spectroscopy over a large area of the sky. The SDSS telescope is equipped with a large-format mosaic CCD camera to image the sky in five optical bands, and two digital spectrographs to obtain the spectra of the celestial sources selected from the imaging data. It is estimated that the SDSS project will generate 15 Terabyte of information by the time of completion.

Despite its enormous extent size, the SDSS is only one of the various surveys of tera- and petabyte size available now or in the next few years for scientific exploitation. Other surveys are the Two Micron All Sky Survey (2MASS), the Deep Near Infrared Survey (DENIS), the Double Interferometer for Visual Astrometry (DIVA), Gaia, etc.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 611–620, 2010.

Although surveys are typically high-dimensional with a very large number of samples, data volumes are becoming still larger. So it is a challenge to deal with these databases efficiently. Data mining is therefore becoming a very useful tool to automate tasks like classification, which can no longer be handled manually to a large extent. The application of a learning algorithm in this domain allows extracting knowledge from the data and to model such knowledge for decision making.

In order to identify a subset of useful features for predicting purposes, the need for feature selection techniques arises. Some benefits of feature selection are:

- A reduced volume of data facilitates the application of different data mining and search techniques. Furthermore, data mining algorithms can be executed faster with smaller datasets.
- Irrelevant and redundant features can generate less accurate and more complex models which are harder to understand.
- Knowing which data is redundant or irrelevant can be used to avoid data collection of those features in the future so that the data collection becomes more efficient and less costly.

A feature selection algorithm combines two processes: one is the search for an optimum and the other is an evaluation of sets of features. The measures obtained in the latter process will guide the search. Therefore, a feature selection algorithm is simply a search algorithm that should optimize a measure that shows how good a subset of features is (or an individual feature). There are many possible combinations of search methods and feature evaluation measures [15]. Feature selection algorithms can be grouped into two types depending on the chosen evaluation measure: the filter model evaluates features according to heuristics based on overall data characteristics whereas the wrapper uses the behaviour of a classification algorithm as a criterion for feature evaluation. The wrapper model chooses the features that show the best classification and helps to improve a learning algorithm's behaviour. The drawback is that its computational cost [13,12] is higher than the filter model.

Data mining techniques are becoming more popular as shown in recent publications that apply neural networks for spectral star classification [24], for physical measures of stellar phantoms [5], for spectral classification of galaxies [23], for morphological classification of galaxies [2,25], for discrimination of stars and galaxies in digitilized images [17], for fast estimation of cosmological parameters [4], and to distinguish quasars from stars [33], support vector machines for automatic classification [32,31], for detecting sources [19], to identify red variables [29] and to estimate redshifts [27], decision trees to build an online system for automated classification of X-ray sources [16] and for star-galaxy classification [6].

With the aim of 1) generating models capable of predicting source types and 2) selecting, analysing and interpreting a subset of useful features for classification, this preliminary work is particularly focused on helping astronomers for predicting astronomical source types. For this purpose, we apply a complete process of $\mathcal{KDD}$.

In this paper we compare five feature selection algorithms -best first (BF) [18], scatter search (SS) [7], genetic algorithm (GA) [9], best incremental ranked subset (BI) [22] and best agglomerative ranked subset (BA) [21]- using three evaluation measures -correlation-based feature subset selection (CFS) [10], consistency based measure (CNS) [14] and a wrapper based measure (CLS) [12]- in order to study their performance. We used a probabilistic learner (naive Bayes [11]) and a decision tree classifier (C4.5) [20] as learning algorithms.

This paper is organized as follows. Section 2 describes the SDSS data. Then Section 3 explains the experimentation and the results achieved. Finally the conclusions are presented in Section 4.

2 Data Description

In this work we studied three different types of sources -galaxies, stars and quasars-. To collect data, first we must define the galactic region under study, and then select the features of intereset of the sources. In order to avoid overcrowded regions of stars, we selected a field of the sky out of the galactic plane: a square region of side length 10^o, positioned in the galactic longitude $b \in [20°, 30°]$ and galactic latitude $\ell \in [190°, 200°]$.

Table 1 shows the characteristics of the field under study. The first column contains the galactic coordinates, then, the type of source followed by the number of instances associated. Finally the total number of sources used in this work is stated.

Table 1. Characteristics of the field under study

Galactic coordinates	type	# sources	# total
$b \in [20°, 30°]$, $\ell \in [190°, 200°]$	Galaxies Stars Quasars	7836 2341 626	11181

SDSS provides photometric and spectroscopic data. Photometry measures the flux of all the light that an astronomical source emits within a region of its spectral energy distribution. SDSS maps the galactic cap in five bands u, g, r, i and z from 3500 to $8900\mathring{A}$ that correspond to near ultraviolet, green, red, near-infrared and infrared regions respectively. Generally, the magnitude measurement is used instead of the flux, which refers to the logarithmic measure of the brightness of the source.

Spectroscopy decomposes the incoming light from sources into its constituent electromagnetic radiation of different wavelengths called spectra. Each spectrum consists of a collection of bright lines in a dark background or dark lines in a bright background depending whether it is an emission or absorption spectrum. These lines, called spectral lines, can be used to characterise a source type and thus be used for classification purposes.

To study the predictive power of photometric and spectroscopic data, we generated three datasets:

- Photometric dataset (Pho): it contains a total of 15 features; the 5 magnitude values (u, g, r, i and z) and 10 that correspond to colour index measures derived from magnitude ($u-g$, $u-r$, $u-i$, $u-z$, $g-r$, $g-i$, $g-z$, $r-i$, $r-z$, $i-z$).
- Spectroscopic dataset (Spe): it consist of 28 individual emission and absorption lines detected.
- Full dataset (All): it contains photometric and spectroscopic data features.

In all cases, we selected the best observation of a source with multiple observations.

3 Experimental Results

To assess model quality we apply a k-fold cross validation with $k = 10$. In order to reduce variability, we run 3 cross validations using different partitions. The validation results are averaged over the runs.

We used, as learning algorithms, two widely used classifiers: naive Bayes [11] (NB) and C4.5 [20]. Naive Bayes is a probabilistic classifier based on Bayes' theorem that assumes that feature values are independent of eachother given the class (conditional independence assumption). C4.5 is a decision tree classifier that, based on the information gain measure, builds the tree model of decisions by means of a divide and conquer strategy.

The five search algorithms presented in this work have been applied successfully to feature selection problems. Best first explores a graph by expanding the most promising node chosen according to a specified rule. Genetic algorihm and scatter search are two evolutionary population-based strategies that handle a set of solutions (each solution corresponds to a subset of features) that evolve until no improvement is reached. The main differences between both algorithms is that the former uses techniques inspired by evolutionary biology to evolve while the latter applies strategies based on intensification and diversification to achieve good solutions. BA and BI are two algorithms that carry out the search taking into account a feature ranking.

To guide the search, we considered three evaluation methods. CFS evaluates the quality of a feature subset taking into account the hypothesis that good feature subsets contain features highly correlated to the class. CNS uses the consistency measure, which estimates, for a given subset of features, the number of sources that match all but their class labels. The inconsistency rate is then used to asses its quality. Finally CLS scheme uses the error rate of the classifier as the evaluation function.

To see if the differences are statistically significant, we follow the recommendations of [8]. All the experiments were conducted using the library WEKA [28].

3.1 Feature Selection Analysis

Tables 2 and 3 show the results of the classification models NB and C4.5 respectively. In both cases, the first column reports the evaluation measure used and the dataset. Then the average accuracy achieved with the baseline classifier is given. The following columns show, for each search algorithm, the accuracy followed by the number of attributes selected. In both cases the average values over the runs is presented.

Table 2. Accuracy (acc) and the number of features selected (#) by best first (BF), scatter search (SS), genetic algorithm (GA), best incremental ranked subset (BI) and best agglomerative ranked subset (BA) with NB

		baseline	BF		SS		GA		BI		BA	
		acc	acc	#	acc	#	acc	#	acc	#	acc	#
CFS	Pho	67.55	85.55	4.17	85.55	4.17	85.20	4.60	85.48	4.13	85.58	4.13
	Spe	91.84	84.74	22.67	84.83	22.60	85.19	22.53	84.73	22.80	86.37	20.90
	All	90.67	90.97	26.07	90.98	26.03	91.22	27.47	91.17	25.00	89.87	24.20
CNS	Pho	67.55	67.97	13.83	68.11	13.80	68.19	13.83	67.70	13.97	67.97	13.83
	Spe	91.84	81.17	6.13	81.11	6.17	80.63	7.80	84.34	6.70	81.60	6.53
	All	90.67	86.81	5.57	88.38	5.63	79.73	8.97	89.22	7.97	87.36	7.03
CLS	Pho	67.55	86.12	2.47	88.17	5.30	88.11	5.33	85.51	3.10	86.65	4.13
	Spe	91.84	92.54	20.13	92.11	18.13	92.40	19.90	91.35	16.07	92.03	16.20
	All	90.67	95.25	19.63	94.81	17.23	94.81	20.57	94.19	20.73	95.11	20.30

For NB, the use of all photometric data achieves very poor results. However feature selection improves the results significantly when using CFS and CLS. Spectroscopic dataset achieves better results without feature selection. Finally combining photometrical and spectroscopical features present no improvement in accuracy with respect to the previous case. As we can see, consistency is the evaluation method which presents the poorest results. All algorithms under study show similar results, both in reduction size and features found. The analysis of the accuracy finds no statistically significant differences.

C4.5 achieves better results than NB, as is shown in Table 3. The baseline classifier displays similar behaviour to NB when comparing the accuracy achieved by the three datasets. In this case CNS doesn't degrade the accuracy. The robustness of the results with C4.5 could be due to embedded feature selection performed by the classifier. No statistically significant differences are found when comparing the accuracy.

3.2 Frequency Analysis

Table 4 presents the features selected, in the full dataset, by each algorithm using CFS as the evaluation measure. Each column shows, in descending order, the features with the highest frequency values. As we can see, all strategies select both photometric and spectroscopic features. Of the five magnitudes, u seems to have less discriminative power because it has only been selected by

Table 3. Accuracy and number of features selected with C4.5. n/a denotes not available.

		baseline	BF		SS		GA		BI		BA	
		acc	acc	#	acc	#	acc	#	acc	#	acc	#
CFS	Pho	94.22	94.08	4.17	94.08	4.17	94.17	4.60	94.03	4.13	94.04	4.13
	Spe	97.42	97.37	22.67	97.36	22.60	97.39	22.53	97.37	22.80	97.32	20.90
	All	97.46	97.85	26.07	97.83	26.03	97.36	27.47	97.87	25.00	97.36	24.20
CNS	Pho	94.22	94.29	13.83	94.29	13.80	94.28	13.83	94.28	13.97	94.29	13.83
	Spe	97.42	97.61	6.13	97.49	6.17	96.09	7.80	97.69	6.70	97.95	6.53
	All	97.46	97.90	5.57	97.69	5.63	97.36	8.97	97.41	7.97	97.55	7.03
CLS	Pho	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a	n/a
	Spe	97.42	72.54	11.53	n/a	n/a	81.04	16.17	94.24	10.83	94.24	9.83
	All	97.46	97.98	10.20	n/a	n/a	97.57	22.97	97.69	12.37	98.09	7.43

Table 4. Rank of features selected by each search algorithm, in descending order of frequency. For each spectral line, the wavelength, in angstroms, is given in brackets. Finally **a** denotes absorption line.

BF	SS	GA	BI	BA
z	z	uz	z	uz
gr	gr	gr	gr	gr
gz	gz	gz	gz	iz
iz	iz	rz	iz	H_γ (4341.68)
HeI (3889)	HeI (3889)	iz	HeI (3889)	OIII (4364.436)
H_δ (4102.89)	H_γ (4341.68)	OII (3727.092)	H_δ (4102.89)	H_β (4862.68)
H_γ (4341.68)	OIII (4364.436)	HeI (3889)	H_γ (4341.68)	OIII (4932.603)
OIII (4364.436)	H_β (4862.68)	H_δ (4102.89)	OIII (4364.436)	OIII (4960.295)
H_β (4862.68)	OIII (4932.603)	H_γ (4341.68)	H_β (4862.68)	OIII (5008.240)
OIII (4932.603)	OIII (4960.295)	OIII (4364.436)	OIII (4932.603)	OI (6365.536)
OIII (4960.295)	OIII (5008.240)	H_β (4862.68)	OIII (4960.295)	NI (6529.03)
OIII (5008.240)	OI (6365.536)	OIII (4932.603)	OIII (5008.240)	NII (6549.86)
OI (6365.536)	NI (6529.03)	OIII (4960.295)	OI (6365.536)	$H\alpha$ (6564.61)
NI (6529.03)	NII (6549.86)	OIII (5008.240)	NI (6529.03)	NII (6585.27)
NII (6549.86)	H_α (6564.61)	OI (6302.046)	NII (6549.86)	SII (6718.29)
H_α (6564.61)	NII (6585.27)	OI (6365.536)	$H\alpha$ (6564.61)	SII (6732.67)
NII (6585.27)	SII (6718.29)	NI (6529.03)	NII (6585.27)	K (3934.777 **a**)
SII (6718.29)	SII (6732.67)	NII (6549.86)	SII (6718.29)	H (3969.588 **a**)
SII (6732.67)	K (3934.777 **a**)	H_α (6564.61)	SII (6732.67)	G (4305.61 **a**)
K (3934.777 **a**)	H (3969.588 **a**)	NII (6585.27)	K (3934.777 **a**)	Mg (5176.7 **a**)
H (3969.588 **a**)	G (4305.61 **a**)	SII (6718.29)	H (3969.588 **a**)	Na (5895.6 **a**)
G (4305.61 **a**)	Mg (5176.7 **a**)	SII (6732.67)	G (4305.61 **a**)	CaII (8500.36 **a**)
Mg (5176.7 **a**)	Na (5895.6 **a**)	K (3934.777 **a**)	Mg (5176.7 **a**)	CaII (8544.44 **a**)
Na (5895.6 **a**)	CaII (8500.36 **a**)	H (3969.588 **a**)	Na (5895.6 **a**)	
CaII (8500.36 **a**)	CaII (8544.44 **a**)	G (4305.61 **a**)	CaII (8500.36 **a**)	
CaII (8544.44 **a**)		Mg (5176.7 **a**)	CaII (8544.44 **a**)	
		Na (5895.6 **a**)	CaII (8664.52 **a**)	
		CaII (8500.36 **a**)		
		CaII (8544.44 **a**)		
		CaII (8664.52 **a**)		

Table 5. Ranking of the most selected features of the dataset All in descending order

feature	λ (Å)	type	frequency (%)
iz	–	–	100
H_γ	4341.68	**e**	100
OIII	4364.436	**e**	100
H_β	4862.68	**e**	100
OIII	4932.603	**e**	100
OIII	4960.295	**e**	100
OIII	5008.240	**e**	100
OI	6302.046	**e**	100
OI	6365.536	**e**	100
NI	6529.03	**e**	100
NII	6549.86	**e**	100
NII	6585.27	**e**	100
SII	6718.29	**e**	100
SII	6732.67	**e**	100
K	3934.777	**a**	100
H	3969.588	**a**	100
Mg	5176.7	**a**	100
Na	5895.6	**a**	100
CaII	8500.36	**a**	100
OII	3727.092	**e**	96
OII	3729.875	**e**	96
G	4305.61	**a**	96
H_α	6564.61	**e**	92
H_δ	4102.89	**e**	86
gz	–	–	70
gr	–	–	68
CaII	8544.44	**a**	62
i	–	–	56

GA (colour ug). Most spectral lines found by the strategies match as is shown. Results obtained with the evaluation measures CNS and CLS are omitted due to lack of space.

To study the usefulness of photometrical and spectroscopical features, we give the occurrence level of the features found by all algorithms on full dataset in Table 5. First column corresponds to the feature selected. Then the wavelength in angstrom is given, followed by the type of line (*a* for absorption line and *e* emission line). Finally the frequency value is shown in the last column. As we can see, colour iz seems to be the most discriminative photometrical feature since it reaches a frequency of 100%. The other features with the same occurrence level correspond to emission lines.

4 Conclusions

In this study we have analysed the prediction power of photometric and spectroscopic data from the SDSS survey. To address this problem we applied a total of 15 feature selection strategies (search algorithm + an evaluation measure) that have proved to be competitive in this type of problem.

In this experiment all strategies studied presented very similar results in both accuracy and reduction. These results are perfectly understandable from the astrophysical point of view since they basically consist of the strongest emission lines in the spectra. The fact that these are mainly (or even exclusively) extragalactic emission lines (not expected in stellar spectra) reflects the fact that colour indices (subtraction of filter magnitudes such as *iz*, *gz* or *gr*) seem to suffice for the star/galaxy separation. Confusion probably arises in the cool end of the stellar Main Sequence (the cooler stars of spectral type M or later) but this can easily be resolved by the absence of most of the spectral lines listed in table 5 in stellar spectra (except for H_α and CaII in magnetically active stars).

Further analysis is needed, in particular for exploration of the extended parameter space that includes line ratios between emitted fluxes in different spectral lines.

Acknowledgements

This work has been partially supported by the project TIN2008-68084-C02-00 (Ministerio de Educación y Ciencia of Spain). Part of the computer time was provided by the Centro Informático Científico de Andalucía (CIC).

References

1. Abazajian, K., et al.: The third data release of the Sloan Digital Sky Survey. The Astronomical Journal 129, 1755–1759 (2005)
2. Adams, A., Woolley, A.: Hubble classification of galaxies using neural networks. Vistas in Astronomy 38(3), 273–280 (1994)
3. Adelman-McCarthy, J.K., et al.: The fourth data release of the Sloan Digital Sky Survey. The Astrophysical Journal Supplement Series 162(1), 38–48 (2006)
4. Auld, T., Bridges, M., Hobson, M.P., Gull, S.F.: Fast cosmological parameter estimation using neural networks. Monthly Notices of the Royal Astronomical Society 376(1), L11–L15 (2007)
5. Bailer-Jones, C., Irwin, M., Gilmore, G., von Hippel, T.: Physical parametrization of stellar spectra: the neural network approach. Monthly Notices of the Royal Astronomical Society 292, 157–166 (1997)
6. Ball, N.M., Brunner, R.J., Myers, A.D.: Robust machine learning applied to astronomical data sets. I. star-galaxy classification of the Sloan Digital Sky Survey DR3 using decision trees. The Astrophysical Journal 650, 497–509 (2006)
7. García, F., García-Torres, M., Melián, B., Moreno-Pérez, J.A., Moreno-Vega, J.M.: Solving feature subset selection problem by a parallel scatter search. European Journal of Operational Research 169(2), 477–489 (2006)

8. García, S., Herrera, F.: An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. Journal of Machine Learning Research 9, 2677–2694 (2008)
9. Goldberg, D.E.: Genetic Algorithms for Search Optimization and Machine Learning. Addison-Wesley, Reading (1989)
10. Hall, M.A.: Correlation-based feature subset selection for machine learning. PhD thesis, University of Waikato, Hamilton, New Zealand (1998)
11. John, G.H., Langley, P.: Estimating continuous distributions in bayesian classifiers. In: Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence, pp. 338–345 (1995)
12. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
13. Langley, P.: Selection of relevant features in machine learning. In: Proceedings of the AAAI Fall Symposium on Relevance, pp. 140–144 (1994)
14. Liu, H., Setiono, R.: A probabilistic approach to feature selection: a filter solution. In: Proceedings of the 13th International Conference on Machine Learning, pp. 319–327. Morgan Kaufmann, San Francisco (1996)
15. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17(3), 1–12 (2005)
16. McGlynn, T.A., Suchkov, A.A., Winter, E.L., Hanisch, R.J., White, R.L., Ochsenbein, F., Derriere, S., Voges, W., Corcoran, M.F., Drake, S.A., Donahue, M.: Automated classification of ROSAT sources using heterogeneous multiwavelength source catalogs. The Astrophysical Journal 616, 1284–1300 (2004)
17. Odewahn, S.C., Nielsen, M.L.: Star-galaxy separation using neural networks. Vistas in Astronomy 38, 281–286 (1994)
18. Pearl, J.: Heuristics: intelligent search strategies for computer problem solving. Addison-Wesley, Reading (1984)
19. Qu, M., Shih, F.Y., Jing, J., Wang, H.: Automatic solar flare detection using MLP, RBF, and SVM. Solar Physics 217(1), 157–172 (2003)
20. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
21. Ruiz, R., Aguilar-Ruiz, J.S., Riquelme, J.C.: Best Agglomerative Ranked Subset for Feature Selection. In: JMLR Workshop and Conference Proceedings. New challenges for feature selection in data mining and knowledge discovery, vol. 4, pp. 148–162 (2008)
22. Ruiz, R., Riquelme, J.C., Aguilar-Ruiz, J.S.: Incremental wrapper-based gene selection from microarray expression data for cancer classification. Pattern Recognition 39, 2383–2392 (2006)
23. Sodré, L., Cuevas, H.: Spectral classification of galaxies. Vistas in Astronomy 38, 287–291 (1994)
24. Storrie-Lombardi, M.C., Irwin, M.J., von Hippel, T., Storrie-Lombardi, L.J.: Spectral classification with principal component analysis and artificial neural networks. Vistas in Astronomy 38(3), 331–340 (1994)
25. Storrie-Lombardi, M.C., Lahav, O., Sodr, L., Storrie-Lombardi, L.J.: Morphological classification of galaxies by artificial neural networks. Monthly Notices of the Royal Astronomical Society 259, 8–12 (1992)
26. Stoughton, C., et al.: Sloan Digital Sky Survey: Early Data Release. The Astronomical Journal 123, 485–548 (2002)
27. Wadadekar, Y.: Estimating photometric redshifts using support vector machines. Publications of the Astronomical Society of the Pacific 117(827), 79–85 (2005)

28. Witten, I.H., Frank, E.: Data mining: practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)
29. Wozniak, P.R., Williams, S.J., Vestrand, W.T., Gupta, V.: Identifying Red Variables in the Northern Sky Variability Survey. The Astronomical Journal 128(6), 2965–2976 (2004)
30. York, D.G., Adelman, J., Anderson, J.E., Anderson, S.F., et al.: The Sloan Digital Sky Survey technical summary. The Astronomical Journal 120, 1579–1587 (2000)
31. Zhang, S., Zhang, C., Yang, Q.: Data preparation for data mining. Applied Artificial Intelligence 17(5-6), 375–381 (2003)
32. Zhang, Y., Zhao, Y.: Automated clustering algorithms for classification of astronomical objects. Astronomy & Astrophysics 422(3), 1113–1121 (2004)
33. Zhang, Y., Zhao, Y.: A Comparison of BBN, ADTree and MLP in separating quasars from large survey catalogues. Chinese Journal of Astronomy and Astrophysics 7(2), 289–296 (2007)

FastXplain: Conflict Detection for Constraint-Based Recommendation Problems

Monika Schubert, Alexander Felfernig, and Monika Mandl

Applied Software Engineering, IST
{monika.schubert,alexander.felfernig,monika.mandl}@ist.tugraz.at
Graz University of Technology,
Inffeldgasse 16b/II, 8010 Graz, Austria

Abstract. Constraint-based recommender systems support users in the identification of interesting items from large and potentially complex assortments. Within the scope of such a preference construction process, users are repeatedly defining and revising their requirements. As a consequence situations occur where none of the items completely fulfills the set of requirements and the question has to be answered which is the minimal set of requirements that has to be changed in order to be able to find a recommendation. The identification of such minimal sets relies heavily on the identification of minimal conflict sets. Existing conflict detection algorithms are not exploiting the basic structural properties of constraint-based recommendation problems. In this paper we introduce the FastXplain conflict detection algorithm which shows a significantly better performance compared to existing conflict detection algorithms. In order to demonstrate the applicability of our algorithm we report the results of a corresponding performance evaluation.

Keywords: Constraint-based Recommender Systems, Conflict Detection, Automated Query Adaptation.

1 Introduction

Recommender systems are interactive software applications that support a user in a personalized way in finding interesting items from a large range of products. This decision support is necessary due to a large size and complexity of products and service assortments. A constraint-based recommender supports the identification of items based on a given set of requirements, knowledge about the item assortment and knowledge about which items should be selected in which context [1]. Requirements as well as knowledge about the item assortment, and selection knowledge are represented in the form of constraints. The first step of a constraint-based recommender is to collect the user requirements. Products (for example different types of computers, mobile phones or financial services) are usually stored in a database. A constraint-based recommender normally recommends products that fulfill all user requirements. In order to retrieve these products from the product table a conjunctive query based on the requirements

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 621–630, 2010.

is created. This query is an *AND*-composition of all constraints. If a query does not return any products, the system needs to calculate explanations [2,3,6] that indicate minimal sets of changes to the user such that an item can be suggested. Existing approaches to handle these changes focus on maximal succeeding sub queries [6] or on low-cardinality diagnoses [2]. Approaches to calculate these sub-queries or diagnoses are based on the identification of minimal conflict sets [5]. In this paper we present a novel approach for an efficient identification of minimal conflict sets - a set of constraints that need to be changed in order to adapt a set of unsatisfyable requirements such that a solution can be found.

The remainder of this paper is organized as follows: First we introduce a motivating example from the domain of mobile phones. This example will be used throughout the paper to explain the algorithm described in Section 3. The results of evaluations including a comparison to the state-of-the-art algorithm QuickXplain [5] are presented in Section 4. In Section 5 we discuss related work and with Section 6 we conclude the paper.

Table 1. Working example from the domain of mobile phones

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
price	149	121	130	169	99	199	219	129	178	163
talktime	420	460	410	480	500	510	400	390	440	430
size	M	L	S	L	M	M	S	S	S	S
color	white	black	blue	white	black	gray	white	pink	white	white
weight	155	92	158	165	178	139	115	101	98	169
camera	yes	yes	yes	no	yes	no	yes	yes	no	no

2 Working Example

The example introduced in this section will be used for further explanations throughout this paper. In this example the user specifies 6 requirements for selecting a mobile phone using a constraint-based recommender. These user requirements are: c_1*: price < 150,* c_2*: talktime > 450,* c_3*: size = small,* c_4*: color = white,* c_5*: weight < 100* and c_6*: camera included.* Table 1 shows all mobile phones (specified as items P_1, ..., P_{10}) of the assortment together with the attributes that can be specified by the user. A recommendation can be identified by generating a conjunctive query based on the user requirements. For our sample requirements this query Q would be: *SELECT * FROM phones WHERE price < 150 AND talktime > 450 AND size = 'S' AND color = 'white' AND weight < 100 AND camera-included = 'yes'.* When executing this query to Table 1 no product is returned because no mobile phone satisfies all user requirements (constraints).

3 The FastXplain Algorithm

In this section we are going to explain the FastXplain algorithm that can be used to calculate minimal conflict sets [5], i.e., minimal sets of changes to the given set of requirements (constraints) that can guarantee the retrieval of a solution. Compared to existing conflict detection algorithms [5] the FastXplain algorithm takes the basic structural properties of constraint-based recommendation problems into account. These structural properties comprehend a table representation of constraints (requirements) and items (products).

On a more formal level our approach can be interpreted as a specific type of constraint satisfaction problem (CSP). This CSP consists of a set of constraints C $= \{c_1, c_2, ..., c_n\}$ and a set of items $P = \{p_1, p_2, ..., p_n\}$. Each item $p_i \in P$ has a set of possible values D_i from the domain. A CSP like this has a solution (is satisfiable) if at least one item from P exists that fulfills all constraints in C, otherwise it is over-constrained. An over-constrained CSP has no solution, i.e. there exists no item in P that fulfills all constraints in C. A recommendation based on a satisfiable CSP can be calculated directly, because every item that fulfills all user requirements can be recommended. But for over-constrained CSPs we need to calculate minimal conflicts sets to suggest alternatives to the user [3]. A conflict set is defined as a subset $CS = \{c_1, c_2, ..., c_m\}$ of the given user requirements such that none of the items in P satisfies CS. CS is minimal if there exists no conflict set which is a subset of CS.

Table 2. Table of products and constraints with satisfaction values

	P_1	P_2	P_3	P_4	P_5	P_6	P_7	P_8	P_9	P_{10}
$c_1(price < 150)$	1	1	1	0	1	0	0	1	0	0
$c_2(talktime > 450)$	0	1	0	1	1	1	0	0	0	0
$c_3(small)$	0	0	1	0	0	0	1	1	1	1
$c_4(white)$	1	0	0	1	0	0	1	0	1	1
$c_5(weight < 100)$	0	1	0	0	0	0	0	0	1	0
$c_6(camera\ included)$	1	1	1	0	1	0	1	1	0	0

3.1 Identification of Minimal Conflict Sets

The underlying data structure of our algorithm is a table representation of the relation between the constraints and the items (see Table 2). If a constraint is satisfied by an item, a 1 (true) is stored in the table. If the item can not fulfill the requirement (constraint) a 0 (false) is stored in the table. Table 2 shows the reduced table for our working example. This table must be recalculated when the requirements are changing and it can be smaller than the product table if not all attributes of the products are specified by the user.

Based on this representation all diagnoses can be directly extracted [4]: for every column (product) take every constraint where the product does not satisfy the constraint (a zero is in the table). This set of constraints (diagnosis) provides a relaxation to the query. The diagnosis retrieved of product P_1 is $\{c_2, c_3, c_5\}$. If we remove these constraints from the query Q and execute the query _Q’: SELECT * FROM phones WHERE price < 150 AND color = ’white’ AND camera-included = ’yes’_ on the database, $\{P_1\}$ is returned as recommendation.

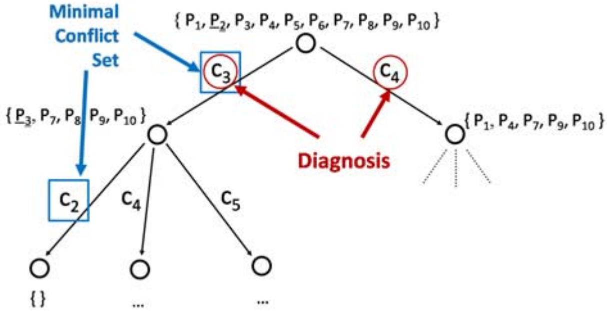

Minimal Conflict Set
$\{ P_1, \underline{P_2}, P_3, P_4, P_5, P_6, P_7, P_8, P_9, P_{10} \}$
c_3
c_4
$\{ \underline{P_3}, P_7, P_8, P_9, P_{10} \}$
$\{ P_1, P_4, P_7, P_9, P_{10} \}$
Diagnosis
c_2
c_4
c_5
{}
...
...

Fig. 1. Directed acyclic graph built of diagnoses to identify one minimal conflict set

The drawback of this approach is that for every product one diagnosis can be calculated. Thus the amount of diagnoses is the same as the amount of products which is generally large. Besides that, most of these diagnoses are not minimal. A diagnosis D is minimal if there exists no subset D' of D such that D' is a diagnosis. Only minimal diagnoses lead the user to a *minimal* set of changes of his original requirements, which makes them especially interesting. The performance of the calculation of minimal diagnoses [7] is highly dependent on the performance of the algorithm to identify minimal conflict sets. In order to calculate these minimal conflict sets, the FastXplain exploits diagnoses from the product assortment and takes the one with the lowest cardinality (lowest number of constraints in the diagnosis). In our working example this would be the diagnosis $\{c_3, c_4\}$ extracted from product P_2. Based on this diagnosis we can built a directed acyclic graph, where the nodes are sets of products and the edges are labeled with constraints (the graph of the example is shown in Figure 1). The root node consists of all products from the product table. After adding the edges labeled with the elements of the diagnosis to the root node we calculate the product set for the children. Consequently, we exclude all products from the assortment which do not satisfy the constraint. For the constraint c_3 only the products $\{P_1, P_7, P_8, P_9, P_{10}\}$ fulfill this constraint. We also identify all the products for the second branch in the tree. This tree is shown in Figure 1.

In the next step we start with the reduced set of products of the first child (in the example c_3). The lowest cardinality diagnoses of the remaining products set $\{P_3, P_7, P_8, P_9, P_{10}\}$ are $\{c_2, c_4, c_5\}$, $\{c_1, c_2, c_5\}$ and $\{c_1, c_2, c_6\}$. Let us assume that the diagnosis $\{c_2, c_4, c_5\}$ is the first one retrieved. We add all constraints of this diagnosis to the tree. For every leaf in the tree we identify the remaining products of the set. When calculating the remaining products for the branch $\{c_3, c_2\}$ no product is left. This means the path $\{c_3, c_2\}$ is a minimal set. We ensure the minimality of the set by performing a breadth-first search.

In order to calculate all minimal conflict sets, this method has to be continued until each leaf has either an empty set of remaining products or the path to the leaf is not minimal (the set of constraints of the path is a superset of another minimal conflict set). If we expand the tree further to calculate all minimal conflict sets, we would get the following minimal conflict sets as a result: $MCS = \{c_2, c_3\}$, $\{c_1, c_2, c_4\}$, $\{c_1, c_3, c_4\}$, $\{c_1, c_3, c_5\}$, $\{c_2, c_4, c_5\}$, $\{c_2, c_4, c_6\}$, $\{c_3, c_5, c_6\}$, $\{c_4, c_5, c_6\}$.

3.2 Algorithm

In this section we dive into a more formal description of FastXplain. FastXplain was inspired by the HSDAG [7] thus we keep the same level of description. The input values for the FastXplain are a root node for the tree and the product table (like in Table 2). The root node is the main reference to the resulting tree which holds all information. The result of the algorithm (all minimal conflict sets) is stored in one global variable MCS, which is empty at the beginning.

Algorithm 1. FastXplain(root, p)

```
{Input: p - table of constraints and products}
{Input: root - the root node of the resulting tree}
{Global: MCS - set of all minimal conflict sets}
$d \leftarrow getMinCardinalityDiagnosis(p)$
for all $constraint\ c\ from\ d$ do
    $p' \leftarrow reduce(c, p)$
    $child \leftarrow addChild(c, p')$
    if $p' = \{\}$ then
        $child \leftarrow ok$
        if path(child) $\notin$ MCS then
            $MCS \leftarrow path(child)$
            $return$
        end if
    end if
    if $\exists\ cs \in MCS : cs \subseteq MCS \subseteq path(child)$ then
        $child \leftarrow closed$
    end if
    if child $\neq closed$ then
        $FastXplain(child, p')$
    end if
end for
```

In the first step the algorithm calculates the diagnosis with the lowest cardinality (*getMinCardinalityDiagnosis*) i.e. the diagnosis consisting of the lowest number of constraints. In the example tree of Figure 1 this set is $\{c_3, c_4\}$. If there exists more than one diagnosis with the same cardinality the first one is chosen. For all constraints of this diagnosis a set of products (P') containing all products that satisfy this constraint is calculated (*reduced*). In our working example this is the set $\{P_3, P_7, P_8, P_9, P_{10}\}$ for the constraint c_3 and for the constraint c_4 it is $\{P_1, P_4, P_7, P_9, P_{10}\}$. Based on this reduced set of products combined with the constraint c_i a node in the tree is created and added to the root node (*addChild*). If the set of products is empty a minimal conflict set (path from the root node of the tree to the child) is found. The minimality is ensured by the breadth-first search of the algorithm. If the path to the child is a minimal conflict set, the child is marked as *'ok'* and no further investigation is needed for this leaf. If this path it is not an element of MCS (set of all minimal conflict sets) yet, then it is added.

If the path or a subset of it is already a minimal conflict set, then there is no need to expand the node anymore, because the conflict set found would not be minimal. In this situation the child is marked as *'closed'*. In all other cases the tree is constructed further in breadth-first manner. In the example this expansion would be the node c_4 on the first level. When calculating all minimal conflicts sets a tree is created where all paths to the leafs marked with *ok* are minimal conflict sets. Otherwise (for non-minimal conflict sets) the leafs are marked with *closed*.

4 Empirical Evaluation

In this section we discuss the performance of the FastXplain algorithm in different settings with different characteristics. In order to analyze the runtime we compared the algorithms FastXplain and QuickXplain [5]. Both algorithms are implemented in Java 1.6 and all experiments were performed on a normal desktop PC. The QuickXplain [5] algorithm calculates one minimal conflict set at a time. In order to calculate all minimal conflict sets using the QuickXplain we combined it with HSDAG (Hitting Set Directed Acyclic Graph) of Reiter [7].

The crucial point of the performance of the QuickXplain [5] is the consistency checking. One possibility to check the consistency is to use a theorem prover. Another one is to use a database when operating on a product table. If no product is returned, then the consistency checking failed. Another possibility for this calculation is to create a table consisting of the constraints and products (see Table 2). From this table it can be determined if at least one product of the assortment fulfills all constraints of the set that need to be checked. A product fulfills the current constraints if for this product (column) all selected rows (depending on the set of constraints) is set to 1.

In our evaluation we wanted to compare the FastXplain with the version of the QuickXplain which has the best performance according to the structural properties (product table) of recommender systems. We evaluated all versions (usage of a theorem prover, consistency checking through the database and using the table datastructure) for different settings and the results have shown clearly that the version with the consistency checking using the table datastructure containing the constraints and items is the one with the best performance.

4.1 Satisfaction Rate

The satisfaction rate indicates how many constraints are satisfied by all items (a satisfaction rate of 10% means that with a probability of 10% an item satisfies the given user requirement). The number and size of minimal conflict sets is highly dependent on the satisfaction rate. A low satisfaction rate results in a

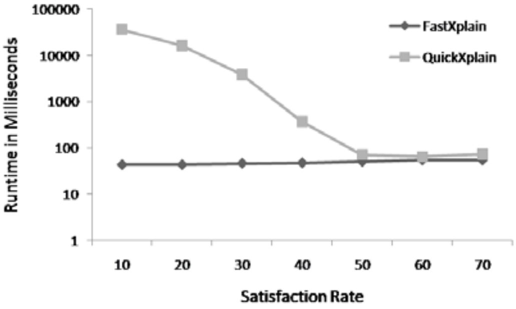


Fig. 2. Runtime comparison of the algorithms FastXplain and QuickXplain to calculate all MCSs for different satisfaction rates

high number of minimal conflict sets with a low cardinality. The performance on a wide range of satisfaction rates is critical for a universal usage of the algorithm for different applications.

For comparing the two algorithms (FastXplain and QuickXplain) we used a typical setting of the field of recommender systems with 2000 products and 10 user requirements. We created 7 different settings with an increasing satisfaction rate r between 10% and 70%. For a satisfaction rate of 0% every constraint would be a minimal conflict set and for a satisfaction rate higher than 70% it would be too difficult to find a setting with at least one minimal conflict set.

In the first test run we compared the average runtime of 30 different settings for each satisfaction rate of the algorithms for calculating *one minimal conflict set*. Both algorithms performed well on this test over the whole spectrum of satisfaction rates. The median runtime of all satisfaction rates for the FastXplain is 46 ms with a standard deviation of 4.47. This result is similar to the average runtime of the QuickXplain which is 48 ms with a standard deviation of 5.33.

In the second test run we compared the runtime of both algorithms to calculate *all minimal conflict sets* for each satisfaction rate. We tested every configuration (2000 items, 10 constraints, different satisfaction rates) with 30 different settings. The average runtime of every satisfaction rate is shown in Figure 2. We can see that the QuickXplain is quite slow when calculating all minimal conflict sets for a low satisfaction rate. This is based on the high number of minimal conflict sets that are needed. With a satisfaction rate higher than 40% the QuickXplain needs about the same runtime for the calculation as the FastXplain. In comparison to this the FastXplain needs nearly the same runtime over all satisfactions rates (mean: 47 ms, standard deviation: 4.67) which makes it more independent of the domain and the user requirements. In a recommender system all different kinds can occur but a satisfaction rate between 40% and 50% is more likely.

4.2 Number of Items

One characteristic to distinguish applications in the field of recommender systems is based on the different size of items. The typical number of items comes from the domain of the recommender system. To study if the algorithm FastXplain is feasible for a large spectrum of applications we compared the runtime to the one of the QuickXplain for 10 user requirements, an increasing amount of items and a satisfaction rate of 50%.

In order to evaluate the performance we compared the run time of both algorithms calculating one minimal conflict set. This conflict set consisting of a subset of user requirements can be used to help the user finding a solution for an inconsistent setting. The usability of a system depends on the performance of the underlying algorithm to help the user finding a solution. Thus a fast computation of at least one minimal conflict set is important. In the left part of Figure 3 shows the average runtime of both algorithms for calculating one minimal conflict set for 1000, 2000, 4000 and 8000 items (30 runs each). While both algorithms calculate equally fast a solution for 1000 and 2000 items, we can see that the FastXplain performs better for a higher number of items.

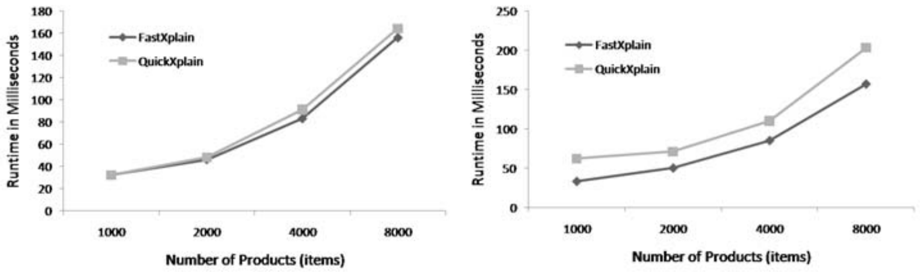


Fig. 3. Runtime comparison of the algorithms FastXplain and QuickXplain to calculate one (left) and all (right) MCSs (10 constraints, satisfaction rate: 50%)

Sometimes it is required to calculate all minimal conflict sets for offering the best option for the user. Therefore we studied the runtime of the algorithms for calculating all minimal conflict sets. From the comparison (right part of Figure 3) we can see that the FastXplain is faster than the QuickXplain. Summarizing we can say that both algorithms are suited for calculating one minimal conflict set independent on the amount of items. But when it comes to the calculation of all minimal conflict sets the FastXplain performs better.

4.3 Different Number of MCS

In some applications it is neither needed to calculate only one minimal conflict set nor is it suited to calculate all minimal conflict sets - just a few are enough. Thus we conducted a study using 2000 items and 10 user requirements. The runtime of the QuickXplain algorithm increases with the number of minimal conflict sets. This is based on the consistency checks for the calculations. In comparison to this the runtime of FastXplain is nearly independent (only very slightly increasing) of the number of minimal conflict sets. Thus no matter if only one or a few minimal conflict sets are needed, the FastXplain is suited for all settings. If we compare the values of the runtime of the FastXplain of the left amd right part of Figure 3 we can see that the values are about the same (standard deviation: 1.4 ms) for calculating one minimal conflict set versus calculating all minimal conflict sets. The increase of the runtime of the QuickXplain for calculating one minimal conflict set to calculate all minimal conflict sets for 8000 items is 20%. Summarizing we can say that the FastXplain can be used to calculate one minimal conflict set as well as to calculate all minimal conflict sets. Both types of calculations are used in constraint-based recommender systems.

5 Related Work

In the context of configuration problems [2] have developed concepts for identifying inconsistent user requirements. The idea was to determine minimal cardinality sets of requirements that need to be changed in order to be able to find a solution. The calculation of these minimal cardinality sets was based on the concepts of Model-Based Diagnoses (MBD) [7]. The conflicts exploited in [2] are not necessarily minimal which increases the runtime due to a larger Hitting Set Acyclic Directed Graph (HSDAG). An efficient algorithm to identify minimal conflict sets is the QuickXplain introduced by [5]. This approach is based on a recursive divide-and-conquer strategy which calculates one minimal conflict set at a time. It is not bounded to a special setting (like the product table of the recommender setting), but can be used for a broader area of over-constrained satisfaction problems.

The approach described in [8] uses pinpointing for the identification of repairs for incoherent terminologies. These pinpoints prevent the algorithm of calculating minimal hitting sets by using the superset to approximate minimal diagnoses. To compute the pinpoints themselves all minimal conflict sets are needed. Compared to this the minimal conflict sets used in Model-Based Diagnoses [7] are computed on demand.

[9] have developed an approach to calculate minimal conflict sets for recommender settings based on product tables. This approach is inspired by the concepts of network analysis and was developed especially for knowledge-based recommender systems. The evaluation in [9] compares the QuickXplain [5] in a version where the consistency checks are done with a database. As the study of this paper has shown the table datastructure to check the consistency is much faster than using database queries, we decided to compare the runtime of the FastXplain with the faster implementation of the QuickXplain.

6 Conclusions

In this paper we presented a novel approach to identify minimal conflict sets of an over-constrained CSP. The efficient determination of minimal conflicts is crucial for the usability and acceptance of constraint-based recommender systems. We came up with the algorithm FastXplain which is based on a table created from the user requirements and the products of the assortment combined with the concepts of the Hitting Set Acyclic Directed Graph (HSDAG) [7]. The results of the evaluation shows that our approach performs even better that the state-of-the-art approach QuickXplain.

References

1. Burke, R.: Knowledge-based recommender systems. In: Encyclopedia of Library and Information Systems, New York, NY, USA, vol. 69, pp. 180–200 (2000)
2. Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M.: Consistency-based diagnosis of configuration knowledge bases. Artificial Intelligence 152(2), 213–234 (2004)
3. Felfernig, A., Friedrich, G., Schubert, M., Mandl, M., Mairitsch, M., Teppan, E.: Plausible repairs for inconsistent requirements. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 791–796 (2009)
4. Jannach, D.: Finding preferred query relaxations in content-based recommenders. In: Intelligent Techniques and Tools for Novel System Architectures, pp. 81–97 (2008)
5. Junker, U.: Quickxplain: Preferred explanations and relaxations for over-constrained problems. In: Proceedings of the 19th National Conference on Artificial Intelligence, pp. 167–172. AAAI Press / The MIT Press (2004)
6. O'Sullivan, B., Papadopoulos, A., Faltings, B., Pu, P.: Representative explanations for over-constrained problems. In: Proceedings of the National Conference on Artificial Intelligence, Cork Constraint Computation Centre, University College Cork, Ireland, pp. 323–328 (2007)
7. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32(1), 57–95 (1987)
8. Schlobach, S., Huang, Z., Cornet, R., van Harmelen, F.: Debugging incoherent terminologies. Journal of Automated Reasoning 39(3), 317–349 (2007)
9. Schubert, M., Felfernig, A., Mandl, M.: Solving over-constrained problems using network analysis. In: Proceedings of the International Conference on Adaptive and Intelligent Systems, Klagenfurt, Austria, pp. 9–14 (2009)

Empirical Knowledge Engineering: Cognitive Aspects in the Development of Constraint-Based Recommenders

Alexander Felfernig, Monika Mandl, Anton Pum, and Monika Schubert

Applied Software Engineering, Graz University of Technology,
Inffeldgasse 16b, A-8010 Graz, Austria
alexander.felfernig@ist.tugraz.at, monika.mandl@ist.tugraz.at,
apum@edu.uni-klu.ac.at, monika.schubert@ist.tugraz.at

Abstract. Constraint-based recommender applications provide valuable support in item selection processes related to complex products and services. This type of recommender operates on a knowledge base that contains a deep model of the product assortment as well as constraints representing the company's marketing and sales rules. Due to changes in the product assortment as well as in marketing and sales rules, such knowledge bases have to be adapted very quickly and frequently. In this paper we focus on a specific but very important aspect of recommender knowledge base development: we analyze the impact of different constraint representations on the cognitive effort of a knowledge engineer to successfully complete certain knowledge acquisition tasks. In this context, we report results of an initial empirical study and provide first basic recommendations regarding the design of recommender knowledge bases.

Keywords: Constraint-based Recommender Systems, Knowledge Engineering, Cognitive Psychology, Empirical Studies.

1 Introduction

Constraint-based recommender systems [5] support users in the retrieval of interesting items from large and potentially complex assortments. In contrast to the more traditional approaches such as collaborative filtering [7], constraint-based recommendation approaches exploit explicitly defined advisory knowledge stored in a recommender knowledge base. Such an explicit representation makes constraint-based recommenders more immune regarding new user as well as new item problems that are typically coming along with collaborative and content-based recommendation [7].

The knowledge exploited by constraint-based recommender systems is on the one hand explicit knowledge about potential customer requirements and on the other hand knowledge about the underlying product assortment and the rules that have to be taken into account when calculating a recommendation. The interaction with a constraint-based recommender application is typically conversational [5], i.e., within the scope of a recommendation session, users have to answer (potentially personalized) questions, select from given sets of requirement repairs (in the case that no solution could be found), and evaluate derived recommendations.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 631–640, 2010.

In this paper we focus on aspects related to the development of recommender knowledge bases rather than aspects related to the interaction with a recommender application. We are considering scenarios where knowledge engineers are interacting with a knowledge acquisition environment and try to develop and maintain recommender knowledge bases.[1] An effect occurring in this context is the so-called *knowledge acquisition bottleneck* which denotes the communication overhead between domain experts and knowledge engineers in the phase of implementing and adapting knowledge bases. This is based on the fact that domain experts know about recommendation rules and heuristics, but it is very hard for them to translate those into a corresponding executable representation, but then knowledge engineers have a limited knowledge about the product domain.

Providers of state-of-the-art constraint-based recommender applications [5] are aware of these limitations and already provide sophisticated graphical knowledge base development environments that make recommendation technologies more accessible to domain experts themselves. These environments provide intelligent modeling functionalities, for example, the automated fault detection in recommender knowledge bases. The principle of such a debugging functionality is that domain experts provide examples for the intended input/output behavior of the recommender knowledge base and a diagnosis component automatically determines the set of potentially faulty recommendation constraints (for details see [4]).

These technologies significantly help to improve the overall quality of knowledge base development processes, however, the issue of understanding the impact of different types of knowledge representations on the cognitive processes of knowledge engineers has not been investigated in detail up to now. In this paper we show how different expressive means to represent specific chunks of knowledge are triggering specific cognitive overheads on the side of the knowledge engineer. Consequently, the way we implement a recommender knowledge base has a significant impact on the effectiveness of related knowledge acquisition and maintenance processes. The following simple example should clarify what we mean by different types of knowledge representation:

$$c_1: \textit{user.knowledge-level} = \textit{beginner} \rightarrow \textit{product.risk-level} \neq \textit{high}$$
$$c_2: \neg \textit{knowledge-level} = \textit{beginner} \vee \textit{product.risk-level} \neq \textit{high}$$

Both recommendation constraints $\{c_1, c_2\}$ have exactly the same semantics, however, with high evidence, implications (→) have significant lower cognitive efforts regarding correct interpretation. The major goal of this paper is to present different results stemming from an empirical study that investigated the impact of different knowledge representations on the understandability of a knowledge base.

The remainder of this paper is organized as follows. In Section 2 we introduce a formal definition of a recommendation problem and introduce a simplified recommender knowledge base from the domain of financial services. In Section 3 we report the results of an empirical study that provides interesting insights into the do's and dont's in recommender knowledge base development. In Section 4 we give an overview of related work. With Section 5 we conclude our paper.

[1] The results presented in this paper have been developed within the scope of the COHAVE project (Austrian Research Agency, Grant Nr. 810996).

2 Example: Recommender Knowledge Base

Recommender knowledge bases are typically containing two sets of variables and three different types of constraint sets. $V_C = \{vc_1, vc_2, …, vc_m\}$ - the first set of variables - describes potential *customer requirements*. The second set is $V_{PROD} = \{vprod_1, vprod_2, ..., vprod_n\}$ that describes the structural properties of the underlying *product assortment*. $C_R = \{cr_1, cr_2, …, cr_u\}$ represents a set of constraints that describe *possible* (allowed) *combinations of customer requirements*, $CF = \{cf_1, cf_2, …, cf_v\}$ represents a set of *filter constraints* that relate customer requirements with the corresponding item properties. Finally, $C_{PROD} = \{cprod_1, cprod_2, …, cprod_w\}$ *enumerates the elements* (items) *contained in the offered product assortment*. The following is a simplified example of a recommender knowledge base for the domain of financial services (see [6]).

$\mathbf{V_C}$ = {$\mathbf{vc_1}$:*user.knowledge-level*, $\mathbf{vc_2}$:*user.willingness-to-take-risks*, $\mathbf{vc_3}$:*user.duration-of-investment*, $\mathbf{vc_4}$:*user.advisory-wanted*, $\mathbf{vc_5}$:*user.funds-available*, $\mathbf{vc_6}$:*user.direct-product-search*, $\mathbf{vc_7}$:*user.type-low-risk-products*, $\mathbf{vc_8}$:*user.type-high-risk-products*}

domain(*user.knowledge-level*) = {*expert*, *average*, *beginner*}, *domain*(*user.willingness-to-take-risks*) = {*low*, *medium*, *high*}, *domain*(*user.duration-of-investment*) = {*shortterm*, *mediumterm*, *longterm*}, *domain*(*user.advisory-wanted*) = {*yes*, *no*}, *domain*(*user.funds-available*) = {*yes*, *no*}, *domain*(*user.direct-product-search*) = {*savings*, *bonds*, *stockfunds*, *singleshares*}, *domain*(*user.type-low-risk-products*) = {*savings*, *bonds*}, *domain*(*user.type-high-risk-products*) = {*stockfunds*, *singleshares*}

$\mathbf{V_{PROD}}$ = {$\mathbf{vprod_1}$:*product.name*, $\mathbf{vprod_2}$:*product.expected-return-rate*, $\mathbf{vprod_3}$: *product.risklevel*, $\mathbf{vprod_4}$:*product.min-invest-period*, ***vprod**$_5$*:*product.financial-institute*}

domain(*product.name*) = *text*, *domain*(*product.expected-return-rate*) = {1..40}, *domain*(*product.risk-level*) = {*low*, *medium*, *high*}, *domain*(*product.min-invest-period*) = {1..14}, *domain*(*product.financial-institute*) = *text*

$\mathbf{C_R}$ = {$\mathbf{cr_1}$:*user.willingness-to-take-risks=high* →
user.duration-of-investment ≠ *shortterm*
$\mathbf{cr_2}$:*user.knowledge-level* = *beginner* → *user.willingness-to-take-risks* ≠ *high*}

$\mathbf{C_F}$ = {$\mathbf{cf_1}$: *user.duration-of-investment* = *shortterm* → *product.min-invest-period* < 3,
$\mathbf{cf_2}$: *user.duration-of-investment* = *mediumterm* →
*product.min-invest-perio*d >= 3 ∨ *product.min-invest-period* < 6,
$\mathbf{cf_3}$: *user.duration-of-investment* = *longterm* → *product.min-invest-period* >= 6,
$\mathbf{cf_4}$: *user.willingness-to-take-risks* = *low* → *product.risklevel* = *low*,
$\mathbf{cf_5}$: *user.willingness-to-take-risks* = *medium* →
product.risklevel = *low* ∨ *product.risklevel* = *medium*,
$\mathbf{cf_6}$: *user.willingness-to-take-risks* = *high* →
product.risklevel = *low* ∨ *product.risklevel* = *medium* ∨ *product.risklevel* = *high*,
$\mathbf{cf_7}$: *user.knowledge-level* = *beginner* → *product.risklevel* ≠ *high*,
$\mathbf{cf_8}$: *user.type-low-risk-products* = *savings* → *product.name* = *savings*,
$\mathbf{cf_9}$: *user.type-low-risk-products* = *bonds* → *product.name* = *bonds*}

$\mathbf{C_{PROD}}$ = {$\mathbf{cprod_1}$: *product.name=savings* ∧ *product.expected-return-rate* = 3 ∧
product.risk-level = *low* ∧ *product.min-invest-period* = 1 ∧
product.financial-institute = *A*;
$\mathbf{cprod_2}$: *product.name=bonds* ∧ *product.expected-return-rate* = 5 ∧
product.risk-level = *medium* ∧ *product.min-invest-period* = 5 ∧
product.financial-institute = *B*;
$\mathbf{cprod_3}$: *product.name=equity* ∧ *product.expected-return-rate* = 9 ∧
product.risk-level = *high* ∧ *product.min-invest-period* = 10 ∧
product.financial-institute = *B*}

On the basis of a recommender knowledge base we are able to calculate a set of recommendations for a user (given that the corresponding requirements have been specified). We denote such a set of requirements as C_C = {cc_1, cc_2, ..., cc_t}. For our example recommender knowledge base we could assume the following set of customer requirements: C_C = {cc_1:*user.knowledge-level=beginner*, cc_2:*user.advisory-wanted* = *yes*, cc_3:*user.willingness-to-take-risks* = *medium*}.

Such a set of concrete customer requirements is the precondition for the calculation of *consistent recommendations* (see Definition 1). The calculation of such recommendations is based on solving a constraint satisfaction problem defined by (V_C, V_{PROD}, C_C, C_F, C_R, C_{PROD}) [5].

Definition 1 (*Consistent Recommendation*)**.** An assignment R of the variables in V_C and V_{PROD} is denoted as consistent recommendation for (V_C, V_{PROD}, C_C, C_F, C_R, C_{PROD}) iff it does not violate any of the constraints in $C_C \cup C_F \cup C_R \cup C_{PROD}$.

A consistent recommendation for our example set of requirements is R = {*user.knowledge-level* = *beginner*, *user.willingness-to-take-risks* = *medium*, *user.advisory-wanted* = *yes*, *user.type-low-risk-products* = *savings*, *product.name* = *savings*, *product.expected-return-rate* = *3*, *product.risk-level* = *low*, *product.min-investment-period* = *1*, *product.financial-institute* = *A*}.

Our example knowledge base consists of 8 variables describing potential customer requirements and 5 variables describing relevant properties of products. Additionally, the knowledge base consists of 2 compatibility constraints that describe the possible (allowed) combinations of customer requirements, 9 constraints describing the relationship between customer requirements and product properties (filter constraints), and 3 constraints describing the offered product assortment.

Our example shall provide an impression of the basic structure of a recommender knowledge base. However, real-world applications are typically of larger size, for example, a financial service recommender knowledge base reported in [6] consists of 20 variables describing potential customer requirements, 15 variables describing the underlying product assortment, 20 constraints describing the possible combinations of customer requirements, 30 constraints describing the relationship between customer properties and the product assortment, and about 80 products.

The above knowledge base is simplified not only in terms of the number of contained elements, but as well in the types of contained constraints. Compatibility constraints as well as filter conditions are simply represented in terms of implication structures, i.e. →, that express a basic type of requirement relationship quite often used in the context of knowledge-based recommendation scenarios. However, different types of knowledge representation formalisms are used to express constraints in recommender knowledge bases: logical operators $\{\neg, \leftrightarrow, \rightarrow, \leftarrow, \wedge, \vee\}$, arithmetical operators such as (+,-,*,/), and relational operators ($<, >, \neq, \geq, \leq$).

In the following sections we will discuss some of the observations we gained from an empirical study which investigated the cognitive efforts of knowledge engineers maintaining such knowledge bases.

3 Study: Understandability of Knowledge Representations

The mentioned user study has been conducted within the scope of a university knowledge engineering course (the study design is depicted in Table 1).

Table 1. Different knowledge base versions used in the study

ver.	variant	#vars	#constr.	g_a	g_b
1.a	variable grouping	5	15	√	-
1.b	no variable grouping	5	15	-	√
2.a	→ implications used	5	7	-	√
2.b	no implications used	5	7	√	-
3.a	≤ and ≥ used for range restrictions	5	5	√	-
3.b	< and > used for range restrictions	5	5	-	√
4.a	→ implications used	5	6	-	√
4.b	← implications used	5	6	√	-
5.a	↔ equivalences used	5	3	-	√
5.b	→ implications used	5	6	√	-

52 subjects participated in the empirical study. The participants were assigned to one of the two groups g_a and g_b (see Table 1). Each of the participants had experiences in the development and maintenance of recommender knowledge bases. Knowledge-based systems were within the major scope of the course.

Each of the participants had to complete 5 tasks where the order of tasks to be completed was randomized. The participants had to register the starting time and end time of each task correspondingly. There was no time limit specified regarding the completion of a task, the major goal set was that the participants should find a correct (consistent) solution (recommendation). This information was the basis for comparing the processing times needed for the different tasks. The major goal of each task was to identify a solution for the given constraint satisfaction problem and to write down the identified solution on a sheet of paper. The knowledge bases used within the scope of the study were defined domain-independently in order to avoid problems regarding the understanding of the product domain. For simplicity, we did not differentiate between V_C and V_{PROD} and as well did not differentiate between the different types of constraints, i.e., C_R, C_F, C_{PROD}. We denoted the set of relevant variables with $V = \{v_1, v_2, \ldots, v_m\}$ and the set of constraints with $C = \{c_1, c_2, \ldots, c_n\}$. In the study we used the knowledge base versions shown in Table 1. We defined the following hypotheses. Note that we measured knowledge base understandability in terms of effectiveness in processing (the time needed to identify a solution). For each of the following tasks participants had to complete *one* version (*version a* or *version b*).

Task 1: (*Variable grouping vs. no variable grouping*): these two versions of the knowledge base are equivalent with the exception that in version 1.a the constraints were grouped regarding variable names, for example, the first group contains constraints that altogether include a reference to variable v_1. The hypothesis defined for Task 1 was the following:

Hypothesis H1: *knowledge bases with constraints grouped on the basis of their variable references are easier to understand.*

In order to investigate Hypothesis H1, we compared the time effort needed by the participants to identify a solution for one of the two knowledge bases. In our analysis (two-sample t-test) of the time efforts in version 1.a and version 1.b (Table 2) we could confirm Hypothesis H1: the average time needed for identifying a solution for knowledge base version 1.a was 4.57 minutes (std.dev. 2.89 minutes) whereas for knowledge base version 1.b it was 5.75 minutes (std.dev. 2.97). A corresponding two-sample t-test resulted in a significant difference of the average values ($p<0.05$).

Summarizing, the grouping of constraints on the basis of referenced variables can improve the understandability of a constraint knowledge base. Such a grouping can be achieved on the basis of simple clustering algorithms or nearest-neighbor algorithms.

Task 2: (*Implications "→" used vs. no implications used*): these two versions of the knowledge base are equivalent with the exception that in version 2.a "→" was used to represent implications whereas in version 2.b "¬" and "∨" were used for the same purpose. The hypothesis defined for Task 2 was the following:

Hypothesis H2: *knowledge bases using "→" for the representation of implications are easier to understand.*

In order to investigate Hypothesis H2, we compared the time effort needed by the participants to identify a solution for one of the two knowledge bases.

In our analysis of the time efforts registered for version 2.a and version 2.b (see Table 3) we can support Hypothesis H2: the average time needed for identifying a solution for knowledge base version 2.a was 4.85 minutes (std.dev. 1.89 minutes) whereas for knowledge base version 2.b it was 6.24 minutes (std.dev. 2.96). A corresponding two-sample t-test resulted in a strong tendency ($p=0.092$). Summarizing, the usage of implications instead of the more basic logical operators of "∨" and "¬" can improve the understandability of a constraint knowledge base. If possible the more compact representation of "→" should be used.

Task 3: (*≤ and ≥ used for range restrictions vs.* < and > *used*): these two versions of the knowledge base (see Table 4) are equivalent with the exception that in version 3.a "≤" and "≥" were used to represent range restrictions whereas in version 3.b "<" and ">" were used for the same purpose. The hypothesis defined for Task 3 was the following:

Table 2. Setting: Variable grouping vs. no variable grouping

version 1.a	version 1.b
$V=\{v_1,v_2,v_3,v_4,v_5\}$	$V=\{v_1,v_2,v_3,v_4,v_5\}$
$domain(v_1)=\{1..4\}$, $domain(v_2)=\{1..4\}$ $domain(v_3)=\{1..5\}$, $domain(v_4)=\{1..5\}$ $domain(v_5)=\{1..6\}$	$domain(v_1)=\{1..4\}$, $domain(v_2)=\{1..4\}$ $domain(v_3)=\{1..5\}$, $domain(v_4)=\{1..5\}$ $domain(v_5)=\{1..6\}$
c_1: $v_1 \neq v_5$	c_1: $v_1 \neq v_5$
c_2: $v_1 > 1$	c_2: $v_2 = 2 \vee v_2 = 3$
c_3: $v_1 = 2 \vee v_5 \neq 2$	c_3: $v_3 \neq v_4$
c_4: $v_2 = 2 \vee v_2 = 3$	c_4: $v_4 \neq v_1$
c_5: $v_2 \neq v_1$	c_5: $v_1 > 1$
c_6: $v_2 < v_4$	c_6: $v_2 \neq v_1$
c_7: $v_3 = 3$	c_7: $v_3 = v_5$
c_8: $v_3 \neq v_4$	c_8: $v_5 \neq v_1$
c_9: $v_3 = v_5$	c_9: $v_1 = 2 \vee v_5 \neq 2$
c_{10}: $v_4 \geq v_3$	c_{10}: $v_2 < v_4$
c_{11}: $v_4 \geq v_5$	c_{11}: $v_4 \geq v_3$
c_{12}: $v_4 \neq v_1$	c_{12}: $v_5 \geq 3$
c_{13}: $v_5 \neq v_1$	c_{13}: $v_3 = 3$
c_{14}: $v_5 \neq 5$	c_{14}: $v_4 \geq v_5$
c_{15}: $v_5 \geq 3$	c_{15}: $v_5 \neq 5$

Table 3. Setting: Implications "→" used vs. no implications used

version 2.a	version 2.b
$V=\{v_1,v_2,v_3,v_4,v_5\}$	$V=\{v_1,v_2,v_3,v_4,v_5\}$
$domain(v_1)=\{1..6\}$, $domain(v_2)=\{1..5\}$ $domain(v_3)=\{1..3\}$, $domain(v_4)=\{1..5\}$ $domain(v_5)=\{1..6\}$	$domain(v_1)=\{1..6\}$, $domain(v_2)=\{1..5\}$ $domain(v_3)=\{1..3\}$, $domain(v_4)=\{1..5\}$ $domain(v_5)=\{1..6\}$
c_1: $v_2 \neq 5 \vee \neg(v_1 \geq 2)$	c_1: $v_1 \geq 2 \rightarrow v_2 \neq 5$
c_2: $v_3 = 2 \vee v_3 = 3 \vee \neg(v_5 = 1)$	c_2: $v_5 = 1 \rightarrow v_3 = 2 \vee v_3 = 3$
c_3: $v_5 \leq 4 \vee \neg(v_4 \geq 1)$	c_3: $v_4 \geq 1 \rightarrow v_5 \leq 4$
c_4: $v_3 = 1 \vee \neg(v_2 = 2)$	c_4: $v_2 = 2 \rightarrow v_3 = 1$
c_5: $v1 \neq 1 \vee \neg(v_3 = 1)$	c_5: $v_3 = 1 \rightarrow v_1 \neq 1$
c_6: $(v_4 = 2 \wedge v_1 > v_5) \vee \neg(v_3 = 1)$	c_6: $v_3 = 1 \rightarrow (v_4 = 2 \wedge v_1 > v_5)$
c_7: $v_3 = 1$	c_7: $v_3 = 1$

Hypothesis H3: *knowledge bases using "≤" and "≥" for the representation of range restrictions are easier to understand* (compared to "<" and ">").

In order to investigate Hypothesis H3, we compared the time efforts needed by the participants to identify a solution for one of the two knowledge bases. In our analysis of the time efforts registered for version 3.a and version 3.b (see Table 4) we could clearly confirm Hypothesis H3: the average time needed for identifying a solution for knowledge base version 3.a was 3.1 minutes (std.dev. 1.71 minutes) whereas for knowledge base version 3.b it was 4.39 minutes (std.dev. 2.25). A corresponding two-sample t-test resulted in a significant difference of the average values ($p<0.05$), i.e., the usage of "≤" instead of "<" and "+1" can improve understandability.

Table 4. Setting: ≤ and ≥ used for range restrictions vs. < and > used

version 3.a	version 3.b
$V=\{v_1,v_2,v_3,v_4,v_5\}$	$V=\{v_1,v_2,v_3,v_4,v_5\}$
see Table 2.	see Table 2.
c_1: $v_1 \leq v_3 \wedge v_2 \geq v_1$ c_2: $v_5 \geq v_1 \vee v_4 \leq v_3$ c_3: $v_3 \leq v_2$ c_4: $v_5 \geq v_2 \wedge v_5 \leq v_3$ c_5: $v_4 \leq v_1$	c_1: $v_1 < v_3 + 1 \wedge v_2 + 1 > v_1$ c_2: $v_5 + 1 > v_1 \vee v_4 < v_3 + 1$ c_3: $v_3 < v_2 + 1$ c_4: $v_5 + 1 > v_2 \wedge v_5 < v_3 + 1$ c_5: $v_4 < v_1 + 1$

Task 4: (*Implications → used vs. ← used*): these two versions of the knowledge base are equivalent with the exception that in version 4.a "→" was used to represent range restrictions whereas in version 4.b "←" was used for the same purpose. The hypothesis defined for Task 4 was the following:

Hypothesis H4: *knowledge bases using "→" for the representation of implications are easier to understand* (compared to implications of the form "←").

In order to investigate Hypothesis H4, we compared the time effort needed by the participants to identify a solution for one of the two knowledge bases. In our analysis of the time efforts registered for version 4.a and version 4.b (see Table 5) we could not confirm H4: the average time needed for identifying a solution for knowledge base version 4.a was 4.30 minutes (std.dev. 2.30 minutes) whereas for knowledge base version 4.b it was 4.0 minutes (std.dev. 2.20). A corresponding two-sample t-test resulted in no significant differences between the average values. Summarizing, "→" and "←" as two possible different representations of implications do show differences in terms knowledge base understandability but without statistical significance.

Task 5: (*Equivalences ↔ used vs. → used*): these two versions of the knowledge base are equivalent with the exception that in version 5.a "↔" was used whereas in version 5.b "→" was used. The hypothesis defined for Task 5 was the following:

Hypothesis H5: *knowledge bases using "↔"are easier to understand* (compared to representations of the form "→").

In order to investigate Hypothesis H5, we compared the time effort needed by the participants to identify a solution for one of the two knowledge bases. In our analysis of the time efforts registered for version 5.a and version 5.b (see Table 6) we could confirm Hypothesis H5: the average time needed for identifying a solution for knowledge base version 5.a was 2.50 minutes (std.dev. 1.70 minutes) whereas for

Table 5. Setting: Implications → vs. ← used

version 4.a	**version 4.b**
$V=\{v_1,v_2,v_3,v_4,v_5\}$	$V=\{v_1,v_2,v_3,v_4,v_5\}$
$domain(v_1)=\{1..4\}$, $domain(v_2)=\{1..4\}$	$domain(v_1)=\{1..4\}$, $domain(v_2)=\{1..4\}$,
$domain(v_3)=\{1..4\}$, $domain(v_4)=\{1..4\}$	$domain(v_3)=\{1..4\}$, $domain(v_4)=\{1..4\}$
$domain(v_5)=\{1..4\}$	$domain(v_5)=\{1..4\}$
c_1: $v_2 \neq 5 \leftarrow v_1 \neq 3$	C1: $v1 \neq 3 \rightarrow v2 \neq 5$
c_2: $v_2 = 2 \leftarrow v_4 = 1$	c2: $v4 = 1 \rightarrow v2 = 2$
c_3: $v_2 > 3 \leftarrow v_3 > 1$	c3: $v3 > 1 \rightarrow v2 > 3$
c_4: $v_3 = 3 \leftarrow v_5 \neq v1$	c4: $v5 \neq v1 \rightarrow v3 = 3$
c_5: $(v_1 = v_3 \wedge v_1 > v_5) \leftarrow v_2 = v_1$	c5: $v2 = v1 \rightarrow (v1 = v3 \wedge v1 > v5)$
c_6: $(v_1 \geq v_2) \leftarrow$ true	c6: true $\rightarrow (v1 \geq v2)$

Table 6. Setting: Equivalences ↔ used vs. → used

version 5.a	**version 5.b**
$V=\{v_1,v_2,v_3,v_4,v_5\}$	$V=\{v_1,v_2,v_3,v_4,v_5\}$
see Table 5.	see Table 5.
	c_1: $v_2 \neq 3 \rightarrow v_3 > 3$
c_1: $v_2 \neq 3 \leftrightarrow v_3 > 3$	c_2: $v_3 > 3 \rightarrow v_2 \neq 3$
c_2: $v_4 \neq 2 \leftrightarrow v_5 > 4$	c_3: $v_4 \neq 2 \rightarrow v_5 > 4$
c_3: $v_1 > 3 \leftrightarrow v_3 > 1$	c_4: $v_5 > 4 \rightarrow v_4 \neq 2$
	c_5: $v_1 > 3 \rightarrow v_3 > 1$
	c_6: $v_3 > 1 \rightarrow v_1 > 3$

knowledge base version 5.b it was 3.10 minutes (std.dev. 1.90). A corresponding two-sample t-test resulted in significant differences between the average values ($p<0.05$).

4 Related Work

The work presented in this paper is one basic step in a broader line of research related to the understanding of cognitive factors in the development and maintenance of knowledge-based applications. Our major goal in this context is to develop concepts that actively support knowledge engineers and domain experts in the development of (recommender) knowledge bases beginning with supporting knowledge base understanding for new employees as well as supporting different maintenance scenarios such as test case generation or automated debugging.

The discipline of program understanding is already established in the area of software engineering but is still in its infancy in the area of knowledge-based systems. Program understanding can be interpreted as a process of building a mental model of the software (code, application domain, design, etc.). Hipikat [3] is an example system that helps project-newcomers in understanding different project artefacts. The system is based on a collaborative filtering recommendation approach that calculates artefact recommendations on the basis of already existing interaction histories.

Compared to the work presented in this paper, Hipikat focuses on learning support on the basis of collaborative techniques but does not explicitly take into account elementary complexity properties of the software artefacts. The RASCAL environment [2] supports software developers in their programming tasks by automatically recommending interesting library components useful in a certain context. Such recommendations are calculated on the basis of collaborative filtering algorithms.

Knowledge base refactoring approaches [1] follow the goal of improving the structural design of given knowledge base. Due to repeated maintenance activities, the overall quality of the knowledge base deteriorates and mechanisms are needed that help to restore quality through a set of reorganization steps. [1] discuss knowledge base design anomalies and provide detailed recommendations on how to deal with such situations. Examples for such anomalies are objects never used by the application, attribute domain values nearly never used, etc.

5 Conclusions

In this paper we have presented first results of an empirical study related to the understandability of (recommender) knowledge bases. The results of this study document the necessity of a deeper understanding of the consequences of different types of knowledge representations on the quality of a knowledge base in terms of, for example, understandability or maintainability. Such an understanding of knowledge base structures helps to develop models about knowledge base quality that can provide valuable support in knowledge base development and maintenance processes.

References

[1] Baumeister, J., Puppe, F., Seipel, D.: Refactoring Methods for Knowledge Bases. In: 12th International Conference on Knowledge Engineering and Knowledge Management Knowledge Patterns, pp. 157–171 (2004)

[2] McCarey, F., Cinneide, M., Kushmerick, N.: Rascal: A recommender agent for agile reuse. Artificial Intelligence Review 24(3-4), 253–276 (2005)

[3] Cubranic, D., Murphy, G.: Hipikat: recommending pertinent software development artifacts. In: 25th International Conference on Software Engineering, Portland, Oregon, pp. 408–418 (2003)

[4] Felfernig, A., Friedrich, G., Jannach, D., Stumptner, M.: Consistency-based Diagnosis of configuration knowledge bases. AI Journal 152(2), 213–234 (2004)

[5] Felfernig, A., Burke, R.: Constraint-based Recommender Systems: Technologies and Research Issues. In: IEEE International Conference on Electronic Commerce, Innsbruck, Austria, pp. 1–10 (2008)

[6] Felfernig, A., Isak, K., Russ, C.: Knowledge-based Recommendation: Technologies and Experiences from Projects. In: 17th European Conference on Artificial Intelligence (ECAI06), Riva del Garda, Italy, pp. 632–636 (2006)

[7] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news Full text. Communications of the ACM 40(3), 77–87 (1997)

Adaptive Utility-Based Recommendation

Alexander Felfernig[1], Monika Mandl[1],
Stefan Schippel[2], Monika Schubert[1], and Erich Teppan[2]

[1] Applied Software Engineering, Graz University of Technology,
Inffeldgasse 16b, A-8010 Graz, Austria
{alexander.felfernig,monika.mandl,
monika.schubert}@ist.tugraz.at
[2] Intelligent Systems and Business Informatics, University Klagenfurt,
Universitaetsstrasse 65-67, A-9020 Klagenfurt, Austria
{stefan.schippel,erich.teppan}@uni-klu.ac.at

Abstract. Knowledge-based recommenders support customers in preference construction processes related to complex products and services. In this context, utility constraints (scoring rules) play an important role. They determine the order in which items (products and services) are presented to customers. In many cases utility constraints are faulty, i.e., calculate rankings which are not expected and accepted by marketing and sales experts. The adaptation of these constraints is extremely time-consuming and often an error-prone process. In this paper we present an approach which effectively supports the automated adaptation of utility constraint sets based on solutions for corresponding nonlinear optimization problems. This approach significantly increases the applicability of knowledge-based recommendation by allowing the automated reproduction of item rankings specified by marketing and sales experts.

Keywords: Utility-based recommendation, non-linear optimization.

1 Introduction

Recommender applications [1][5][6][8][13] support users in the identification of interesting products and services (items) in situations where the amount and/or complexity of offers outstrips the capability of a user to survey it and to reach a decision [5]. Knowledge-based recommenders (e.g., [1][5][6][8]) exploit deep knowledge about the product/service domain in order to determine solutions fitting the wishes and needs of customers. Customers purchasing items such as digital cameras, computers, or financial services are much more in the need of intelligent interaction mechanisms supporting the calculation and explanation of appropriate solutions. Exactly such mechanisms are supported by knowledge-based recommender technologies. Items selected by a knowledge-based recommender are ranked according to their utility for the customer [6][8]. Due to serial position effects [9][11], which induce customers to preferably take a look at and select items at the beginning of a list, such rankings can increase the trust in the presented recommendation as well as the willingness to buy an item.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 641–650, 2010.

We apply Multi-Attribute Utility Theory (MAUT) [12][14] for the calculation of item rankings. Each item is evaluated according to a predefined set of interest dimensions. *Profit*, *availability*, and *risk* are examples of interest dimensions in the domain of financial services. If a customer is more interested in high return rates than in the availability of an item, the dimension *profit* is very important. In this way, the importance of customer requirements influences the importance of corresponding interest dimensions. A successful application of recommender technologies requires that utility constraints (scoring rules - see Figure 1) consistently reflect the marketing and sales rules of a company. In the financial services domain an example for such a marketing and sales rule is *if both, balanced funds and bonds are part of the recommendation result, the utility of bonds should be higher than that of balanced funds.* Experiences from commercial projects [6][8] show the need for knowledge acquisition support which alleviates the development and maintenance of utility constraint sets. The manual repair of utility constraints to produce results consistent with marketing and sales rules is a time-consuming and error-prone task since those constraints are strongly interdependent. Therefore our goal was to develop techniques that effectively support knowledge engineers in the automated adaptation of faulty utility constraint sets. In the remainder of this paper we present a technique that automatically proposes repairs for utility constraint sets which are inconsistent with a given set of rules stemming from marketing and sales departments.

The remainder of the paper is organized as follows. The next section introduces a set of utility constraints for the ranking of product/service alternatives. This constraint set is used as working example throughout the paper. Section 3 sketches our approach to transform a set of utility constraints into a non-linear optimization problem which is used to identify minimum distance repairs compared to the original set of utility constraints. In Section 4 we present empirical findings related to the development of utility constraint sets. Related work is discussed in Section 5. With Section 6 we conclude the paper.

2 Working Example: Utility Constraints

We now present a simplified set of utility constraints used as a working example throughout the paper. The basic elements of Multi-Attribute Utility Theory (MAUT) [12][14] are interest dimensions that describe focuses of interest of customers. Our example contains two such interest dimensions: *profit* and *availability* (see Figure 1). Profit denotes the expected performance in terms of return rate. Availability is related to the accessibility of the invested sum within the targeted investment period. The degree to which a customer is interested in such dimensions can be directly derived from the requirements articulated within a recommendation session. A customer interested in short term investments (*investment period* = *short term*) has a higher interest in availability than a customer who is interested in long term investments (*investment period* = *long term*). Similarly, customers interested in stable-growth investments (*goal* = *stable growth*) have a lower interest in very high profits than those interested in speculations (*goal* = *speculation*). In our example we use the customer requirements shown in Figure 1: customer Robert is interested in *medium term investments* with the goal of *putting money by for a rainy day*. By interpreting the utility constraints of Figure 1 we can determine a distribution for the interest dimensions, i.e., to which extent Robert has a focus on the dimensions *profit* and *availability* (see Table 1). Robert requires a *medium term* investment solution which contributes an importance of 6 to the dimension profit and an importance of 5 to the interest dimension availability. Furthermore Robert is interested in putting *money by for a rainy day* which contributes an importance of 2 to the dimension profit and an importance of 6 to availability. On the basis of customer preferences (expressed as interest in the different interest dimensions) we evaluate the suitability of alternative products. We introduce the following reduced assortment of financial services {*balanced funds*, *bonds*, *bonds2*} depicted in Table 2. Next, we define the utility of products with respect to interest dimensions. We define a dependence between item attribute values and the dimensions. For instance, financial services including shares support a higher profit (score 5 in Figure 1). Furthermore, financial services without shares have a higher degree of availability (score 7), and services with a higher value fluctuation have a higher (expected) profit (score 7)[1].

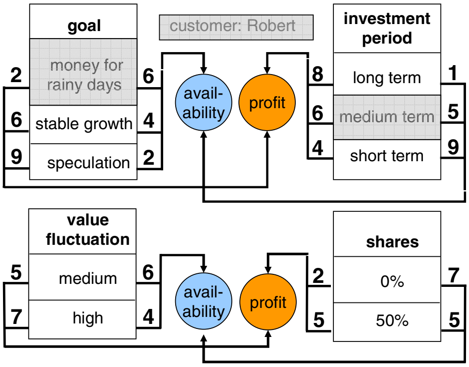


Fig. 1. Scoring rules for customer properties {*goal*, *investment period*}, example customer requirements {*customer*:*Robert*}, and scoring rules for item properties {*value fluctuation*, *shares*}

By interpreting the utility constraints of Figure 1, we can determine the extent to which our financial services contribute to the interest dimensions profit and availability (see Table 3): balanced funds have a higher profit than bonds but by the majority a lower degree of availability. Vice-versa, bonds by the majority have a higher degree of availability than balanced funds. On the basis of the identified item utilities, we can determine the customer-specific utility of each item (see Table 4).

The item utility can be determined using Formula (1):

$$utility(x) = \sum_{i=1}^{n} in_i * con_i(x) \tag{1}$$

where *utility*(*x*) specifies the utility of an item *x* for a specific customer. This utility is defined as sum of a customer's interest values in dimensions *i* (*in*$_i$) times the

[1] For the reasons of simplicity we restrict our working example to three services.

contribution of item *x* to dimension *i* (*con*$_i$). In our application example, *balanced funds* have a higher utility for Robert than *bonds* and *bonds2*.

Table 1. Example customer interests

customer	profit	availability
Robert	6+2=8	5+6=11

Table 2. Example set of financial services

name	shares	value fluctuation
balanced funds	50%	Medium
bonds	0%	Medium
bonds2	0%	High

Table 3. Utilities regarding interest dimensions

name	profit	availability
balanced funds	5+5=10	5+6=11
Bonds	2+5=7	7+6=13
bonds2	2+7=9	7+4=11

Table 4. Utilities of items for example customer

customer	Item	profit	avail-ability	utility
Robert	balanced funds	8*10	11*11	201
	Bonds	8*7	11*13	199
	bonds2	8*9	11*11	193

In order to test whether a given set of utility constraints calculates intended rankings, a corresponding set of examples (test cases) can be provided by marketing and sales experts (see, e.g., Table 5).

Table 5. Examples for intended service rankings: u(x) = utility of product x

example	inv. period	goal	ranking
e$_1$	med. term	rainy days	u(bonds) > u(balanced funds)
e$_2$	med. term	rainy days	u(bonds2) > u(balanced funds)
e$_3$	med. term	rainy days	u(bonds) > u(bonds2)

In the case that the rankings calculated by the utility constraint set are in contradiction with the rankings of the given examples, we have to identify repairs such that consistency with the examples is restored. In our scenario, the examples (E = {e$_1$, e$_2$, e$_3$}) are partially contradicting with the rankings shown in Table 4 (e.g., the utility of bonds is lower than the utility of balanced funds if a customer is interested in medium term investments for rainy days, the contrary is specified in e_1: $u(bonds) > u(balancedfunds)$). Consequently, we have to identify an adaptation of our utility constraints. An approach to derive such adaptations automatically will be discussed in the following sections.

3 Constraint Set Adaptation as Optimization Problem

Following the definitions of Figure 1, we now introduce a set of utility constraints related to the required investment period and the personal goals of the customer. For instance, constraint c_1 denotes the fact that for customers requiring financial services with short term investment periods, the dimension profit is of medium importance, whereas availability aspects play a significantly more important role (c_2). $\{c_1, \ldots, c_{12}\}$ are utility definitions derived from Figure 1.

c_1: $\text{profit}_{(\text{investmentperiod short})} = 4$	c_7: $\text{profit}_{(\text{goal rainydays})} = 2$
c_2: $\text{availability}_{(\text{investmentperiod short})} = 9$	c_8: $\text{availability}_{(\text{goal rainydays})} = 6$
c_3: $\text{profit}_{(\text{investmentperiod medium})} = 6$	c_9: $\text{profit}_{(\text{goal growth})} = 6$
c_4: $\text{availability}_{(\text{investmentperiod medium})} = 5$	c_{10}: $\text{availability}_{(\text{goal growth})} = 4$
c_5: $\text{profit}_{(\text{investmentperiod long})} = 8$	c_{11}: $\text{profit}_{(\text{goal speculation})} = 9$
c_6: $\text{availability}_{(\text{investmentperiod long})} = 1$	c_{12}: $\text{availability}_{(\text{goal speculation})} = 2$

We denote each constraint defining utility values as utility constraint $c_i \in C$. Since we are interested in a utility constraint set consistent with all the examples $e_i \in E$ (provided by domain experts), we have to check the consistency of the given set of utility constraints with $\cup e_i$. This type of consistency check (done by a non-linear optimization component [9]) requires a representation where each example is described by a separate set of variables. For instance, the contribution to the interest dimension profit provided by the customer attribute investment period in example e_1 is stored in the variable $\text{profit}_{(\text{investmentperiod e1})}$.

e_1: $\text{profit}_{(\text{investmentperiod e1})}=\text{profit}_{(\text{investmentperiod medium})} \wedge$
$\text{availability}_{(\text{investmentperiod e1})}=\text{availability}_{(\text{investmentperiod medium})} \wedge$
$\text{profit}_{(\text{goal e1})}=\text{profit}_{(\text{goal rainydays})} \wedge \text{availability}_{(\text{goal e1})}=\text{availability}_{(\text{goal rainydays})} \wedge$
$\text{utility}_{(\text{balancedfunds e1})} < \text{utility}_{(\text{bonds e1})}$

e_2: $\text{profit}_{(\text{investmentperiod e2})}=\text{profit}_{(\text{investmentperiod medium})} \wedge$
$\text{availability}_{(\text{investmentperiod e2})}=\text{availability}_{(\text{investmentperiod medium})} \wedge$
$\text{profit}_{(\text{goal e2})}=\text{profit}_{(\text{goal rainydays})} \wedge \text{availability}_{(\text{goal e2})}=\text{availability}_{(\text{goal rainydays})} \wedge$
$\text{utility}_{(\text{balancedfunds e2})} < \text{utility}_{(\text{bonds2 e2})}$

e3: $\text{profit}_{(\text{investmentperiod e3})}=\text{profit}_{(\text{investmentperiod medium})} \wedge$
$\text{availability}_{(\text{investmentperiod e3})}=\text{availability}_{(\text{investmentperiod medium})} \wedge$
$\text{profit}_{(\text{goal e3})}=\text{profit}_{(\text{goal rainydays})} \wedge$
$\text{availability}_{(\text{goal e3})}=\text{availability}_{(\text{goal rainydays})} \wedge$
$\text{utility}_{(\text{bonds2 e3})} < \text{utility}_{(\text{bonds e3})}$

The overall customer interest in the dimension profit is represented by profit(e_i). The values of these variables represent the sum over all defined contributions of customer requirements of example e_i to the dimension profit. This approach is analogously applied to the interest dimension availability (availability(e_i)). We denote constraints summing up customer utilities as $s_i \in S$, for example:

s_1: profit(e_1)=$\text{profit}_{(\text{investmentperiod e1})}$+$\text{profit}_{(\text{goal e1})}$ ...

For each service item of our example assortment, we specify its contribution to the interest dimensions. For instance, the shares percentage specified for the service *balanced funds* defines an average contribution to the dimension profit. We define this fact as $\text{profitshares}_{(\text{balancedfunds})} = 5$. Analogously, we define the relationship between the interest dimension availability and shares percentage as $\text{availabilityshares}_{(\text{balancedfunds})} = 5$. We denote each constraint defining a utility value for a certain service as utility constraint $p_j \in P$. The following constraints show the implementation of the definitions of Figure 1.

p_1: $\text{profitshares}_{(\text{balancedfunds})}$=5	p_7: $\text{profitfluctuation}_{(\text{balancedfunds})}$=5
p_2: $\text{availabilityshares}_{(\text{balancedfunds})}$=5	p_8: $\text{availabilityfluctuation}_{(\text{balancedfunds})}$=6
p_3: $\text{profitshares}_{(\text{bonds})}$=2	p_9: $\text{profitfluctuation}_{(\text{bonds})}$=5
p_4: $\text{availabilityshares}_{(\text{bonds})}$=7	p_{10}: $\text{availabilityfluctuation}_{(\text{bonds})}$=6
p_5: $\text{profitshares}_{(\text{bonds2})}$=2	p_{11}: $\text{profitfluctuation}_{(\text{bonds2})}$=7
p_6: $\text{availabilityshares}_{(\text{bonds2})}$=7	p_{12}: $\text{availabilityfluctuation}_{(\text{bonds2})}$=4

For each $c_i \in C$ (and each $p_i \in P$) we add a corresponding repair constraint cr_i (pr_i) which specifies repair domains for c_i (p_i). We denote $\bigcup cr_i \cup \bigcup pr_i$ as the set of repair constraints R.

cr_1: $\text{profit}_{(\text{investmentperiod short})}$ IN [0..10]	pr_1: $\text{profitshares}_{(\text{balancedfunds})}$ IN [0..10]
cr_2: $\text{availability}_{(\text{investmentperiod short})}$ IN [0..10]	pr_2: $\text{availabilityshares}_{(\text{balancedfunds})}$ IN [0..10]
cr_3: $\text{profit}_{(\text{investmentperiod medium})}$ IN [0..10]	pr_3: $\text{profitshares}_{(\text{bonds})}$ IN [0..10]
cr_4: $\text{availability}_{(\text{investmentperiod medium})}$ IN [0..10]	pr_4: $\text{availabilityshares}_{(\text{bonds})}$ IN [0..10]
cr_5: $\text{profit}_{(\text{investmentperiod long})}$ IN [0..10]	pr_5: $\text{profitshares}_{(\text{bonds2})}$ IN [0..10]
cr_6: $\text{availability}_{(\text{investmentperiod long})}$ IN [0..10]	pr_6: $\text{availabilityshares}_{(\text{bonds2})}$ IN [0..10]
cr_7: $\text{profit}_{(\text{goal rainydays})}$ IN [0..10]	pr_7: $\text{profitfluctuation}_{(\text{balancedfunds})}$ IN [0..10]
cr_8: $\text{availability}_{(\text{goal rainydays})}$ IN [0..10]	pr_8: $\text{availabilityfluctuation}_{(\text{balancedfunds})}$ IN [0..10]
cr_9: $\text{profit}_{(\text{goal growth})}$ IN [0..10]	pr_9: $\text{profitfluctuation}_{(\text{bonds})}$ IN [0..10]
cr_{10}: $\text{availability}_{(\text{goal growth})}$ IN [0..10]	pr_{10}: $\text{availabilityfluctuation}_{(\text{bonds})}$ IN [0..10]
cr_{11}: $\text{profit}_{(\text{goal speculation})}$ IN [0..10]	pr_{11}: $\text{profitfluctuation}_{(\text{bonds2})}$ IN [0..10]
cr_{12}: $\text{availability}_{(\text{goal speculation})}$ IN [0..10]	pr_{12}: $\text{availabilityfluctuation}_{(\text{bonds2})}$ IN [0..10]

The profit of a financial service is defined by the sum of contributions of the values of shares and fluctuation. Availability of a service is as well defined by the sum of related contributions of those attribute values. We denote each rule summing up service utility values as $s_\alpha \in S$. For example, the following constraint implements the summing of profit utility values as defined in Table 3.

s_a: profit$_{\text{(balancedfunds)}}$=profitshares$_{\text{(balancedfunds)}}$ + profitfluctuation$_{\text{(balancedfunds)}}$ ...

The following constraint exemplifies the specification of the calculation of product/service utilities, where utility(x_{ei}) specifies the utility of item x in the context of example e_i (see Table 4).

s_b:utility$_{\text{(balancedfundse1)}}$=profit$_{\text{(balancedfunds)}}$*profit$_{\text{(e1)}}$+
availability$_{\text{(balancedfunds)}}$*availability$_{\text{(e1)}}$

The above defined constraints are the basic elements of an optimization problem that minimizes repair distances using the following function – Formula (2):

$$Minimize : \sum_{i=1}^{m} \left|val(c_i) - val(cr_i)\right| + \sum_{j=1}^{n} \left|val(p_j) - val(pr_j)\right| \qquad (2)$$

In this formula, |*val*(c_i)-*val*(cr_i)| denotes the degree in which the original value of the utility constraint (scoring value for customer requirements) has been changed. Furthermore, |*val*(p_j)-*val*(pr_j)| denotes the degree in which the original value of the (product) utility constraint has been changed. Based on the definitions above, the Minos solver [10] calculates the following solution for the optimization problem. The distance to the original scoring values of the repair constraints is 11.667.

val(cr_1) = profit$_{\text{(investmentperiod short)}}$ = 4	val(pr_1) = profitshares$_{\text{(balancedfunds)}}$=5
val(cr_2) = availability$_{\text{(investmentperiod short)}}$ = 9	val(pr_2)= availabilityshares$_{\text{(balancedfunds)}}$=4.99
val(cr_3) = profit$_{\text{(investmentperiod medium)}}$ = 5	val(pr_3) = profitshares$_{\text{(bonds)}}$=2
val(cr_4) = availability$_{\text{(investmentperiod medium)}}$ = 4	val(pr_4) = availabilityshares$_{\text{(bonds)}}$=6.33
val(cr_5) = profit$_{\text{(investmentperiod long)}}$ = 8	val(pr_5) = profitshares$_{\text{(bonds2)}}$=2.00
val(cr_6) = availability$_{\text{(investmentperiod long)}}$ = 1	val(pr_6) = availabilityshares$_{\text{(bonds2)}}$=7
val(cr_7) = profit$_{\text{(goal rainydays)}}$ = 2	val(pr_7) = profitfluctuation$_{\text{(balancedfunds)}}$=5.00
val(cr_8) = availability$_{\text{(goal rainydays)}}$ = 5	val(pr_8) = availabilityfluctuation$_{\text{(balancedfunds)}}$=5
val(cr_9) = profit$_{\text{(goal growth)}}$ = 6	val(pr_9) = profitfluctuation$_{\text{(bonds)}}$=5
val(cr_{10}) = availability$_{\text{(goal growth)}}$ = 4	val(pr_{10}) = availabilityfluctuation$_{\text{(bonds)}}$=6.22
val(cr_{11}) = profit$_{\text{(goal speculation)}}$ = 9	val(pr_{11}) = profitfluctuation$_{\text{(bonds2)}}$=6.00
val(cr_{12}) = availability$_{\text{(goal speculation)}}$ = 2	val(pr_{12}) = availabilityfluctuation$_{\text{(bonds2)}}$=4.67

The application of those repairs results in the new rankings depicted in Table 6. Those rankings now are consistent with the given set of examples $e_i \in$ E.

Table 6. Utilities of items for example customer (after repair)

customer	item	utility
Robert	balanced funds	160.004
	Bonds	162.004
	bonds2	161.004

4 Evaluations

Experiences from commercial projects. Our experiences from commercial recommender projects clearly showed the need for more effective engineering of utility constraint sets. An investment recommender of the Hypo Alpe-Adria-Bank in Austria has been implemented without the repair functionalities presented above. The system comprises 15 parameters for specifying customer requirements, 10 item properties and about 500 scoring rules (interest dimensions: availability, profit, risk). The recommender application has been designed, developed, and deployed with an overall effort of about 12 man months. Before deploying the first version of the application, new versions of the utility constraint set have been released every third week and tested by domain experts. About 15 adaptation cycles were needed before deploying the application in the productive environment. Adaptation efforts of each release consumed about 12 hours. This results in 180 hours of development and maintenance efforts specifically related to the adaptation of the utility constraint set. In each adaptation cycle, the knowledge engineer tried to adapt the current utility constraint set to be consistent with the example rankings provided by domain experts. The process was error-prone and time-consuming and triggered requirements to automate the repair of utility constraint sets. The major problem in this context was and is the (practical) impossibility (!) to manually detect a set of repair actions that make a utility constraint set consistent with the set of examples. By exploiting the presented repair functionalities a reduction of the overall efforts related to utility constraint maintenance by about 60% can be expected which means more than 100 hours of time savings in projects similar to the described case.

Evaluation of optimization algorithm. We have evaluated the performance of the Minos solver [10] component with commercial utility constraint sets from the product domains of refrigerators (*refrig1-4*), financial services (*finserv1-4*), and computer monitors (*mon1-4*) (see Table 7). The Minos solver was integrated with the MAUT component of our commercially available knowledge-based recommender environment [5]. From the MAUT definitions of the recommender environment we automatically generated utility constraint sets interpretable by the Minos solver.

Table 7. Example test settings

rec.	#e	#p	#s	#ch	Avg(d)	t(msec)
refrig1	5	16	39	39	<0.001	761
refrig2	10	30	69	55	0.085	1221
refrig3	15	43	80	62	0.082	1998
refrig4	20	55	100	74	0.077	2252
finserv1	5	19	266	160	0.070	1622
finserv2	10	37	396	244	0.054	5599
finserv3	15	53	439	288	0.055	6389
finserv4	20	71	503	340	0.056	9464
mon1	5	22	109	40	0.046	731
mon2	10	42	177	75	0.041	1813
mon3	15	61	214	105	0.044	1542
mon4	20	80	246	125	0.047	2063

For each of the above mentioned application domains we have defined four different settings which differed in the number of examples (*#e*) and the number of products (#p). For example, in *finserv4 #e=20* examples were defined for *#p=71* products. The resulting utility constraint set comprised *#s=503* utility constraints (scoring rules). In order to make the *20* examples consistent with the given set of scoring rules, *#ch=340* rules have been adapted with an average change distance *avg*(*d*)=*0.056* where each scoring rule is defined over the domain [*0..10*], for example, the scoring rule

$$p_7\text{: profitfluctuation}_{(\text{balancedfunds})}=5$$

has been adapted to

$$p_7\text{' = profitfluctuation}_{(\text{balancedfunds})}=5.00061$$

which corresponds to a change distrance of 0.00061 for the scoring rule p_7.

The time needed by the Minos solver [10] to calculate the adaptations for *finserv4* was *t=9464 milliseconds*. The Minos solver is capable of calculating adaptations for faulty utility constraint sets within a reasonable time span acceptable for utility constraints engineering scenarios. In such scenarios, the system uses either examples defined by marketing and sales experts or examples automatically derived from existing user interaction logs. In our test settings, we use examples directly provided by domain experts, the automated derivation of examples is the goal for future work.

5 Related Work

The work presented in [4] follows a pure learning-based approach where preference functions are learned on the basis of examples. These functions are subsequently used to order a set of new solutions. In contrast to pure learning-based approaches, our approach defines a kind of ranking seed knowledge (the initial utility constraint set) which is subsequently automatically repaired to conform with new situations. Thus we avoid cold start problems and effectively support continuous updates of utility constraint sets. The work of [2] deals with modelling constraint problems using soft constraints. In this context, solution preferences are treated as examples which are exploited for the learning of constraint preferences. Compared to our work, [2] apply a learning algorithm that does not require the consistency between the given set of solution preferences and the learned constraint preferences. In this case, the optimality of a solution strongly depends on the accepted solution preferences. Finally, [6] present an approach to the identification of minimal sets of scoring rules which should be adapted in order to recover consistency between the examples $e_i \in E$ and the defined set of scoring rules. The major difference between the work presented in this paper and the approach of [6] is that [6] focuses on the determination of minimal-cardinality sets of scoring rules which have to be adapted in order to achieve consistency between examples and utility constraint set. This is achieved using the concepts of model-based diagnosis that calculates minimal sources of inconsistencies on the basis of pre-calculated conflict sets [6]. Minimality in the sense of [6] is interpreted in terms of the number of identified faulty utility constraints (scoring rules) whereas minimality in the sense of this paper is interpreted as minimal distance between the original scoring definitions and the scoring definitions in the adapted set of scoring rules - see Formula (2). Both approaches have their advantages. The approach presented in [6] can be applied to identify typical modelling faults and the approach of this paper can be used to support periodical update processes which keep the new set of scoring rules as near as possible to the original version.

6 Conclusions

In this paper we have presented innovative knowledge engineering techniques that effectively support knowledge engineers and domain experts in the development and maintenance of utility constraint sets used for the calculation of item rankings. We proposed a repair approach for identifying faulty elements in utility constraint sets that is based on solutions for non-linear optimization problems. With this approach we can achieve significant time savings in important but error-prone, difficult, and often frustrating knowledge base development tasks.

References

[1] Burke, R.: Knowledge-based Recommender Systems. Encyclopedia of Library and Information Systems 69(32), 180–200 (2000)

[2] Biso, A., Rossi, F., Sperduti, A.: Experimental Results on Learning Soft Constraints. In: KR02, pp. 435–444 (2000)

[3] Carenini, G., Moore, J.: Generating and evaluating evaluative arguments. AI Journal 170, 925–952 (2006)

[4] Cohen, W., Schapire, R., Singer, Y.: Learning to order things. Journal of Artificial Intelligence Research 10, 243–270 (1999)

[5] Felfernig, A., Friedrich, G., Schmidt-Thieme, L.: Recommender Systems. IEEE Intelligent Systems 22(3), 18–21 (2007)

[6] Felfernig, A., Friedrich, G., Jannach, D., Zanker, M.: An Environment for the Development of Knowledge-based Recommender Applications. International Journal of Electronic Commerce (IJEC) 11(2), 11–34 (2006)

[7] Felfernig, A., Friedrich, G., Teppan, E., Isak, K.: Intelligent Debugging and Repair of Utility Constraint Sets in Knowledge-based Recommender Applications. In: 13th ACM Intl. IUI Conf., Canary Islands, Spain, January 13-16 (2008)

[8] Felfernig, A., Isak, K., Szabo, K., Zachar, P.: The VITA Financial Services Sales Support Environment. In: AAAI/IAAI 2007, Canada, pp. 1692–1699 (2007)

[9] Felfernig, A., Friedrich, G., Gula, B., Hitz, M., Kruggel, T., Melcher, R., Riepan, D., Strauss, S., Teppan, E., Vitouch, O.: Persuasive Recommendation: Exploring Serial Position Effects in Knowledge-based Recommender Systems. In: de Kort, Y.A.W., IJsselsteijn, W.A., Midden, C., Eggen, B., Fogg, B.J. (eds.) PERSUASIVE 2007. LNCS, vol. 4744, pp. 283–294. Springer, Heidelberg (2007)

[10] Fourer, R., Gay, D., Kernighan, B.: AMPL: A Modeling Language for Mathematical Programming. Cole Publishing Company (2002)

[11] Gershberg, F., Shimamura, A.: Serial position effects in implicit and explicit tests of memory. Journal of Experimental Psychology: Learning, Memory, and Cognition 20, 1370–1378 (1994)

[12] Keeney, R., Raiffa, H.: Decisions with Multiple Objectives: Preferences and Value Tradeoffs. John Wiley and Sons, Chichester (1976)

[13] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: applying collaborative filtering to Usenet news Full text. Communications of the ACM 40(3), 77–87 (1997)

[14] Winterfeldt, D., Edwards, W.: Decision Analysis and Behavioral Research. Cambridge University Press, Cambridge (1986)

Hybrid Integration of Reasoning Techniques in Suspect Investigation

Keehyung Kim, Hyukgeun Choi, and RI (Bob) McKay

Computer Science and Eng., Seoul National University,
599 Gwanak-ro, Gwanak-gu, Seoul, 151-744 South Korea
{keehyung,thegray0}@snu.ac.kr, rimsnucse@gmail.com

Abstract. Crime investigation is a challenging and difficult task, especially if there are many suspects, together with inconsistencies between witnesses, alibis and physical evidence. We propose integration of a rule-based reasoner and a Bayesian network profiler through an Assumption-based Truth Maintenance System (ATMS) to determine the most plausible suspect. The process combines a profiling classification and an alibi credibility measure. It showed effective results in a simulation.

Keywords: Integrated reasoning systems, knowledge-based-system methodology, decision support system, crime investigation.

1 Introduction

Crime investigation is a complex and difficult problem, requiring reliable judgements about many suspects from uncertain and conflicting evidence. Wrong judgements may cause huge damage to innocent people. So all evidence needs to be considered carefully through appropriate methods.

Traditional methods, such as criminal profiling, depend heavily on experienced personnel. There is a shortage of expert investigators able to deal effectively with such complex situations. To assist less-experienced investigators, criminal profiling techniques have been introduced [1]. While based in forensic psychology, they still rely on personal experience, lacking empirical scientific support. Today, the availability of large-scale, structured empirical crime data permits a systematic approach to offender profiling.

In many cases, physical evidence is insufficient to pick out a single suspect. The difficulty is compounded if eyewitness testimony and alibis generate inconsistency. In this situation, people find it difficult to consider multiple alternative hypotheses simultaneously, instead trying to confirm hypotheses one by one [2].

Recently, research in knowledge-based systems has emphasised support for decision making under uncertainty. In crime investigation, where conflicting evidence abounds, well-constructed reasoning systems have been effective in reducing problem spaces and deciding between hypotheses.

Many reasoning techniques have been proposed in support of criminal investigation. Baumgartner et al. [3,4] proposed Bayesian networks for criminal profiling. A Bayesian network is trained from a dataset of offender behavior, and

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 651–660, 2010.

implemented through an inference engine. Ferrari et al. [5] suggested suspect profiling by a neural network. Rogers [6] developed offender-specific profiling to narrow a list of potential suspects.

Human investigators find it difficult to determine a single suspect and a plausible crime scenario when evidence does not support a single hypothesis, generating conflict between many suspects. Probabilisitic offender profiling based on empirical crime data might play an important role in reducing suspect sets, by considering only suspects with high probability. But to include someone as a possible suspect, we also need plausible evidence in support. Our approach integrates suitable knowledge based systems – a rule-based reasoner, a Bayesian Network profiler, and an Assumption-based Truth Maintenance System (ATMS) – for complex crime cases, especially for single-offender, single-victim homicides.

In the following, section 2 surveys the technical background for the model. In section 3, we present our integrated methodology, with section 4 illustrating it through a realistic example. Section 5 presents our conclusions and discusses future work.

2 Background

Reasoning about crime scenarios has received attention from a few previous researchers. Keppens and Zeleznikow [2] presented a decision support system generating plausible crime scenarios from evidence. The central component is an ATMS integrating the space of hypotheses. Keppens and colleagues later explained the need for qualitative probabilistic reasoning [7]. Shen et al.'s scenario-driven decision support system [8] assists investigators by constructing plausible scenarios, reasoning about the best investigative action given the evidence. They use compositional modeling and Bayesian evidence evaluation to create scenario spaces from the evidence.

A variety of methods have been used to integrate different reasoning techniques. Min et al [9] combined rule-based reasoning about game strategy and Bayesian reasoning about sporting outcomes to predict the results of football matches. Jankowska [10] presented a two-level TMS to integrate uncertainty into a reasoning process. Certainty factors are allocated to rules so as to preserve consistency. The approach is illustrated through a medical treatment example.

Bayesian networks(BN) acquire a mathematical model from empirical data using algorithms based on Bayes' rule [11]. Bayesian network learning has thus become one of the most successful methods in criminal profiling. Depending on the kind of data, Bayesian profiling may be categorized into several types – geographical, characteristic, and behavioral. Recent work of Baumgartner et al. [4], constructing Bayesian network profilers with the K2 and K2' algorithms for behavioral profiling, showed high accuracy.

Assumption-based Truth Maintenance Systems (ATMS [12,13,14]) are a powerful and general mechanism for many types of default reasoning. It consists of two components: a problem solver which makes justifications – inferences about the domain – and a TMS which determines the belief in assumptions – problem-solving hypotheses. A set of assumptions are called an environment, and the set of data derivable from the assumptions is the context of the environment. The ATMS associates with every datum a minimal set of environments. If false is derivable, the set of assumptions is inconsistent and labeled as a nogood, so that the combination cannot be used again.

Crime investigation aims to discover the sole truth (or the possibility closest to the truth) among numerous possibilities. The investigation problem lies here: the capacity for collecting evidence, judging the value of the evidence, and reasoning logically about it depends critically on the individual abilities of the investigator. It plunges the whole investigation process into a "maze of uncertainty", with the risk of collecting valueless evidence and making logical mistakes in setting up hypotheses. Thus it is desirable to judge all possibilities and eliminate (or at least reduce) logical faults in the investigation process.

A taxonomy of alibis, as proposed by Olson [15], is an efficient framework for reasoning about the trustworthiness of alibis. It involves two forms of supporting evidence, physical and human. Each is categorized in more detail, based on its ease of fabrication. Physical evidence is divided into three types: None(0), Easy(1), and Difficult(2). Human evidence is likewise divided into four types: None(0), Motivated Familiar Other (1: easy to fabricate, not likely mistaken), Non-Motivated Stranger (2: difficult to fabricate, possibly mistaken), and Non-Motivated Familiar Other (3: difficult to fabricate, not likely mistaken). Thus there are twelve possible combinations in the taxonomy of alibis. Olson noted the different credibility between the two forms of evidences: human evidence has less impact than the highest level of physical evidence. So physical evidence often overwhelms human evidence when they are in conflict.

3 System Description

In this section, we introduce the framework, describing how to integrate a rule-based reasoner and a Bayesian profiler through an ATMS to determine a prime suspect. The flow of interaction in the system is illustrated in figure 1:

1. Investigators feed in text-format information about the crime scene.
2. The rule-based system processes the objective information.
3. The Bayesian profiler generates probabilities for specific suspects.
4. The ATMS calculates the credibility of each suspect, based on the consistency of evidence and a taxonomy of alibis.
5. Combining these measures, the system determines the most plausible suspect as the prime suspect.

The system uses text-format files to record information about the crime – the scene, person, evidence, alibi, and location.

The rule-based reasoner (JESS [18]) pre-processes the data to generate intermediate information for subsequent steps. It collects a list of possible suspects from the crime circumstance (R1). If there is evidence e about person n at location l near the crime scene, n is regarded as a suspect. The list of suspects is transferred to the Bayesian Profiler. It generates two lists (R2): the supporting

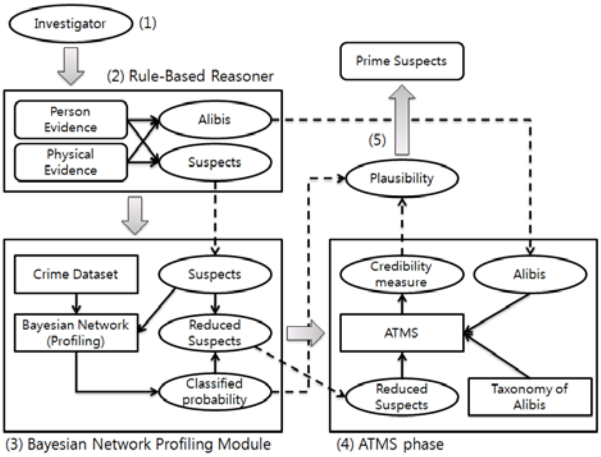


Fig. 1. Framework of the proposed system

scene.txt:	LOCATION NAME WEAPON CIRCUMSTANCE
person.txt:	NAME AGE GENDER RELATION
evidence.txt:	EVIDENCE person|physical TYPE
alibi.txt	EVIDENCE NAME LOCATION
location.txt:	LOCATION near|far

R1 Make a list of suspects from given knowledge by an investigator
$\exists_{n,e,l} Alibi(e,n,l) \wedge Location(l, Near) \rightarrow Suspect(n)$
R2 Make a list of relation between each evidence for each suspect
$\exists_{n,e,l} Suspect(n) \wedge Alibi(e,n,l) \wedge Location(l, Near) \wedge Evidence(e,o,d)$
$\rightarrow Support(e,o,n)$
$\exists_{n,e,l} Suspect(n) \wedge Evidence(e,o,n,l) \wedge Location(l, Far) \rightarrow Alibi(e,o,n)$

evidence and the alibi evidence. Supporting evidence supports the hypothesis that the given suspect is the culprit, while alibi evidence tends to exculpate them. The lists are passed to the ATMS phase.

The Bayesian profiler constructs four BNs of offender characteristics. Table 1 describes five crime variables and four offender variables. A detective can recover the victim's sex, age, and race from the scene. The likely weapon and circumstance are often available from witnesses testimony and autopsy. From these five crime scene variables, it is possible to infer four offender variables – the likely sex, age, race and relation with the victim – through offender profiling.

$$Pr_{su}(ch) = Accuracy(ch) \times Pr_{ch}(Value_{su}(ch)|evidence) \quad (1)$$

$$Pr_{su} = \bigvee_{ch} Pr_{su}(ch) \quad (2)$$

Formula 1 computes the probability (for characteristic *ch*) of the suspect being the offender, based on that characteristic. Formula 2 then infers the combined probability, based on all characteristics.

Table 1. Crime characteristic variables for offender profiling

Variable	Description (Attributes)
victim sex	Gender of the victim (Male, Female)
victim age	0-18, 19-24, 25-30, 31-40, 41-50, 51-99
victim race	White, Black, American Indian, Asian and Pacific Islander
weapon	Firearm, Sharp object, Blunt object, Personal weapon, Strangulation, Poison or drug, Fire or explosion
circum	Circumstance of the crime scene (Sexual, Theft, Fire, Drug, Other Felony, Argument, Lover's triangle, Other non-felony)
offender sex	Gender of the victim (Male, Female)
offender age	0-18, 19-24, 25-30, 31-40, 41-50, 51-99
offender race	White, Black, American Indian, Asian and Pacific Islander
relation	Relationship between victim and offender (Offspring, Spouse or ex-lover, Relatives, Acquaintance, Stranger)

The ATMS reasons about the consistency of all the evidence. It starts from the hypothesis that each suspect is the criminal, and attempts to justify that physical and human evidence supports the hypothesis. By using evidence which does not produce a conflict, we can calculate a credibility measure, under the taxonomy of alibis, for each hypothesis. We use the justification rules in table 2. From these, the system can infer whether particular pieces of evidence are valid or forged. If a nogood node is reached, the corresponding assumption is rejected, and the next assumption process is tried. The assumption step initially assumes that evidence e is valid, but treats it as forged if inconsistency occurs.

$$f_{tax}(x, y) = 1 - 0.25 \times (2 - x) - 0.15 \times (3 - y) \tag{3}$$

$$Cv_{su} = f_{tax}\left(\underset{k \in ph(va(e))}{avg} Type(k), \underset{k \in he(va(e))}{avg} Type(k)\right) \tag{4}$$

$$Cf_{su} = f_{tax}\left(\underset{k \in ph(fo(e))}{avg} Type(k), \underset{k \in he(fo(e))}{avg} Type(k)\right) \tag{5}$$

We determine *credibility* from the taxonomy of alibis[1]. It is computed from formulas 3, 4 and 5, with the notations su:suspect, f_{tax}:alibi taxonomy prediction of credibility, $ph(e)$:physical evidence, $he(e)$:human evidence, $va(e)$:valid evidence, $fo(e)$:forged evidence, and $Type(e)$:type of evidence[2].

Once the ATMS finishes resolving inconsistencies, we calculate the credibility measures for valid and forged evidence, using equations 4 and 5.

$$C_{su} = \frac{Cv_{su}}{Cf_{su}} \tag{6}$$

[1] Alibi credibility ranges from 0.05 (least) to 1.0 (most) using equation 3.

[2] Type of evidence is the value described in section 2.5.

Table 2. Justification Rules for ATMS

$\forall_{e,n} Assume(valid(e)) \wedge Support(e,o,n) \rightarrow suspect(n)$
$\forall_{e,n} Assume(forged(e)) \wedge Alibi(e,o,n) \rightarrow \neg suspect(n)$
$\exists_{n} suspect(n) \wedge \neg suspect(n) \rightarrow nogood$
$\exists_{n,s} primesuspect(n) \wedge suspect(s \neq n) \rightarrow nogood$
$\exists_{e} valid(e) \wedge forged(e) \rightarrow nogood$
$\exists_{e} \neg valid(e) \wedge \neg forged(e) \rightarrow nogood$

The general credibility for each suspect is computed using equation 6. It measures our degree of trust for the hypothesis, based on evidential reasoning.

$$Pl_{su} = Pr_{su} \times C_{su} \tag{7}$$

Finally, the plausibility measure, the most important factor for determining the prime suspect, is computed from the probability measure and the credibility measure using formula 7.

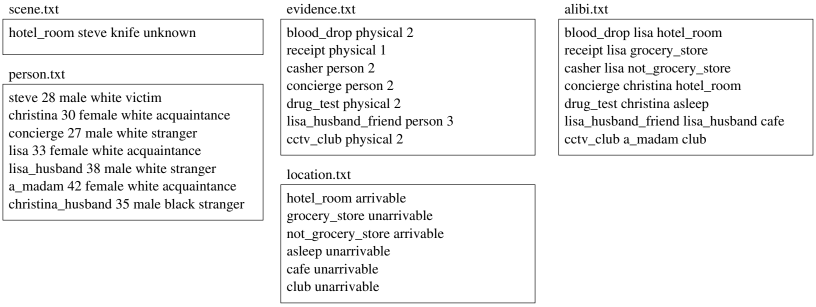
4 Simulation Trace

To illustrate the workings of the system, we use a complex case from a TV series [17]. The story incorporates several suspects, a complex situation, and a lack of sufficient evidence to easily identify the prime suspect. It is very difficult, in this example, to find the prime suspect without reasoning support. Although fictional, it is a good example of a complex case with conflicting evidence.

Steve (28)[3] is found stabbed to death in his hotel room. Christina (30) is lying next to him in bed unharmed, but with no recollection of what happened. During interviews of hotel staff, the concierge (27) attests Christina was with Steve the whole night. The investigators drug-test Christina, finding that Steve probably gave her a narcotic. Then blood drop evidence leads to another housewife, Lisa (33), with a similar story involving Steve; she says she was at a grocery store, providing a store receipt – not a strong alibi, especially as a cashier in the store does not recognize her. The investigators then discover that Steve was hired by a madam (42) to seduce, drug, and rob married women. The madam has a strong alibi, a CCTV shot at a club, for the time of the murder. Since Steve targeted married women, the investigators look not only at his scorned one night stands, but also at their husbands. Lisa's husband (38) has an alibi from a friend. After further investigation, the team finds a military knife used in the homicide, belonging to Christina's husband (35).

The TV story is that Christina's husband knew that Christina was with Steve at the hotel. He went to the hotel making threats, left a knife beside Steve. Later, Lisa came to find Steve to get back the ring Steve robbed her of. Seeing another girl in the same situation angered her sufficiently to kill Steve with the knife lying beside him. Can our system recover this story from the evidence?

[3] The number in parentheses indicates the person's age.



scene.txt

```
hotel_room steve knife unknown
```

person.txt

```
steve 28 male white victim
christina 30 female white acquaintance
concierge 27 male white stranger
lisa 33 female white acquaintance
lisa_husband 38 male white stranger
a_madam 42 female white acquaintance
christina_husband 35 male black stranger
```

evidence.txt

```
blood_drop physical 2
receipt physical 1
casher person 2
concierge person 2
drug_test physical 2
lisa_husband_friend person 3
cctv_club physical 2
```

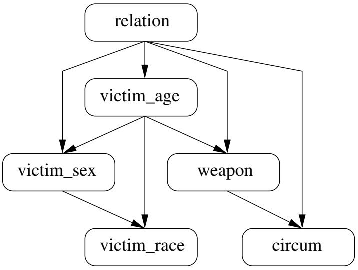
location.txt

```
hotel_room arrivable
grocery_store unarrivable
not_grocery_store arrivable
asleep unarrivable
cafe unarrivable
club unarrivable
```

alibi.txt

```
blood_drop lisa hotel_room
receipt lisa grocery_store
casher lisa not_grocery_store
concierge christina hotel_room
drug_test christina asleep
lisa_husband_friend lisa_husband cafe
cctv_club a_madam club
```

Fig. 2. Crime scene information in text format



$$Pr_{sex}(female|evidence) = 0.552 \quad (8)$$
$$Pr_{age}(31-40|evidence) = 0.268 \quad (9)$$
$$Pr_{race}(white|evidence) = 0.465 \quad (10)$$
$$Pr_{relation}(acquaintance|evidence) = 0.610 \quad (11)$$
$$Pr_{Lisa} = 0.834 \quad (12)$$

Fig. 3. Bayesian Network Profiler for Relation

Fig. 4. Lisa's classified measure based on profiling

We use ICPSR's database [16] of USA offender data, collected over ten years 1993-2002. We filtered out homicide cases (HOMTYPE=1), limited to single-victim/single-offender cases (SITUAT=1), excluding instances with any missing data. This resulted in 55,054 cases in total. Each case includes variables representing the characteristics of offender and victim.

Using the K2 algorithm from the Weka library [19], we trained a Bayesian network classifier for profiling offender characteristics. We set the initial graph as *Naïve graph*[4] and *empty graph* in separate runs, selecting the final BN profiler which has higher test set accuracy. To limit overfitting, we set the maximum number of parent nodes to two. We model four different Bayesian profilers, one for each offender characteristic. Figure 3 shows a typical example.

To find the best profilers, the system divided the database into a training set (70% – 38538 cases) and an independent test set (30% – 16516 cases). Table 3 shows the BN profilers' test-set accuracy for each characteristic. With the exception of offender age, the profilers show highly acceptable accuracy. The prediction of the offender age profiler is not vastly different from the actual value, thus it is still useful.

Figure 2 shows the textual representation of the crime-scene evidence as used in the system. The rule-based reasoner produces a list of suspects for the

[4] The objective node for classification is connected to all the other nodes.

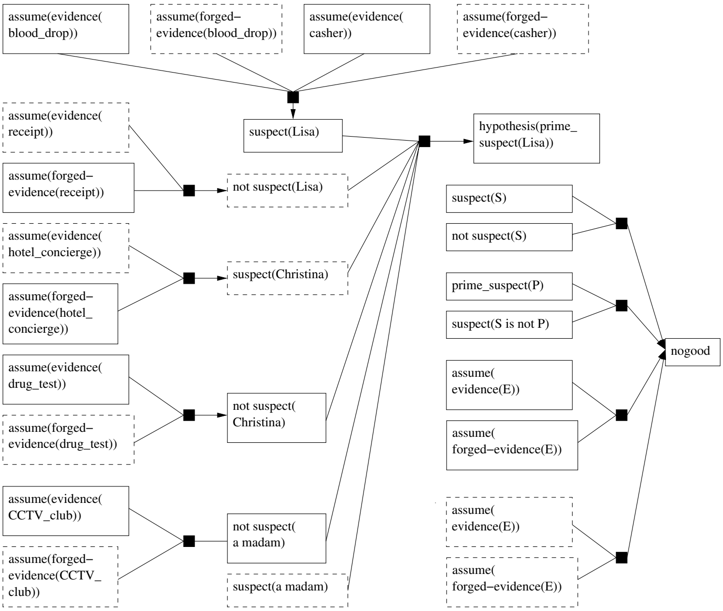


Fig. 5. ATMS example with a hypothesis Lisa as the prime suspect

Bayesian profiler. In this case, we trace the workings of the profiler in the case of Lisa. The gender profiler calculates the probability of a female offender based on the physical evidence, and the victim's sex, age, race, and circumstance. For other offender variables, the corresponding profiler is used. Finally, the overall profiling measure for Lisa as prime suspect is calculated using equations 1 and 2 – see figure 4 for the detailed calculations for Lisa.

At this stage, the investigator can eliminate all but a small number of most plausible suspects. In this example, we set 50% of the suspects as a cutoff, thus the remaining suspects are Christina, Lisa, and the madam (Table 4).

Finding the Prime Suspect. We illustrate the ATMS phase, again with Lisa's case. In figure 5, nodes which can be consistently set true are denoted by solid lines; if setting a node to true results in inconsistency, it is delimited by dotted lines. Evidence from the cashier, blood drops, drug test, and club CCTV support Lisa as the prime suspect, otherwise the receipt would have to be forged and the concierge lying. Thus in total, three pieces of physical evidences and one item of witness evidence support her as the prime suspect, so $Cv_{Lisa} = 0.85$ using the taxonomy of alibis. We can conclude that $Cf_{Lisa} = 0.6$ since two pieces of evidence – one witness, the other physical – imply inconsistency, thus should be nogoods. Finally, the credibility measure of Lisa as the prime suspect is $C_{Lisa} = 1.417 (= Cv_{Lisa}/Cf_{Lisa})$.

The results of the simulation – classification probability measure, credibility measure and plausibility – are shown in table 4. Lisa is the prime suspect,

Table 3. Accuracy of BN Profilers

Characteristic	Accuracy
offender sex (empty)	87.4%
offender age (naive)	38.3%
offender race (naive)	88.5%
relation (empty)	64.6%

Table 4. Result of three measures for each suspect

Suspect	Pr	C	Plausibility
Christina	0.837	0.853	0.714
Lisa	0.834	1.417	1.182
a madam	0.823	0.5	0.412
Hotel concierge	0.759	-	-
Lisa's husband	0.755	-	-
Christina's husband	0.755	-	-

since her plausibility is the highest, even though her classification probablilty measure is lower than Christina's. This illustrates how a credibility measure from probabilistic evidential reasoning using an ATMS can compensate for an incorrect classification probability measure from profiling, and lead to a correct decision. In this hybrid system, BN profiling and ATMS evidential reasoning work together to give complementary reasoning.

5 Conclusion and Future Work

We presented a framework integrating three reasoning systems – rule-based, Bayesian inference, and ATMS – and illustrated how they may be used in criminal investigation. The method of integrating the systems using a classification probability value and a credibility measure is a unique approach. The main contribution of this paper is an effective framework for crime investigation in complex cases. We particularly emphasise the combination of ATMS with the alibi taxomony as a significant contribution.

The main direction for further work is extending this decision support system by incorporating more factors. In the present system, we only used Bayesian profilers for victim and offender factors. Integrating other profiling techniques, such as behavioral profiling, should improve accuracy and thus further reduce the suspect space. We believe it would also be useful to evaluate fuzzy reasoners as an alternative to the Bayesian profiler.

Acknowledgments. We thank ICPSR for making their crime database available for research purposes. This work was supported by the Brain Korea 21 Project. The ICT at Seoul National University provided research facilities for this study.

References

1. Palermo, G., Kocsis, R.N.: Offender Profiling: An Introduction to the Sociopsychological Analysis of Violent Crime. Charles C Thomas Publishers, Springfield (2004)
2. Keppens, J., Zeleznikow, J.: A Model Based Reasoning Approach for Generating Plausible Crime Scenarios from Evidence. In: Proceedings of the 9th International Conference on Artificial Intelligence and Law, pp. 51–59 (2003)

3. Baumgartner, K.C., Ferrari, S., Salfati, C.G.: Bayesian Network Modeling of Offender Behavior for Criminal Profiling. In: 44th IEEE Conference on Decision and Control, pp. 2702–2709 (2005)
4. Baumgartner, K., Ferrari, S., Palermo, G.: Constructing Bayesian networks for criminal profiling from limited data. Knowledge-Based Systems 21(7), 563–572 (2008)
5. Ferrari, S., Baumgartner, K.C., Palermo, G., Bruzzone, R., Strano, M.: Network Models of Criminal Behavior: Comparing Bayesian and Neural Networks for Decision Support in Criminal Investigations. IEEE Control Systems Magazine 28(4), 65–77 (2008)
6. Rogers, M.: The role of criminal profiling in the computer forensics process. Computers and Security 22(4), 292–298 (2003)
7. Keppens, J.: Towards Qualitative Approaches to Bayesian Evidential Reasoning. In: Proceedings of the 11th International Conference on Artificial Intelligence and Law, pp. 17–25 (2007)
8. Shen, Q., Keppens, J., Aitken, C., Schafer, B., Lee, M.: A scenario-driven decision support system for serious crime investigation. Law, Probability and Risk 5(2), 87–117 (2007)
9. Min, B., Kim, J., Choe, C., Eom, H., McKay, R.I.: A compound framework for sports results prediction: A football case study. Knowledge-Based Systems 21(7), 551–562 (2008)
10. Jankowska, B.M.: Howto secure a high quality knowledge base in a rule-based system with uncertainty? International Journal of Applied Mathmatics and Computer Science 16(2), 251–262 (2006)
11. Jensen, F.: Bayesian Networks and Decision Graphs. Springer, New York (2001)
12. de Kleer, J.: An assumption-based TMS. Artificial Intelligence 28(2), 127–162 (1986)
13. de Kleer, J.: Extending the ATMS. Artificial Intelligence 28(2), 163–196 (1986)
14. de Kleer, J.: Problem Solving with the ATMS. Artificial Intelligence 28(2), 197–224 (1986)
15. Olson, E.A., Wells, G.L.: What Makes a Good Alibi? A Proposed Taxonomy. Law and Human Behavior 28(2), 157–176 (2004)
16. Fox, J.A.: Uniform Crime Reports (United States): Supplementary Homicide Reports, 1976-2002 (Computer file). Complied by Northeastern University, College of Criminal Justice. ICPSR ed. Inter-University Consortium for Political and Social Research, Ann Arbor, MI (2005)
17. Cheating Death - CSI:Miami Season 7 Episode 7, http://en.wikipedia.org/wiki/Cheating_Death_CSI:_Miami (retrieved November 26, 2009)
18. JESS, http://www.jessrules.com
19. Weka Library, http://www.cs.waikato.ac.nz/ml/weka/

Ontology-Based Expert System for Home Automation Controlling

Pablo A. Valiente-Rocha and Adolfo Lozano-Tello

Universidad de Extremadura. Área de Lenguajes y Sistemas Informáticos
Escuela Politécnica, Campus Universitario s/n, 10071, Cáceres, Spain
{pvaliente,alozano}@unex.es

Abstract. In the present paper we describe the model process of an ontology-based expert system for the control of domotic installations and SWRL rules management. From the domotic system database where the attribute values of each device are stored, a background process converts these values into ontology's instances representing the system physic devices. From these instances, a software application -known as DomoRules- allows creating production rules in SWRL language that will be useful to regulate the system behaviour. IntelliDomo draws inferences from the ontology and the SWRL rules by using parameters previously indicated by the user, so the state of the physic devices in the domotic system can be modified in real time.

Keywords: Expert System, Ontologies, SWRL, Ambient Intelligence, Domotic, IntelliDomo.

1 Introduction

The use of Ambient Intelligence (AmI) is one of the areas which are rapidly gaining importance in the application of intelligent systems in companies and homes. In the early 90s, with the so-called 'ubiquitous computing', it was suggested that computer and electric systems should be integrated into a physical environment and form part of it. Moreover, they would behave in an intelligent reasonable way [1]. Thus, intelligent environments would be able to manage system mechanisms on their own, regarding several defined criteria, adapting their behaviour to the users' habits and the environment variables.

AmI systems have sensors able to collect information in the environment. These sensors, together with established performance parameters and the fulfilled learning of former performances and cases, are able to make decisions in order to modify the status of the system mechanism for obtaining an optimum behaviour regarding the users' preferences.

One of the characteristics AmI systems seek is the gift of interaction with the user so that communication is as far as possible natural and the configuration of preferences takes less time for the user. The ideal is that the user can finally forget the presence of the system by getting used to domotic devices anticipating to his/her needs and preferences.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 661–670, 2010.

These systems are usually knowledge-based systems containing the specification of domotic elements and they are based on production rules that represent the system's reasoning elements. Upon these grounds, AmI systems use behaving policies able to draw deductions with the information provided by certain devices and the represented rules. Moreover, they make decisions regarding the management of the installation.

A correct way of representing domotic systems and behaving rules is throughout the use of ontologies and production rules based on the concepts established on these ontologies.

For many years, ontologies have generally become the way of representing the systems upon current knowledge [2]. This is due to the fact that they are designed in order to make their knowledge easily reusable and shared by communities and users of the same domain. Besides, the acceptance of OWL (Ontology Web Language) as a de facto language for its representation and the development of tools such as Protégé for its construction have favored its wide use in many fields, especially in the Semantic Web [3].

As far as the representation of production rules is concerned, SWRL (Semantic Web Rule Language) [4] has been adopted by W3C as the representing standard for production rules based on ontologies. SWRL is a very complete language based on OWL, more specifically on the OWL-DL branch. It is a language able to build up rules to perform reasoning about OWL ontology instances (OWL Individuals) and infer new knowledge about them. Moreover, there are several plug-ins for the editing of rules in the Protégé tool, which have favored its spreading and its use. There are currently some reasoners based on OWL and SWRL that allow maintaining consistency and inferring values for the attributes of classes in the ontologies.

The use of ontologies as representing bases, together with the representation of rules in SWRL in AmI systems, will provide the most precise definition of the taxonomy of physical devices that may exist in a system, the attributes of these devices and the possible relations among them. Furthermore, these representations will be more reusable by other users and will favor the completion of the classification of domotic components, the useful information to be represented upon these components and the connection rules that will allow deducting new information regarding the values of other components.

The present paper describes IntelliDomo, an ontology-based AmI system for the control of domotic systems. It uses domotic components and its state values represented as instances of an ontology, and takes advantage of the power of the production rules specified by the user in order to change the state of the system components in real time. In section 2 of the present paper, we will identify some publications using ontologies in domotic systems. In section 3, we will describe the architecture of IntelliDomo and, in its subsections, we will describe each of the components and tools from the model software. In section 4, we will propose a laboratory example to show the use of the model. Finally, section 5 is dedicated to conclusions and future lines.

2 Existent Works about the Use of Ontologies on AmI Systems

The design of expert systems for installations and buildings controlling is not new. Nevertheless in recent years a lot of initiatives for AmI systems controlling and representing are emerging from both research team and domotic companies which in most cases try to establish their own standard [5]. However, the amount of literature about intelligent domotic environments, which makes use of ontologies as representing bases, is poor. DomoML is one of the main reference works [6] [7]; it is a markup language focused on defining a communication method among domotic devices. Three ontologies are proposed -DomoML-Env, DomoML-Fun and DomoML-Core- as a base to build an architecture that allows setting up hierarchies and position and categorize domotic devices. DomoML operates as a communication protocol among devices and as a representation language to describe resources and functionalities.

Related to the previous project, DogOnt [8] proposes a system able to recognize the devices that comprise the domotic environment automatically. For this end, DogOnt uses two essential elements: DogOnt Ontology, which allows formalizing all aspects from Intelligent Domotic Environment (IDE); and DogONT Rules, which supplies model process and allows associating physics devices to appropriate ontology instances through semantic relationships.

DogOnt uses SWRL language to define rules that allow completing the ontology model which represents the domotic environment. In this way, throughout modeling process, a designer can instantiate needed domotic devices and execute Jess inference engine to obtain a complete model, thus avoiding inconsistence issues.

In these projects ontologies are used as representing bases of domotic devices and SWRL rules are used to maintain consistency in the information of the system components values. However, these production rules are not used to control general autonomous and real time functioning of the system. Compared with previous projects, our proposal is based on both, a comprehensive ontology, which defines the taxonomy of home automation devices, and production rules that are used to establish the engine inference behaviour.

3 Ontology-Based Model Process for Home Automation Controlling

This paper describes IntelliDomo, an expert system able to control the components of a domotic system automatically and in real time. The overview model was presented at [9]. It is based on an ontology called OntoDomo which contains the information about the devices of the system, on production rules in SWRL language which shape its behaviour and on a series of interconnection software tools, rules design and inference motor (Figure 1).

The main characteristic of IntelliDomo is the ability of making decisions and reacting to the changes that arise in both, the values of the OWLIndividuals' properties and the status of the physical devices. These decisions are taken based on the state of the system elements represented in OntoDomo and also on a set of established rules, expressed in the SWRL rules definition language. These rules can be modified by the user with the use of the DomoRules tool presented in the following subsection. In order to draw knowledge deductions, IntelliDomo uses Jess inference motor, this tool will allow inferring the concepts of the ontology and obtaining new data according to what it is stated in SWRL rules.

In order to manage this knowledge, IntelliDomo is built upon an ontology whose concepts are related to the domotic components. The ontology has been modeled to be in sync with the physic devices that form a domotic environment, so that it can store its outstanding values and properties. The use of ontologies as a basis for the specification of home automation components can be very useful due to their characteristics for a complete and precise representation of terms. These characteristics allow us to structure this knowledge and categorize each device accurately. Thus, the representation of domotic data scheme will be easier and will require less effort to reuse and adapt this information in future software systems.

IntelliDomo has been designed to interact with an existing domotic database (DomoBD), where the state and values of the domotic components are updated in real time. This database can be obtained throughout the QDSConnect software module that has been developed by the research team from the Universidad de Extremadura, Quercus. This daemon provides the physic connection with EIB/KNX bus. QDSConnect uses FALCON library to translate the values of the physic devices into the relational database system and to detect any changes that may occur in the mentioned database in order to modify it straight in the physic devices.

Therefore, IntelliDomo can make deductions and update the domotic system instantly by transferring the new values from its hardware components to the database. DomoBD is the link between IntelliDomo and the physic domotic system. DBConnection Module constantly translates the information of this database into instances of OntoDomo. Similarly, when IntelliDomo fires the correspondent rules, DBConnection Module updates DomoBD database, which will entail the respective changes of state in the physic devices.

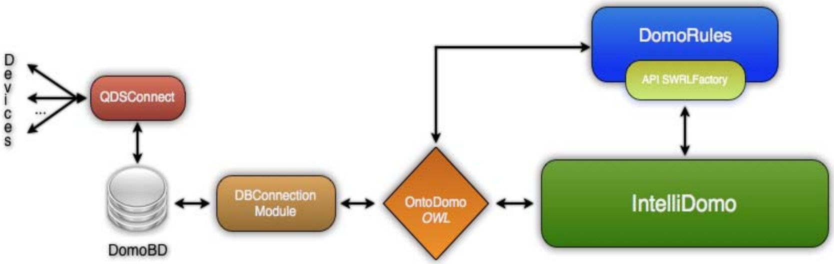


Fig. 1. Overview of IntelliDomo architecture

3.1 OntoDomo Ontology

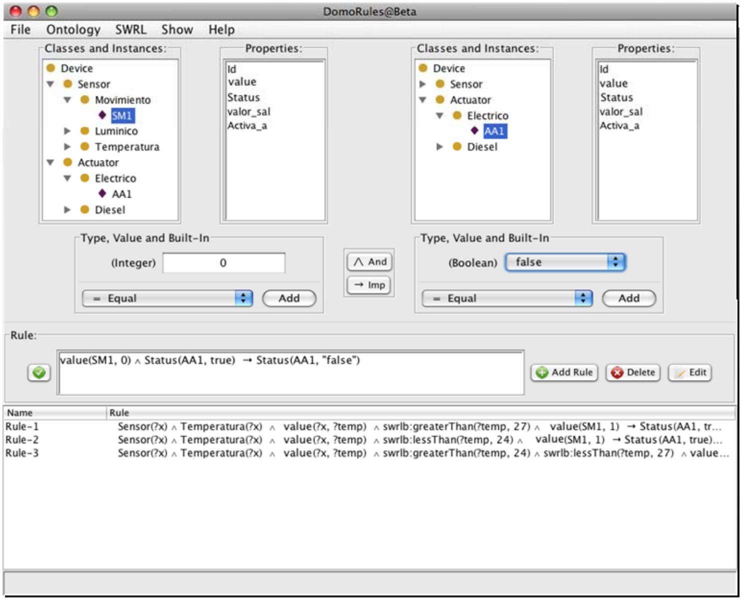
Nowadays, ontologies are the most used way of knowledge representation in different business or research projects in fields such as databases, intelligent information integration, cooperative information systems, information retrieval, electronic commerce, enterprise application integration, and knowledge management [10]. IntelliDomo uses ontologies as a representing base of domotic elements that constitute the system.

These elements are stored in OntoDomo.owl, which is responsible for reusing the knowledge of the rest of ontologies constituting IntelliDomo. IntelliDomo allows the users to establish configurations able to guide the system behaviour and that are adapted to its needs and preferences. In order to do so, it uses an ontology that allows storing the configuration into profiles and to make inferences and reasoning able to incorporate new knowledge to the system. The ontology Preferences.owl is used with this end. Here, each user will be able to store his/her personal information for a certain situation: security, temperature, comfort... A series of variables defining the user's preferences can be established for every situation. For instance, as for temperature, the user will be able to specify that ‘hot’ ranges between 25 ºC and 36 ºC, consequently, these values will be used in the corresponding production rules.

Besides, the ontology Preferences.owl controls the so-called ‘system variables’. These are control variables which could be used in SWRL rules to work with them and draw inferences. These variables work as global variables that will be shared by each user and known by the inference system in order to use them as a reference for certain decisions. These variables are identified with the symbol ‘#’ prior to the name.

3.2 DomoRules Tool

DomoRules is a tool developed in Java that facilitates the SWRL rules creation throughout API SWRLFactory. It provides a graphic interface appropriate to assist in the construction of these rules. It allows obtaining from OntoDomo all the necessary



value(SM1, 0) ∧ Status(AA1, true) → Status(AA1, "false")

Name	Rule
Rule-1	Sensor(?x) ∧ Temperatura(?x) ∧ value(?x, ?temp) ∧ swrlb:greaterThan(?temp, 27) ∧ value(SM1, 1) → Status(AA1, tr...
Rule-2	Sensor(?x) ∧ Temperatura(?x) ∧ value(?x, ?temp) ∧ swrlb:lessThan(?temp, 24) ∧ value(SM1, 1) → Status(AA1, true)...
Rule-3	Sensor(?x) ∧ Temperatura(?x) ∧ value(?x, ?temp) ∧ swrlb:greaterThan(?temp, 24) ∧ swrlb:lessThan(?temp, 27) ∧ value...

Fig. 2. DomoRules creation rules graphic interface. The interface shows data from the main ontology (OntoDomo.owl). It is divided into two sections: Antecedent and Consequent. The user can select any class or any class property and add it to the rule’s antecedent or rule’s consequent.

elements to build a SWRL rule (classes, instances, properties), to use SWRL variables and system variables that refer to the values defined by the user and finally to add SWRL language built-ins to the rules.

With DomoRules we seek to build SWRL rules easily without knowing the syntax of this language. All the elements constituting OntoDomo.owl ontology are introduced to the user in the interface, so that, by simply choosing a class or an instance, all the properties it has will be listed and the user will be able to pick the one he/she needs and establish its values in order to build the rules.

DomoRules is born aiming to offer the user a simple SWRL rules creating application. The task of designing an intuitive interface to abstract the SWRL language user is not easy because SWRL is a powerful language with a very rich vocabulary that provides high interaction with all its elements. Accordingly, by using DomoRules the user will be able to produce rules that shape the system behaviour that could be used by IntelliDomo. The user can use the knowledge of several ontologies for modeling the same rule because DomoRules can access each ontology in the system through IntelliDomo's API.

3.3 IntelliDomo Module

IntelliDomo is the main control module of the system using the represented elements and the different modules, which constitute the system. The information about configuration and location of the system can be observed graphically on a plan and, when the state of a device changes, a message alerts appearing upon the device icon. At the same time these events happen, it stores a set of log files where each of the actions carried out by the system and all the inferences it makes are detailed. As it was mentioned above, it uses the DBConnection module to get connected to the different databases that constitute the application, which can be either DomoBD or the private databases IntelliDomo uses to store its own data.

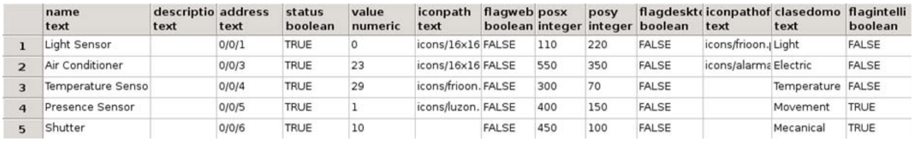
4 IntelliDomo System Example Usage

In order to understand how this system works, this section describes a simple example that has been tested in the laboratory. The installed components are both, two sensors (*temperature sensor and presence sensor*) and an actuator (*a wall plug able to trigger air conditioning*). These elements have been configured using ETS v3 tool and their state information is reflected on DomoBD database (see figure 3) with the help of QDSConnect application. From this table, IntelliDomo's DBConnection module is able to read steadily the table fields (*name*, *status*, *value*, *posx*, *posy*, *flagdesktop*, *clasedomo*, *flagIntelli*) and transform them into OntoDomo instances with their respective values for this three-device domotic installation.

OWL code example that represents Presence_Sensor OWL Individual. Each property has a value that has been obtained from DomoDB database. Thus, IntelliDomo automatically generates each ontology instance.

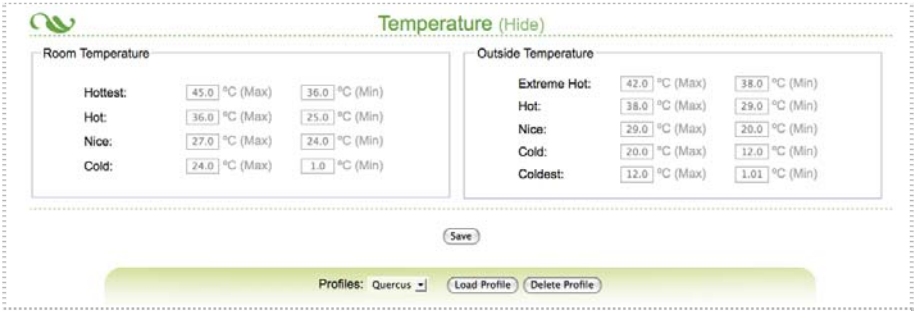
```
<Movement rdf:ID="Presence_Sensor">
  <status rdf:datatype="http://www.w3.org/2001/
  XMLSchema#boolean">true</status>
  <value rdf:datatype="http://www.w3.org/2001/
  XMLSchema#int">1</value>
  <out_value rdf:datatype="http://www.w3.org/2001/
  XMLSchema#int">0</out_value>
  <posx rdf:datatype="http://www.w3.org/2001/
  XMLSchema#int">400</posx>
  <posy rdf:datatype="http://www.w3.org/2001/
  XMLSchema#int">150</posy>
  <Id rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
  >Presence_Sensor</Id>
</Movement>
```

Previously, the user has defined some global situations that help IntelliDomo to manage environmental values. For example, users can define that a ‘nice’ temperature value is when temperature is between 24 ºC and 27 ºC. Moreover, this information will be stored into Preferences.owl ontology. Users can use the web interface to create new configuration profiles or to modify the existing ones. Each user will have stored his/her own values for each profile. (See Figure 4)



	name	descriptio	address	status	value	iconpath	flagweb	posx	posy	flagdeskt	iconpathof	clasedomo	flagintelli
	text	text	text	boolean	numeric	text	boolean	integer	integer	boolean	text	text	boolean
1	Light Sensor		0/0/1	TRUE	0	icons/16x16	FALSE	110	220	FALSE	icons/frioon.	Light	FALSE
2	Air Conditioner		0/0/3	TRUE	23	icons/16x16	FALSE	550	350	FALSE	icons/alarma	Electric	FALSE
3	Temperature Senso		0/0/4	TRUE	29	icons/frioon.	FALSE	300	70	FALSE		Temperature	FALSE
4	Presence Sensor		0/0/5	TRUE	1	icons/luzon.	FALSE	400	150	FALSE		Movement	TRUE
5	Shutter		0/0/6	TRUE	10		FALSE	450	100	FALSE		Mecanical	TRUE

Fig. 3. DomoBD database that contains the data used in the example. Every row in this table, represents a real physical device.



Temperature (Hide)

Room Temperature

Hottest:	45.0 °C (Max)	36.0 °C (Min)
Hot:	36.0 °C (Max)	25.0 °C (Min)
Nice:	27.0 °C (Max)	24.0 °C (Min)
Cold:	24.0 °C (Max)	1.0 °C (Min)

Outside Temperature

Extreme Hot:	42.0 °C (Max)	38.0 °C (Min)
Hot:	38.0 °C (Max)	29.0 °C (Min)
Nice:	29.0 °C (Max)	20.0 °C (Min)
Cold:	20.0 °C (Max)	12.0 °C (Min)
Coldest:	12.0 °C (Max)	1.01 °C (Min)

Profiles: Quercus

Fig. 4. Temperature preferences configuration window. Each profile becomes an OWL Individual of the ontology Preference.owl.

This simple example consisting of four rules that have been defined using DomoRules interface (see figure 2); these rules link the installation devices among them for the purpose of maintaining the laboratory temperature in a ‘nice’ value taking into account that the user has to be inside laboratory. Once the rules have been set, DomoRules exports them to SWRL language (see next) so that they can be used by IntelliDomo's rule engine inference.

This code has been created with the help of DomoRules Tool. When DomoRules finds something like #T_USU_Nice, it has to look in Preference.owl ontology to find the appropriate value and finally complete the SWRL sentence.

```
1. Temperature(?x) ^ Mecanical(?y) ^ value(?x, ?temp) ^
   swrlb:greaterThan(?temp, #T_USU_NiceMax) ^ value(SM1,1) ^
   value(?y,?level) ^ swrb:greaterThan(?level, 5) ➔ Status(AA1,
   true) ^ value(AA1, #T_USU_Nice) ^ value(?y, 2)
2. Temperature(?x) ^ value(?x, ?temp) ^ swrb:lessThan(?temp,
   #T_USU_NiceMin) ^ value(SM1, 1) ^ Mecanical(?y) ^
   value(?y,?level) ^ swrb:lessThan(?level, 5) ➔ Status(AA1,
   true) ^ value(AA1, #T_USU_Nice) ^ value(?y, 8)
3. Sensor(?x) ^ Temperature(?x) ^ value(?x, ?temp) ^
   swrlb:greaterThan(?temp, #T_USU_NiceMin) ^
   swrlb:lessThan(?temp, #T_USU_NiceMax) ^ value(SM1, 1) ➔
   Status(AA1, false)
4. value(SM1, 0) ^ Status(AA1, "true") ➔ Status(AA1, false)
```

With the previous representations, IntelliDomo will continuously check the properties' values of the four devices; meanwhile, changing logs from their values are being monitored in the log interface. When IntelliDomo detects any change on any device values, it updates its knowledge base with these new values and runs Jess inference rule engine to evaluate the established production rules and fire the respective ones. When this occurs, it would update its knowledge base and send the new data to DomoBD database. Finally, QDSConnect daemon would detect these changes and send them to EIB/KNX bus to update physic domotic devices. These actions are carried out in a loop while IntelliDomo is running.

5 Conclusions and Future Lines

The design of smart environments able to adapt to the concept of Ambient Intelligence is a complicated area because a lot of external factors can affect the behaviour of the system and because that system has to adapt to the users' preferences. Nonetheless, the significant amount of physic devices on the market and the lack of a standard method to control them make the design of intelligent domotic environments considerably difficult.

Ontologies provide an appropriate kind of representation for identifying types and characteristics of domotic devices. Furthermore, production rules expressed in SWRL language allow establishing relationships among these domotic devices to shape the integral behaviour of a domotic installation.

In the present paper, we describe IntelliDomo. On the one hand, it is based on an ontology -known as OntoDomo- where all types of domotic devices, together with their useful characteristics, are represented; and, on the other hand, it is based on production rules in SWRL language defined according to the users' preferences and needs. With these knowledge bases, IntelliDomo allows managing the control of the domotic system itself. The state of the elements that comprise a determined domotic installation is continuously read from the database where its values are stored and translated into instances of OntoDomo ontology. With this information, together with the rules defined by the user, IntelliDomo's inference engine would fire the appropriate rules that will change the state of the system devices.

In addition to the IntelliDomo's control module, we have developed DomoRules application, which works as a SWRL production rules generating wizard. The interface design has been developed aiming to be simple and intuitive for every user. As a result, the user may only select the type of component (a concrete physic component in the system) and the appropriate values to build the condition of the rule antecedent (body). Furthermore, the user will have to indicate which values should be modified in the system devices when the rule is fired.

IntelliDomo application and DomoRules rule generator are fully operative. Moreover, both have been tested in the laboratory with some domotic devices and with production rules designed with the help of the interface, accordingly, some state devices are automatically modified when other devices fulfill an established rule antecedent. Model architecture is surely complete and its functioning already proved. However, two interesting items set as objectives of the project are still to be established: i.e. classification of behaviour aspects and system learning.

With respect to the classification of behaviour aspects, classes defined in OntoDomo (domotic elements type) should be categorized with aspects that IntelliDomo should consider and ponder in order to fire one rule or another. Aspects such as 'temperature', 'security', 'anticipation', 'consumption', etc. must be established and domotic elements able to influence these aspects must be identified. Thus, the user can be offered a behaviour template where he can define and ponder his performance rules.

Nevertheless, we are inclined to provide IntelliDomo with a learning module where production rules would be modified according to the historical procedure of the users. The intention is that, meanwhile the user interacts with the system or some determined values are obtained under certain conditions, the rules can change or their execution precedes other rules. Therefore, once the user has established the reasonable importance of behaviour aspects, rules can be modified in order to adapt the system behaviour according to the importance of these criteria.

Acknowledgements

This work has been developed under support of Junta de Extremadura Project (PDT08A023).

References

1. Weiser, M.: The computer for the 21st century. Scientific American (September 1991)
2. Fensel, D.: Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce. Springer, Berlin (2001)
3. Berners-Lee, T.J., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, 29–37 (May 2001)

4. Horrocks, I., Patel-Schneider, P., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML (May 2004), http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/
5. Gasson, M., Warwick, K.: D12.2: Study on Emerging AmI Technologies. FIDIS Deliverables 12(2) (2007)
6. Sommaruga, L., Perri, A., Furfari, F.: DomoML-env: an ontology for Human Home Interaction. In: Proceedings of SWAP (2005)
7. Furfari, F., Sommaruga, L., Soria, C., Fresco, R.: DomoML: The definition of a standard markup for interoperability of human home interactions. In: Proceedings of the 2nd European Union symposium on Ambient Intelligence (2004)
8. Bonino, D., Corno, F.: Dogont-ontology modeling for intelligent domotic environments. In: 7th International Semantic Web Conference, pp. 790–800 (2008)
9. Valiente, P., Lozano-Tello, A.: Control Model of Domotic Systems based on Ontologies. In: 2nd International Conference on Agents and Artificial Intelligence. Valencia, Spain (January 2010)
10. Hepp, M., De Leenheer, P., de Moor, A., Sure, Y. (eds.): Ontology Management. Semantic Web, Semantic Web Services, and Business Applications, vol. 7. Springer, Heidelberg (2008)

Ontology Applied in Decision Support System for Critical Infrastructures Protection

Michał Choraś[1,2], Rafał Kozik[2],
Adam Flizikowski[1,2], and Witold Hołubowicz[1,3]

[1] ITTI Ltd., Poznań
michal.choras@itti.com.pl
[2] Institute of Telecommunications, UT&LS Bydgoszcz
chorasm@utp.edu.pl
[3] Adam Mickiewicz University, Poznań
holubowicz@amu.edu.pl

Abstract. In this paper, a decision support engine based on ontology for Critical Infrastructure Protection is presented. The proposed ontology provides vulnerabilities classification and and their relationships with other security aspects. Described approach is used in decision support tool developed within the INSPIRE project (INcreasing Security and Protection through Infrastructure REsilience). Our application called INSPIRE DAT (Decision Aid Tool) is also presented.

1 Introduction

SCADA is an acronym for Supervisory Control and Data Acquisition. SCADA systems are used to monitor and control a plant or equipment in industries or critical infrastructures such as water and waste control, energy oil and gas refining and transportation. Rapid success of information and communication systems had significant influence on developing new way of controlling and managing critical infrastructures. Those have been rapidly moved from dedicated solutions for particular operator to integrated and IP-based frameworks. However the evolution exposed the SCADA for cyber attacks and unauthorized access of hackers. This requires a new approach to critical infrastructure protection which will engage expert knowledge, decision support systems and such network elements as firewalls, intrusion and anomaly detection systems. That was not a case when the systems controlling critical infrastructures were created. Critical Infrastructure Protection (CIP) including cyber defense is one of the crucial security and safety aspects in EU [1]. In order to cope with those problems we propose approach based on ontology driven decision support system for SCADA security assessment. The role of the proposed DSS (called INSPIRE Decision Aid Tool - DAT) is to provide the SCADA operator system with all the necessary information about the threats and vulnerabilities the specific critical infrastructure is exposed to. Additionally, DAT can propose appropriate reactions and countermeasures for the particular threat.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 671–680, 2010.

1.1 Decision Support Systems for CIP

Decision Support Systems (DSS) are information systems that support human in different decision-making activities. The DSS applications are successfully and widely used in industry and critical infrastructure protection (CIP).

In 1987 Texas Instruments company released GADS (Gate Assignment Display System) decision support system for United Airlines. As a result, the travel delays have been reduced significantly. The system was aiding the management of ground operations at various airports. Another good example of successfully deployed decision support applications are expert systems in the banking area (expert systems for mortgages). The decision support systems are also widely used for river systems management to effectively cope with floods. For example, The German Federal Institute of Hydrology (BfG) funded the development of a Decision Support System for the Elbe river system. The great flooding in summer 2002 demonstrated the importance of such solutions. Some examples of DSS used in the energy sector are described in [2]. DSS are also successfully deployed in nuclear power plants [3], urban water pollution control [4] or oilfield flood precaution [5].

All these DSS examples are customized and focused on some particular branch of critical infrastructures. Decision Support Systems are usually designed for special kind of industry or application. Although they use different methodologies (Bayesian, multiagent, Hidden Markov Models), they rarely use ontologies description to support reasoning.

INSPIRE Decision Aid Tool (DAT), presented here, is on the other hand, more general and applicable to more than one critical infrastructure. We focus rather on the used SCADA properties to enhance protection and security of critical sectors. INSPIRE Decision Aid Tool may be also considered as a framework since it is reconfigurable by means of uploading other ontologies or SCADA system topology.

2 INSPIRE Security Ontology

The major goal of the INSPIRE Security Ontology is to describe interdependencies between particular assets, vulnerabilities, threats (with defined level of risk), and safeguards. Main approach to security ontology is based on ISO/IEC 133351:2004 standard [6]. According to the standard, vulnerabilities are considered as properties of a network security system. In this approach SCADA assets and components have weak points named vulnerabilities. These vulnerabilities can be exploited by threats, leading to attacks. This security system is depicted into a form of classification with properties and relationships between security issues [7].

2.1 Vulnerability Description Standards

There are several approaches to describe vulnerabilities. Design vulnerabilities differ from implementation vulnerabilities (i.e. application faults) on which NVD

(National Vulnerabilities Database) is focused. The proposed Security Ontology is based on the CVE (Common Vulnerabilities and Exposures) vulnerability naming standard and uses the following SCAP (Security Content Automation Protocol [10]) standards:

- Common Configuration Enumeration (CCE [14])
- Common Platform Enumeration (CPE [11])
- Common Vulnerability Scoring System (CVSS [12])

The Common Configuration Enumeration provides common identifiers to system configurations in order to facilitate fast and accurate correlation of configuration data across multiple information sources and tools. CCE is primarily used to identify security related configuration issues. The Common Platform Enumeration is a structured naming scheme for information technology systems, software, and packages. Finally, the Common Vulnerability Scoring System is an open standard for assigning a score to a vulnerability that indicates its relative severity compared to other vulnerabilities.

The use of such standards enables automated vulnerability management, measurement, and policy compliance evaluation.

2.2 The Proposed Ontology Structure

Main concepts, which compose main classes of proposed ontology on the basis of the ISO/IEC 13335-1:2004 standard are [6]:

- Assets (anything that has value to the organization)
- Vulnerabilities (include weaknesses of an asset or group of assets which can be exploited by threats)
- Threats (potential cause of an unwanted incident which may result in harm to a system or organization)
- Source of attacks
- Safeguards (practices, procedures or mechanisms that reduce vulnerabilities)

These classes are connected to each other by properties. Properties can show relations, dependence of one class on another or can represent some attributes. The Vulnerabilities class properties are as follows:

- are exploited by (Threats)
- category CWE (String)
- CPE name (String)
- CVE id (String)
- CVSS access complexity (String: Low, Medium, High, Insufficient Information)
- CVSS access vector (String)
- CVSS authentication (String)
- CVSS availability (String)
- expose asset (Asset)

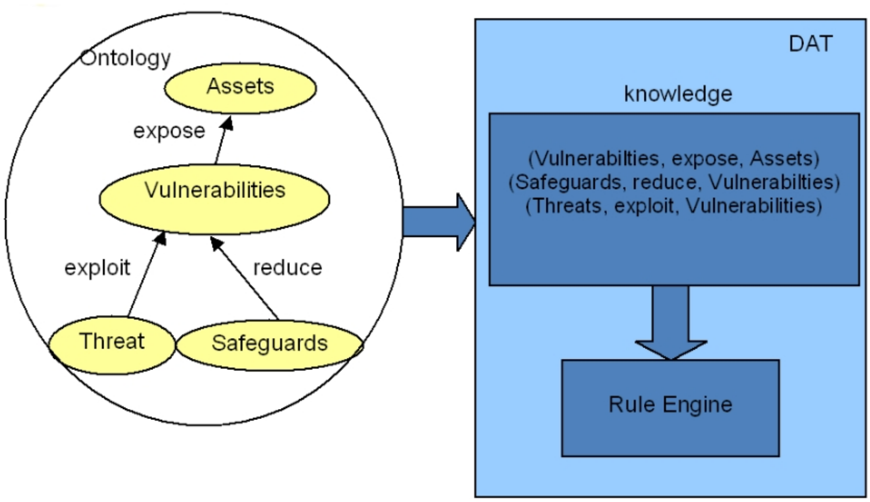

Ontology
Assets
expose
Vulnerabilities
exploit
reduce
Threat
Safeguards
DAT
knowledge
(Vulnerabilties, expose, Assets)
(Safeguards, reduce, Vulnerabilties)
(Threats, exploit, Vulnerabilities)
Rule Engine

Fig. 1. DAT and ontology interoperability

- have safeguard (Safeguard)
- have attack vector (String)
- have risk level (String: Low, Moderate, Height, Critical)

The properties allow to model interconnection between particular Vulnerabilities and between Vulnerabilities and Assets. The Assets class properties are:

- has vulnerability (Vulnerability),
- depends on (Application),
- runs on (Operating System),
- installed on (Hardware),
- connected to (Network).

The Safeguards properties allows to model relation between Vulnerabilities and Safeguards. The properties of this class are:

- reduces (Vulnerability),
- name.

The Threat class properties are:

- is caused by (Source of attack),
- is related to (Asset),
- exploits (Vulnerability).

3 INSPIRE Decision Aid Tool (DAT)

The proposed ontology mimics the complicated relationships between SCADA components and security aspects. That is why this ontology can support security solutions. However, the ontology is just a representation of relationships between particular classes (or instances) and as it is, cannot provide any knowledge-based reasoning or give feedback to its operator.

3.1 Ontology Mapping

The proposed ontology cooperates with Decision Aid Tool (DAT) as it is shown in Figure 1. However, the format the ontology is maintained is not directly accepted by the inference engine and requires mapping. Such an engine consists of knowledge and rules (production rules).

The idea was to map instances (and also relation between instances) stored in ontology into facts and SWRL rules into production rules and perform reasoning using inference engine (In this case JESS inference engine) [16]. DAT uses ontology classes and instances to get the knowledge as RDF triples and process them in rule engine. Each RDF triple consists of [17]:

- Subject
- Predicate
- Object

Each triple is able to fully describe one property of the instance. The interpretation of a triple is that "subject" has property "predicate" whose value is "object". Such strategy allows DAT to be more flexible to ontology schema changes, because adding new property to the particular instance has no impact on mapping

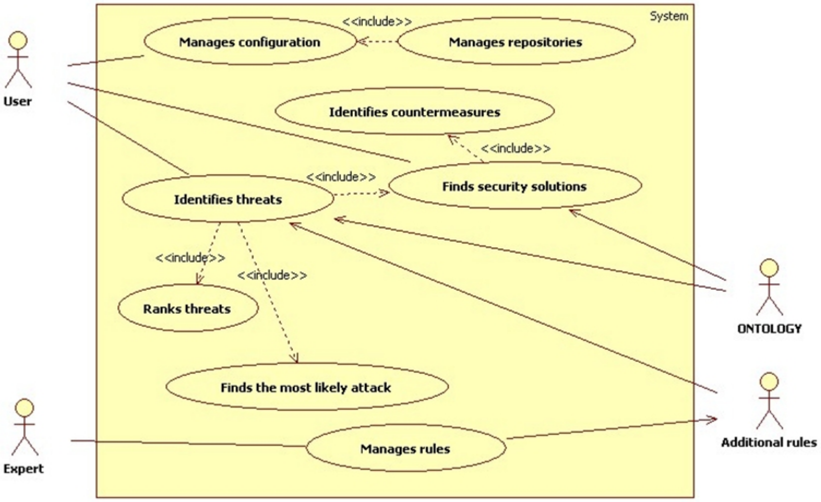


Fig. 2. DAT use cases

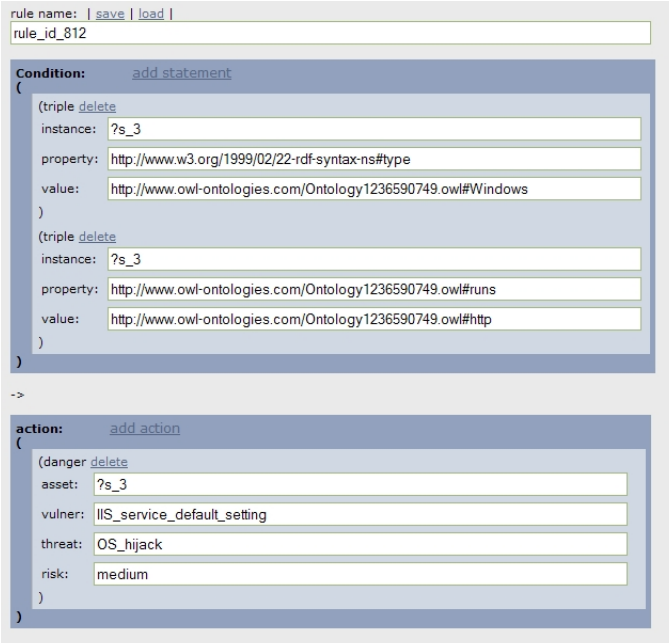
mechanism (new triple is handled like the other ones) and no impact on DAT source code.

In example the relation ”Asset x hasVulnerability y” is mapped into (triple (subject x) (predicate ’hasVulnerability’) (object y)). The SWRL rules are mapped into production rules as follows:

$$hasParent(?x1, ?x2) \wedge hasBrother(?x2, ?x3) \Rightarrow hasUncle(?x1, ?x3)$$

will be mapped into:

(defrule rule-1
(triple (predicate ’hasParent’) (subject ?x1) (object ?x2))
(triple (predicate ’hasBrother’) (subject ?x2) (object ?x3))
⇒
(assert (triple (predicate ’hasUncle’) (subject ?x1) (object ?x3))))



rule name:
rule_id_812

Condition:
(
(triple
instance: ?s_3
property: http://www.w3.org/1999/02/22-rdf-syntax-ns#type
value: http://www.owl-ontologies.com/Ontology1236590749.owl#Windows
)
(triple
instance: ?s_3
property: http://www.owl-ontologies.com/Ontology1236590749.owl#runs
value: http://www.owl-ontologies.com/Ontology1236590749.owl#http
)
)

->

action:
(
(danger
asset: ?s_3
vulner: IIS_service_default_setting
threat: OS_hijack
risk: medium
)
)

Fig. 3. GUI for expert rules generation

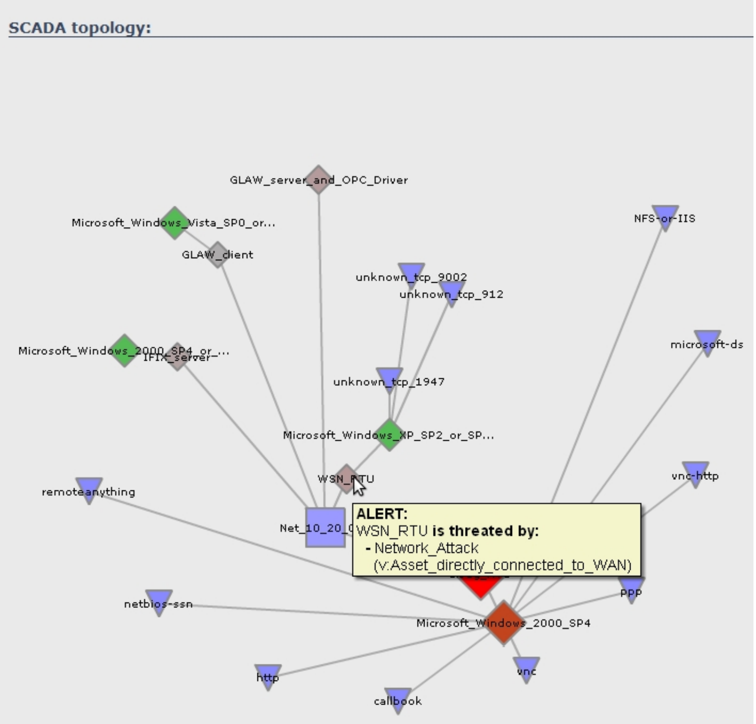
3.2 Ontology Engagement for Security Evaluation

Thanks to the ontology and the mapping technique, described in 3.1, Decision Aid Tool is facilitated with information that allows to:

- dynamic threat ranking (the value is not hard coded into ontology)
- inform operator what may cause fault in SCADA system
- asses which element in the network needs some attention and safeguard
- detect fault and security holes
- suggest topology and configuration modification to minimize the threat of an attack
- identify interconnection between assets and provide user with it graphical representation
- generate security reports

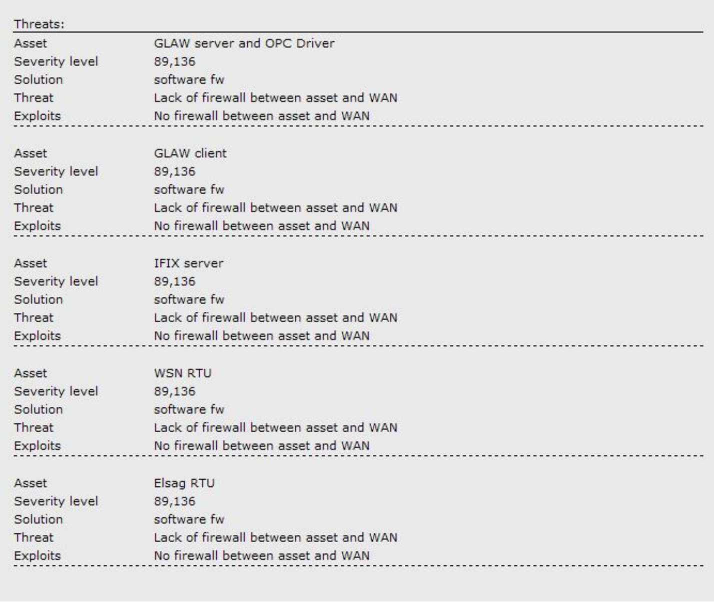
Our design of the tool specifies following type of users (Figure 2): a plain user (which has no expert knowledge about the SCADA system and uses the DAT to identify SCADA system vulnerabilities and threats) and an expert user (provides tool with additional customized knowledge). The expert rules allow to enrich the information stored in ontology and customize the tool behavior for particular deployment. Thanks to the GUI (Figure 3) provided for expert user it is easy to define such rules, validate their syntax and test their behavior.

The rules have been divided into following groups of concerning: topology issues, configuration faults, general issues. The "topology issues" group allows to



ALERT:
WSN_RTU **is threated by:**
- Network_Attack
(v:Asset_directly_connected_to_WAN)

Fig. 4. Topology diagram with the marked threats



```
Threats:
Asset             GLAW server and OPC Driver
Severity level    89,136
Solution          software fw
Threat            Lack of firewall between asset and WAN
Exploits          No firewall between asset and WAN
-----------------------------------------------------------------------

Asset             GLAW client
Severity level    89,136
Solution          software fw
Threat            Lack of firewall between asset and WAN
Exploits          No firewall between asset and WAN
-----------------------------------------------------------------------

Asset             IFIX server
Severity level    89,136
Solution          software fw
Threat            Lack of firewall between asset and WAN
Exploits          No firewall between asset and WAN
-----------------------------------------------------------------------

Asset             WSN RTU
Severity level    89,136
Solution          software fw
Threat            Lack of firewall between asset and WAN
Exploits          No firewall between asset and WAN
-----------------------------------------------------------------------

Asset             Elsag RTU
Severity level    89,136
Solution          software fw
Threat            Lack of firewall between asset and WAN
Exploits          No firewall between asset and WAN
-----------------------------------------------------------------------
```

Fig. 5. Security report with the ranked threats

inform the operator about faults that were found in relations between particular assets. In example: "If asset RTU is connected directly to WAN then raise an alert and suggest the reconfiguration". The "configuration issues" rules inform the user about the configuration faults and suggest a reconfiguration (e.g. "use SSH instead of telnet", "use WPA instead of WEP", etc.). The last group is used to maintain threat ranking strategy, topology diagram (Figure 4) or security report generation, and other rules that concerns the tool behavior.

The sample result of CI system security evaluation by DAT is presented in Figure 5. The presented security report contains the ranked threats for discovered assets with their threat severity value. Moreover, in the security report, details about the detected threats and the proposed solutions are given.

4 Conclusion

In this paper an ontology-based approach for the description of SCADA systems vulnerabilities has been presented. Functionalities of the ontology applied to knowledge representation for security and protection of critical infrastructures have been shown. Our solution has been developed as a research activity in the INSPIRE Project that aims at increasing security and protection through infrastructure resilience. The presented ontology approach is the basis for the presented Decision Aid Tool (DAT) as a part of INSPIRE security framework. INSPIRE DAT works in two modes and can be used both for real system security assessment as well as for security simulations. In the latter case DAT can be used to verify what happens in terms of security when a particular component is added to the existing system.

Acknowledgment

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 225553 (INSPIRE Project).

References

1. European Parliament legislative resolution on the proposal for a Council directive on the identification and designation of European Critical Infrastructure and the assessment of the need to improve their protection (COM(2006)0787 C6-0053/2007 2006/0276(CNS)) (July 10, 2007)
2. Xiao Feng, D., Yu Jiong, G., Kun, Y.: Study on Intelligent Maintenance Decision Support System Using for Power Plant Equipment. In: Proc. of the IEEE International Conference on Automation and Logistics Qingdao, China, September 2008, pp. 96–100 (2008)
3. Lee, S.J., Mo, K., Seong, P.H.: Development of an Integrated Decision Support System to Aid the Cognitive Activities of Operators in Main Control Rooms of Nuclear Power Plants. In: Proc. of IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (MCDM), pp. 146–152 (2007)
4. Zhang, B., Wu, G., Shang, S.: Research on Decision Support System of Water Pollution Control Based On Immune Agent. In: Proc. of International Symposium on Computer Science and Computational Technology, ISCSCT, vol. 1, pp. 114–117 (2008)
5. Xie, L., Wang, Z., Bian, L.: The Research of Oileld Flood Precaution Decision Support System. In: Proc. of International Seminar on Business and Information Management, ISBIM '08, December 2008, vol. 2, pp. 236–239 (2008)
6. ISO/IEC 13335-1:2004, Information Technology Security Techniques Management of information and communications technology security Part 1: Concepts and models for information and communications technology security management
7. Choras, M., Stachowicz, A., Kozik, R., Flizikowski, A., Renk, R.: Ontology-based approach to SCADA systems vulnerabilities representation for CIP. Electronics 11, 35–38 (2009)
8. Choras, M., Flizikowski, A., Kozik, R., Renk, R., Holubowicz, W.: Ontology-Based Reasoning Combined with Inference Engine for SCADA-ICT Interdependencies, Vulnerabilities and Threats Analysis. In: Pre-Proc of 4th International Workshop on Critical Information Infrastructures Security, CRITIS'09, Bonn, Germany, pp. 203–214, Fraunhofer IAIS (2009)
9. http://cve.mitre.org/
10. http://scap.nist.gov/
11. http://cpe.mitre.org/
12. http://www.first.org/cvss/
13. http://cwe.mitre.org/
14. http://nvd.nist.gov/cce.cfm

15. http://www.jessrules.com/
16. SWRL: A Semantic Web Rule Language Combning OWL and RuleML, W3C Member Submission, http://www.w3.org/Submission/SWRL/
17. Deliverable D2.3, Ontological approach and inference engine, INSPIRE Project (2009)
18. Macaulay, T.: Critical infrastructure: Understanding Its Component Parts, Vulnerabilities, Operating Risks, and Interdependencies (August 2008)
19. Lewis, T.G.: Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation. Wiley-Interscience, Hoboken (2006)
20. McClanahan, R.H.: The benefits of networked SCADA systems utilizing IP-enabled networks. IEEE, Los Alamitos (2002)

Learning User Preferences to Maximise Occupant Comfort in Office Buildings

Anika Schumann, Nic Wilson, and Mateo Burillo

Cork Constraint Computation Centre, University College Cork, Ireland

Abstract. It is desirable to ensure that the thermal comfort conditions in offices are in line with the preferences of occupants. Controlling their offices correctly therefore requires the correct prediction of their thermal sensation which is often determined using the ISO 7730 norm. The latter defines the predicted mean vote, i.e. the mean thermal preferences of an average group of people, based on a number of variables that are either difficult to measure in practice or require the placement of many sensors in the offices of a building, which is very costly. This paper addresses these issues and predicts the comfort preferences of users solely based on the temperature readings and their previous comfort votes. In order to determine how relevant the latter are to a new state a distance measure is defined that quantifies the similarity between two states. Based on that similarity the previous votes are weighted and the expected comfort vote for the new state is determined. The paper concludes with an experimental analysis using real field study data that show under which climatic conditions our approach outperforms existing approaches.

Keywords: Intelligent systems, thermal comfort, decision support systems.

1 Introduction

Several studies show that a comfortable working environment will lead to increased productivity and reduced sick leave days [8,11,12]. Hence, ensuring, or even maximising, occupant comfort is an important goal in the operation of any office building. Some of these buildings allow the individual control of room temperature that enable the occupants to create optimal comfort conditions for themselves [13]. There are also a number of approaches that control the rooms automatically according to the individual preferences of the occupants [3,4,10].

However, these approaches assume that the user comfort feedback can be acted upon straight away and thus either that the office is occupied by a single user only or by several users with identical comfort preferences. Since it is very unlikely that a group of occupants have the same preferences, shared offices or meeting rooms are mostly controlled automatically according to well defined standards that catch the thermal comfort needs of about 80% of the occupants [1,2]. These standards are based on average criteria for population comfort under the guideline of the widely used Predicted Mean Vote (PMV) and Predicted Percentage of People Dissatisfied (PPD) indexes [6]. The PMV value is defined

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 681–690, 2010.

in terms of four environmental variables (indoor air temperature, mean radiant temperature, relative air velocity, and humidity), and two personal variables (activity level and the value of clothing worn).

PMV based approaches have some drawbacks: They require many environmental data whose retrieval is costly due to the sensors needed, and they require precise personal dependent data which are often difficult to obtain in practice. Our approach tackles these issues. It takes only the temperature sensor readings and the comfort votes of the occupants into account and defines a distance measure to quantify the similarity between two states. Based on the distance to a new state the comfort votes of the existing states are weighted and the predicted comfort vote for that new state is determined.

The paper is organised as follows: Section 2 describes existing approaches for predicting the thermal comfort vote of occupants. Our algorithm is presented in Section 3 and Section 4 compares the performance of that algorithm with respect to the existing ones. The paper is concluded in Section 5.

2 Thermal Comfort and the Predicted Mean Vote

Thermal comfort is often measured using the predicted mean vote (PMV) (adapted by ISO 7730 [2]) which is defined as the heat load that would be required to restore a state of 'comfort' for an average group of people. It depends on six variables: the two personal dependent variables: activity level and the value of clothing worn by the users and the four environmental dependent variables: indoor air temperature, mean radiant temperature, relative air velocity, and humidity. For any state w for which these six variables are known the PMV value can be computed as shown in Algorithm 1[1].

Algorithm 1. GetPMVvalue(w)

```
  return $PMV(w)$
```

The algorithm for computing the PMV value is taken from the ISO norm 7730 [2].

A room is said to have optimal thermal conditions if its PMV value is zero. Depending how well environments provide these conditions they are classified into the three desirable categories below [2]:

category	PMV
A	$-0.2 <$ PMV < 0.2
B	$-0.5 <$ PMV < 0.5
C	$-0.7 <$ PMV < 0.7

However, these conditions are only desirable if the PMV value correctly predicts the comfort votes of users. Field studies on thermal comfort have shown that this model does not give correct predictions for all environments [7]. The

[1] The Algorithm is stated here since it will be used in Section 4.

work of [9] picks up this aspect and presents an approach that adjusts the PMV value depending on the climatic conditions. Here the predicted comfort vote for a state is computed by resorting to a database DB in which comfort votes of users in the same climate zone are recorded along with the PMV values of the states in which these votes were taken. Formally, each element of DB is of the form (q, w, v), where

- q is the occupant,
- w is the state,
- v is the comfort vote on the seven point ASHRAE scale [1] that the occupant q provided in state w, i.e. $v = 3$ (resp. 2,1,0,-1,-2,-3) if his/her vote was "I'm hot." (resp. warm, slightly warm, comfortable, slightly cool, cool, cold).

As shown in Algorithm 2, this database is used to determine the difference between comfort votes and the corresponding PMV value (see line 5) in order to adjust future predicted votes accordingly (line 6).

Algorithm 2. PMVadj: Compute predicted occupant vote $\hat{v}$ for state w'

```
1: Input:
      DB consisting of tuples of the form $(q, w, v)$
2: if DB is empty then
3:    $\hat{v} := PMV(w')$
4: else
5:    $PMVdif := \sum_{(q,w,v) \in DB} (PMV(w) - v)$
6:    $\hat{v} := PMV(w') - \frac{PMVdif}{|DB|}$                $|DB|$ : number of entries in DB
7: return $\hat{v}$
```

The two presented algorithms both aim at correctly predicting the comfort votes of occupants. This allows one to determine the conditions in buildings that meet the thermal comfort needs of its users. Typically these conditions are then enforced by regulating the operative temperature depending on the other parameters that influence the PMV value. These other parameters therefore need to be measured precisely which is costly and possibly difficult in practice.

3 Learning of Individual User Preferences

The aim of this section is to describe an approach that predicts the comfort votes of occupants at least as good as the previous two algorithms but that, in contrast to the latter, depends only on the temperature measurements and the comfort votes which the occupants have chosen to give. It thus only needs to resort to a database DB in which each element δ is of the form (q, t, v) where t denotes the room temperature at which the vote v of occupant q was recorded. In order to determine how relevant a vote is to a new situation s two *distance* measures are defined. One describes the degree of similarity between two states and the other one describes the degree of similarity between two occupants.

3.1 Defining the Degrees of Similarity

Naturally, the degree of relevance of a database entry $\delta = (q, t, v)$ to a situation $s = (q', t')$ is the higher the more similar q is to q' and t is to t'. Formally, the distance measure for temperature readings ν_{temp} and for occupants ν_{occ} are defined as follows:

$$\nu_{temp}((q,t,v),(q',t')) := |t - t'| \text{ and}$$

$$\nu_{occ}((q,t,v),(q',t')) := \begin{cases} 1 & \text{if } q = q' \\ 0 & \text{otherwise.} \end{cases}$$

Based on these distance measures the relevant database entries can be identified.

3.2 Quantifying the Relevance of Database Entries to New Situations

The following describes a parameterised approach for defining the degree of relevance of an element δ to a situation s. The specific parameter choices and their robustness is discussed in the next section. The degree of relevance is obtained from two normalised functions R_1 and R_2. The former considers only the elements in the database referring to the same occupant, and fairly similar temperature. The latter considers only the elements in the database with a very similar temperature, but based on other occupants.

Specifically, $R_1(\delta, s) := 0$ if $\nu_{occ}(\delta, s) = 0$ or if the temperature readings differ by more than $c_1 > 0$ degrees. Otherwise, we have:

$$R_1(\delta, s) := \frac{1}{K_s}(\nu_{temp}(\delta, s) + 1)^{-2},$$

where K_s is a normalisation constant given by:

$$K_s = \sum_{\delta \in DB \text{ with } \nu_{occ}(\delta,s)=1 \text{ and } \nu_{temp}(\delta,s) \le c_1} (\nu_{temp}(\delta, s) + 1)^{-2}.$$

Database entries of other occupants are only considered if they were taken at a very similar temperature that differs no more than $c_2 > 0$ degrees from the current situation.

$$R_2(\delta, s) := \begin{cases} \dfrac{1}{N_2} & \text{if } \nu_{occ}(\delta, s) = 0 \text{ and } \nu_{temp}(\delta, s) \le c_2 \\ 0 & \text{otherwise,} \end{cases}$$

where N_2 is the normalisation constant that is equal to the number of entries δ in DB with $\nu_{occ}(\delta, s) = 0$ and $\nu_{temp}(\delta, s) \le c_2$.

3.3 Weighting of the Two Relevance Functions

The intention is that when there is sufficient relevant data about the occupant q' of the new situation, then this data is largely used to predict his/her comfort vote also for the new situation. In contrast, if there are only few such data recorded then the new comfort vote is mainly retrieved from data regarding other occupants at similar temperature settings.

In order to achieve the desired behaviour, we used the following definition for the weighting function μ:

$$\mu := \begin{cases} 1 - 0.8 \left(\frac{N_2}{N_2 + c_4} \right) \left(\frac{c_3}{N_1 + c_3} \right) & \text{if } N_1 > 0 \\ 0 & \text{otherwise} \end{cases}$$

where N_1 denotes the number of database entries with $R_1(\delta, s) > 0$. μ ranges between 0 and 1 and is used to compute the degree of relevance R of entries in the database as follows:

$$R(\delta, s) := \mu R_1(\delta, s) + (1 - \mu) R_2(\delta, s).$$

If the number N_1 of relevant items relating to the current occupant is large (much larger than c_3) then μ is close to 1, which implies that mostly the relevant data of the current occupant is taken into account. Hence, R is approximately just R_1. Similarly, if there is only a little relevant data for the current occupant but much for the others then R is approximately just R_2.

3.4 Predicting of Comfort Votes Based on Relevant Database Entries

Naturally, the more relevant an entry in the database is, the more impact its comfort vote should have on the computation of the predicted vote $\hat{v}$. Therefore, the latter is obtained as the R-weighted average of the recorded votes. This results in:

$$\hat{v} := \sum_{(q,t,v) \in DB} R((q,t,v),(q',t')) \cdot v.$$

The complete computation of this vote is detailed in Algorithm 3.

4 Experimental Evaluation

4.1 Identifying Suitable Field Studies

We have compared the presented Algorithm 3 with the two existing Algorithms 1 and 2 (see Section 2) using field studies gathered by de Dear and Brager [5]. Their collection contains 52 studies with over 20,000 individual comfort votes from different countries. However, some of these field studies contain only few votes per user and are thus not well suited for testing our algorithm. This results from the fact that the latter seeks to learn the user preferences based on their votes, and thus it requires sufficiently many data records. Hence, we considered only field studies that have on average at least five comfort votes per user. Also, for predicting the average vote for shared offices with n people no studies could be

Algorithm 3. Compute predicted occupant vote $\hat{v}$ for situation $s = (q', t')$

```
Inputs:  DB consisting of tuples of the form $\delta = (q, t, v)$
         $N_1$ : number of entries $\delta$ in DB for occupant $q'$ with $|t - t'| \leq c_1$
         $N_2$ : number of entries $\delta$ in DB for occupants $q \neq q'$ with $|t - t'| \leq c_2$

One-off computations (see Subsections 3.2 and 3.3):
$K_{(q',w')} := \sum_{(q,t,v) \in DB \text{ with } q=q' \text{ and } |t-t'| \leq c_1} \frac{1}{(|t - t'| + 1)^2}$
if $N_1 \neq 0$ then
    $\mu := 1 - 0.8 \left( \frac{N_2}{N_2 + c_4} \right) \left( \frac{c_3}{N_1 + c_3} \right)$
else
    $\mu := 0$

For each $(q, t, v) \in DB$ compute $R((q, t, v), (q', t'))$ as follows:
                                                  (see Subsections 3.2 and 3.3)
if $q = q'$ and $|t - t'| \leq c_1$ then
    $R((q, t, v), (q', t')) := \frac{\mu}{K_{(q',w')} \cdot (|t - t'| + 1)^2}$
else if $q \neq q'$ and $|t - t'| \leq c_2$ then
    $R((q, t, v), (q', t')) := \frac{1 - \mu}{N_2}$
else
    $R((q, t, v), (q', t')) := 0$

Result retrieval (see Subsections 3.4):
if DB is empty then
    $\hat{v} := 0$
else
    $\hat{v} := \sum_{(q,t,v) \in DB} R((q, t, v), (q', t')) \cdot v$
```

used that involved less than n occupants. Furthermore, some of the field studies contain votes that were collected for conditions with a PMV value beyond ±3. Since the ASHRAE scale does not extend beyond these limits we removed the comfort votes taken in extreme conditions with $PMV(w) > 3$. This is in line with the analysis in [7] that did not consider such votes either.

After removing the studies and records as described above we were left with 16 field studies from 8 climate zones containing 7781 comfort votes from 519 different users. Each of these votes is associated with the state in which it was taken that is also composed of the room temperature and the PMV value. Hence, these studies contain all the data that are needed as inputs for the three algorithms.

4.2 Computing the Predicted Average Vote for Realistic Scenarios

In practice, comfort votes are collected iteratively and databases are build up step by step. We have therefore also chosen an iterative approach to test the three algorithms. Specifically, we iteratively added the records of each study to the database used by the algorithms. When adding a record, we first added all its entries apart from the comfort vote. Then we used the existing database DB to predict the vote for that situation and computed the deviation Δv from the actual vote. After that we also added the actual vote to the database in order to use it for future computations.

Initially the database DB is empty and thus for the first iteration the predicted vote $\hat{v}$ equals $PMV(w)$ for Algorithms 1 and 2, and 0 for Algorithm 3. After that, when adding the k^{th} record to the database we computed the predicted vote based on the available k-1 records. This allows the prediction of comfort votes for individuals and thus to control a single person office in a way that maximises his/her comfort.

In order to determine how the algorithms perform when used for determining the predicted average vote of n users sharing the same office we sorted the records of each field study according to their PMV value. Then we grouped the records in blocks of size n. This ensures that records of one block refer to similar states and hence can be used for predicting the average vote for shared offices in which all occupants are exposed to similar environmental conditions. Similar to the single person office approach we then iteratively add these blocks to our database. Every time a block is added we computed for each of the n occupants the predicted comfort vote as described above, determined the average of these votes, and computed the difference ΔV from the average of the actual votes.

Note, the order in which the records or blocks of a field study are added to the database can play a significant role, especially when occupants have voted quite differently in similar states. For each field study we have therefore generated 15 random orders of adding their blocks to the database. The presented results depict average values over these 15 runs.

4.3 Defining the Evaluation Criteria for the Experimental Analysis

The three algorithms are evaluated with respect to how accurately they predict the average comfort votes. In line with the classification of buildings into the three categories (see page 682) we distinguish the following three accuracy levels:

Accuracy level	Accuracy label	ΔV
A	precise	$\Delta V < 0.2$
B	correct	$\Delta V < 0.5$
C	approximation	$\Delta V < 0.7$

For instance, we say that a vote was computed correctly if it deviates less than 0.5 units from the actual vote. In that case it would have correctly predicted the discrete value on the seven point ASHRAE scale.

4.4 Experimental Results

Since the field studies were conducted in different places we ran the algorithms separately on each of them. Our approach was run with the following constants:

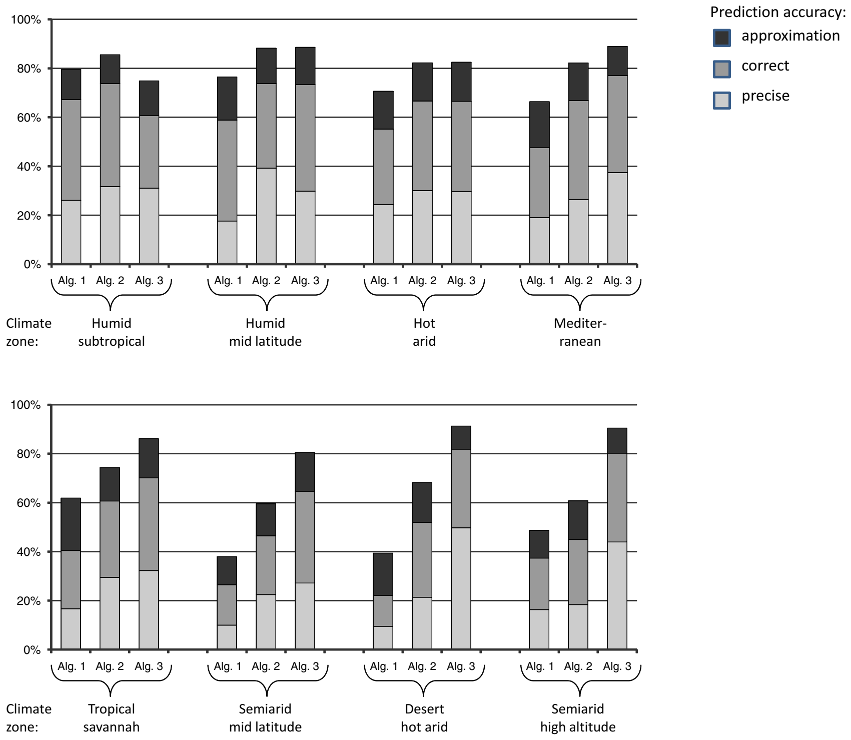


Fig. 1. Average accuracy of predicting user comfort for offices shared by 5 people in different climate zones

$c_1 = 1$, $c_2 = 0.25$, $c_3 = 10$, and $c_4 = 100$ (see below for an evaluation of their robustness). Figure 1 illustrates how accurately the three algorithms predict the average comfort vote for an office shared by five people. It shows that only for humid subtropical conditions our algorithm performs worse than the others. For that study it precisely predicts the average vote in 31% of cases. 61% (resp. 75%) of the predictions are still correct (resp. are a close approximation). For the field studies of this climate zone it turned out that the best predictions can be made using Algorithm 2 that has 32%, 73%, and 86% of its predictions in category A, B, and C, respectively. On the other hand, considering the field studies conducted in semiarid high altitude conditions, it turns out that our algorithm significantly improves the prediction accuracy in relation to the existing algorithms. Here, 46%, 83%, and 92% of predictions belong to categories A, B, and C, as opposed to 18%, 45%, and 61% for the best existing algorithm.

The experiments have also shown that for our algorithm the prediction accuracy does not so much depend on the climate zone as the existing algorithms do. Hence, our approach should particularly be used for the environmental conditions identified in [7] for which the PMV model does not give precise predictions. Furthermore, it turned out that the performance of our approach depends more

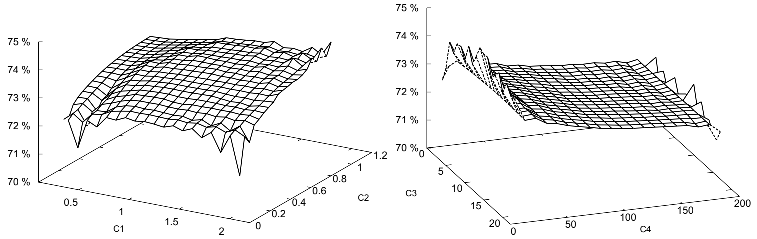


Fig. 2. Average percentage of correct comfort vote predictions for offices shared by 5 people for different parameter values

on the size of the database than on the climate zone. For instance, the worst results were obtained for humid subtropical conditions whose data were recorded in databases with an average size of 163 entries. In contrast, all the databases of the climate zones shown on the bottom of Figure 1 had at least 400 records.

Finally, we analysed to what extent the results of Figure 1 depend on carefully chosen values for the four constants of our approach. For this purpose we considered the following variable ranges: $0.2 \leq c_1 \leq 2$, $0.1 \leq c_2 \leq 1$, $c_2 \leq c_1$, $1 \leq c_3 \leq 20$, and $10 \leq c_4 \leq 200$. As Figure 2 shows the overall results hardly vary with changing parameter values. In fact, the average variance is less than 1. Our experiments also revealed that our approach always outperforms the existing methods for Mediterranean conditions and all the climate zones shown on the bottom of Figure 1 for any of the parameter settings within the above ranges. For most of these ranges our algorithm also outperforms existing methods for humid mid latitude and hot arid climate zones although there exist some parameter choices like $c_1 = 1$, $c_2 = 0.1$, $c_3 = 15$, and $c_4 = 30$ for which our approach performs worse. For humid subtropical conditions our approach can only keep up with the other algorithms with respect to precise predictions. For the other accuracy levels our approach performs always worse regardless of the parameter choices.

5 Conclusions

We have presented a new algorithm that predicts the comfort votes of occupants with high accuracy while requiring few sensor data only. Our experimental evaluation has shown that only for one of the considered eight climate zones, namely for humid subtropical conditions, our algorithm performed slightly worse than the existing ones. For the remaining field studies for which at least five comfort votes per user were recorded our method allowed both: an improve in prediction accuracy and a reduction of the required sensor data. The latter implies that the deployment of our approach in practice would lead to big savings compared to the use of existing PMV based approaches. This results also from the fact that the data that is additionally required by our algorithm, the comfort votes of occupants, can be cheaply gathered using available equipment like desktop computers or mobile phones.

Acknowledgments

This material is based upon works supported by the Science Foundation Ireland within the SRC project ITOBO. We also thank those whose field studies were included in the database, and also of those who compiled it and made it freely available: Richard de Dear, Gail Brager, Donna Cooper.

References

1. ANSI/ASHRAE Standard 55-2004. Thermal environmental conditions of human occupancy (2004)
2. IS0/DIS 7730:2003. Ergonomics of the thermal environment – Analytical determination and interpretation of thermal comfort using calculation of the PMV and PPD indices and local thermal comfort (2003)
3. Baus, Z.L., Nikolovski, S.N., Maric, P.Z.: Process control for thermal comfort maintenance using fuzzy logic. Electrical Engineering 49(1), 34–39 (2008)
4. Dalamagkidis, K., Kolokotsa, D., Kalaitzakis, K., Stavrakakis, G.S.: Reinforcement learning for energy conservation and comfort in buildings. Building and Environment 42(7), 2686–2698 (2007)
5. de Dear, R.J.: A global database of thermal comfort field experiments. ASHRAE Transactions 104(1), 1141–1152 (1998)
6. Fanger, P.O.: Thermal Comfort: Analysis and Applications in Environmental Engineering. McGraw-Hill, NY (1972)
7. Humphreys, M.A., Nicol, J.F.: The validity of ISO-PMV for predicting comfort votes in every-day thermal environments. Energy and Buildings 34(6), 667–684 (2002)
8. Karjalainen, S., Koistinen, O.: User problems with individual temperature control in offices, building and environment. Building and Environment 42(8), 2880–2887 (2007)
9. Kumar, S., Mahdavi, A.: A combined analytical and case-based approach to thermal comfort prediction in buildings. In: Proceedings of Building and Simulation Conference, pp. 369–376 (1999)
10. Liang, J., Du., R.: Design of intelligent comfort control system with human learning and minimum power control strategies. Energy Conversion and Management 49(4), 517–528 (2008)
11. Lorsch, H., Abdou, O.: The impact of the building indoor environment on occupant productivity-part 2: effects of temperature. ASHRAE Transactions 100(2), 895–901 (1994)
12. Roelofsen, P.: The impact of office environments on employee performance: The design of the workplace as a strategy for productivity enhancement. Facility Management 1, 247–264 (2002)
13. Wyon, D.P.: Individual control at each workplace: the means and the potential benefits. Creating the productive workplace, 192–206 (2000)

A Decision Support Tool for Evaluating Loyalty and Word-of-Mouth Using Model-Based Knowledge Discovery

Benoît Depaire*, Koen Vanhoof, and Geert Wets

Transportation Research Institute
Faculty of Business Economics
Hasselt University
3590 Diepenbeek – Belgium
benoit.depaire@uhasselt.be, koen.vanhoof@uhasselt.be,
geert.wets@uhasselt.be

Abstract. It is crucial for any manager to keep a close watch on customer satisfaction, customer loyalty and the customer's intention to recommend the company. In this article, a new decision support tool is developed to support a manager with this task. This tool has been developed with companies in mind that posses limited customer satisfaction data. It uses model-based knowledge discovery to extract the customer's expectation and the expectation-performance compatibility from the data. Two hypotheses are formulated which posit that compatibility between product performance and customer expectation have a positive influence on the customer's intentions. Both hypotheses are supported by the data. Finally, a decision support tool is developed which visualizes the impact of customer satisfaction, product performance and expectation-performance compatibility on the customer's intentions. The decision support tool contains two views which offer the manager important information at a glance.

1 Introduction

Customer satisfaction, customer loyalty and other customer intentions such as word-of-mouth are widely recognized as important features of a competitive business [12]. Therefore it is crucial for managers to keep a close watch on customer satisfaction, customer loyalty and the customer's intention to recommend the company. To this end, a new decision support tool (DST) is developed in this article to support managers to focus on the most efficient criteria to improve loyalty and word-of-mouth. The DST only requires customer satisfaction, product performance, loyalty and word-of-mouth data, and extracts new information from the available data with a model-based knowledge discovery approach which is discussed in the next section. The most important new construct extracted from the data is expectation-performance compatibility. According to the Decision Support System (DSS) categorization framework of Power [9], the DST

* Corresponding author.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 691–700, 2010.

in this article is a function-specific knowledge based DSS targeted at managers and other decision makers. In the subsequent sections, the data set used for this research is presented and the relationships between the compatibility construct and customer loyalty and word-of-mouth intentions are analyzed. Finally, the DST is introduced and applied on data from two separate companies, and the final conclusions are presented.

2 Model-Based Knowledge Discovery from Limited Customer Satisfaction Data

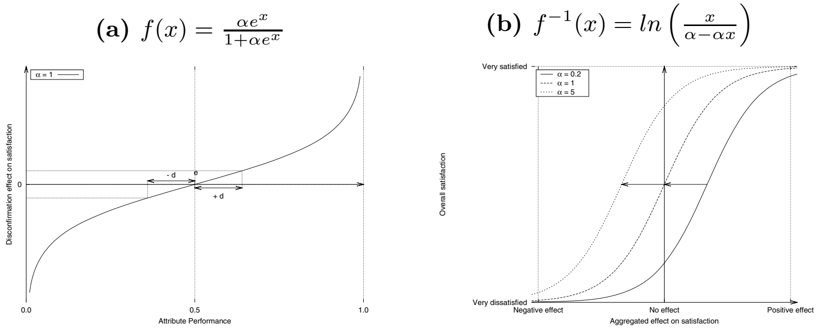
The DST developed in this article is particularly useful for companies that only collect basic information regarding customer satisfaction. More specifically, this article assumes that a company only possesses data about customer satisfaction, product performance, customer loyalty and word-of-mouth behavior. Furthermore, it is assumed that product performance is measured for several product dimensions, e.g. a car's performance is evaluated for the dimensions 'driving comfort', 'fuel consumption' and 'luxury'. It is not uncommon that companies have such limited data [8]. To compensate for the limited data, the DST uses knowledge discovery (KD) to enrich the data set. The KD approach is model-based as it starts from theoretical models of human behavior, translating these into mathematics and exploiting them to retrieve new information from the data.

for companies, it is important to have a good understanding of the process behind customer satisfaction. The past two decades, various psychological models tried to capture the customer satisfaction process, among which Oliver's Expectancy-Disconfirmation Paradigm (EDP) [7] became one of the most dominant models. This model identifies three antecedents of customer satisfaction, i.e. product performance, customer expectation and disconfirmation. Customers compare the product performance against their expectation, which causes positive, negative or zero disconfirmation. Next, the expectation sets the initial satisfaction level and positive (negative) disconfirmation will increase (decrease) the satisfaction level. When the EDP is compared with the available data, customer expectation and disconfirmation appear to be missing. The first step of the model-based KD approach retrieves a proxy of the customer's expectation from the available data.

The ED Paradigm can be divided into three pieces. First customers transform the product performance for each product dimension into a disconfirmation level by comparing it against their expectation, secondly a customer aggregates all the disconfirmations at product dimension level into a general disconfirmation level and thirdly the final disconfirmation level changes the initial satisfaction level set by expectation. It can be shown that Dombi's uninorm [5], a well-known aggregation operator in the field of fuzzy set theory and aggregation operators [2], is an appropriate function to model the EDP when only product performance and customer satisfaction are available. Dombi's uninorm is a representable uninorm [2] which can be written as

$$U(x_1, x_2, \ldots, x_n) = f\left(\sum_{i=1}^{n} f^{-1}(x_i)\right) \tag{1}$$

with $f(x) = \frac{\alpha e^x}{1+\alpha e^x}$ and $f^{-1}(x) = ln\left(\frac{x}{\alpha - \alpha x}\right)$. The function $f(x)$ is a monotone function, plotted in Fig. 1a which illustrates how performance is transformed into disconfirmation. The intersection between the X-axis and the function represents the performance level which leads to a zero disconfirmation, i.e. the customer's expectation. The discrepancy between the expectation and the performance is indicated with the letter d and represents the objective disconfirmation which can differ from the customer's subjective disconfirmation level. The plot illustrates that small discrepancies between expectation and performance lead to a near zero disconfirmation level, while large discrepancies are reinforced into much larger subjective disconfirmations. This behavior corresponds well with the assimilation-contrast theory which is part of the foundations of the EDP [1].



(a) $f(x) = \frac{\alpha e^x}{1+\alpha e^x}$

(b) $f^{-1}(x) = ln\left(\frac{x}{\alpha - \alpha x}\right)$

Fig. 1. Generator Functions of Dombi's Uninorm

Next, Dombi's uninorm takes the sum of all $f(x_i)$ which corresponds to the second step of the EDP, i.e. aggregating the disconfirmations from the dimension level. Finally, the outer function $f^{-1}(x)$ transforms the general disconfirmation level into the satisfaction level. This function is a monotone S-shaped function (cf. Fig. 1b). The intersection of the function with the Y-axis represents the satisfaction level for zero disconfirmation, i.e. the initial satisfaction level. It can be shown that this corresponds to the expectation level, which models the relationship between expectation and initial satisfaction.

Using Dombi's uninorm to model customer satisfaction and the EDP has both strengths and limitations. The main limitation of this approach is that it assumes that customers expect the same for all product dimensions. Despite this restrictive assumption, various authors have found that the expectation derived with this model is a good proxy for the true underlying expectation [4,13,14]. The strongest advantage of this modeling approach is that the modeling function contains only one parameter, which allows the researcher to build a separate model for each customer and learn the customer's individual expectation level. If the function $f(x)$ and its inverse $f^{-1}(x)$ are entered into (1), the model's parameter α can be calculated as follows where x_i represents the product's performance for dimension i and S represents the satisfaction score:

$$\alpha = \left[\frac{\prod_{i=1}^{n} P_i}{\prod_{i=1}^{n} (1 - P_i)} \quad \frac{1 - S}{S} \right]^{\frac{1}{n-1}} . \tag{2}$$

Next, the customer's expectation ν is x_i for which $f(x_i) = 0$:

$$\nu = \frac{\alpha}{1 + \alpha} . \tag{3}$$

Once this model is used to extract new knowledge from the data set, the new expectation construct is used to construct the expectation-performance compatibility. The idea for this construct stems from the participatory learning paradigm introduced by Yager [15]. Central in the formulation of the participatory learning system is the compatibility measure. A high compatibility between the learner's beliefs and the new information has a positive effect on the learning process. Analogously, the customer's intentions (of loyalty or word-of-mouth behavior) are also formed in the framework of what has already been learned about the product. Previous product usages result in expectations, which act as the customer's belief and the recently experienced product performance acts as new information. This results in the following two hypotheses:

Hypothesis 1. *The compatibility between expectation and performance is positively correlated to the customer's intended loyalty.*

Hypothesis 2. *The compatibility between expectation and performance is positively correlated to the customer's recommendation intention.*

Note that incompatibility caused by a better product performance than expected will also have a negative effect on the customer's intentions. This counterintuitive effect relates to the uncertainty about the product caused by the incompatibility. To calculate the customer's expectation-performance compatibility, the incompatibility I_i for dimension i is calculated first.

$$I_i = x_i - \nu \tag{4}$$

Note that the domain of I_i is linearly transformed to the $[-1, 1]$ interval such that both -1 and 1 reflect complete incompatibility and 0 reflects perfect compatibility. Next, the various dimension incompatibility scores need to be aggregated into a compatibility score C at the product level such that the following properties hold.

Property 1.

$$\text{a) } 0 \leq C \leq 1$$

$$\text{b) } \forall i : I_i = 0 \Rightarrow C = 1$$

$$\text{c) } \forall i : I_i \in \{-1, 1\} \Rightarrow C = 0$$

Property 1a) sets the boundaries of the compatibility measure. Property 1b) states that if the performance of each product attribute is compatible with the expectation, the product level compatibility measure should be maximized. Property 1c) implies that if each product attribute is completely incompatible with the expected performance, the product level compatibility should be minimized. A first proper aggregation function to define compatibility is the same as the compatibility measure in Yager's participatory learning paradigm [15]. This compatibility measure will be denoted as C^1 and is defined as follows:

$$C^1 = 1 - \frac{1}{n} \sum_{i=1}^{n} |I_i| \tag{5}$$

A second possible aggregation function to define compatibility takes the square of the attribute incompatibilities instead of the absolute value. This compatibility measure is denoted as C^2 and can be expressed as follows:

$$C^2 = 1 - \frac{1}{n} \sum_{i=1}^{n} I_i^2 \tag{6}$$

3 Data

The data sets come from two companies active in the Belgian energy market. Both data sets measure the customer's satisfaction, the customer's intention to switch (loyalty) and to recommend (word-of-mouth), together with performance of various product attributes. During pre-processing, model-based imputation and factor analysis were used to improve data quality. The customer's expectations were calculated by means of (3) and (2), while the compatibility measures were calculated with (5) and (6). Table 1 provides descriptive statistics about the data for company 1 and 2. In total, the data set for company 1 contains 525 observations and 528 observations for company 2.

4 Evaluating the Compatibility Construct

To the knowledge of the authors, the relationships between the customer's intentions and compatibility, as suggested by hypotheses 1 and 2, have not yet been established in marketing literature. Therefore, prior to the development of the DST, these hypotheses need to be validated.

The Pearson correlations between both compatibility measures on the one hand and loyalty and recommendation on the other hand were calculated. Both

Table 1. Descriptive statistics: company 1

	Company 1			Company 2		
	Range	Mean	Std. Dev.	Range	Mean	Std. Dev.
Satisfaction	[1.00-10.0]	7.90	1.45	[1.00-10.0]	7.01	2.10
Intention to Switch	[1.00-5.00]	3.63	1.07	[1.00-5.00]	3.09	1.26
Intention to Recommend	[1.00-5.00]	2.29	1.26	[1.00-5.00]	2.71	1.38
Availability	[1.00-5.00]	3.51	1.02	[1.00-5.00]	3.81	0.95
Employees	[1.00-5.00]	4.14	0.75	[1.00-5.00]	4.22	0.80
Service	[1.00-5.00]	3.80	0.83	[1.00-5.00]	3.55	1.06
Information	[1.00-5.00]	3.39	1.00	[1.00-5.00]	3.06	1.15
Invoices	[1.33-5.00]	4.22	0.75	[1.00-5.00]	3.85	1.08
Expectation	[1.23-5.36]	3.69	0.80	[1.17-5.54]	3.68	0.89
Compatibility C^1	[0.72-1.00]	0.92	0.05	[0.66-1.00]	0.90	0.05
Compatibility C^2	[0.92-1.00]	0.99	0.01	[0.88-1.00]	0.98	0.02

hypotheses are compared against the null hypothesis that no correlation exists between the compatibility measures and the customer's intentions. The results are presented in Table 2.

Table 2. Pearson correlations

	Company 1		Company 2	
	C^1	C^2	C^1	C^2
Intention to Switch	-0.21^a	-0.19^a	-0.39^a	-0.33^a
Intention to Recommend	0.29^a	0.26^a	0.47^a	0.44^a

[a] Statistically significant at 1%.

These results reveal that all correlations are statistically significant, which rejects the null hypotheses and confirms the assumed relationships between performance-expectation compatibility and the customer's intentions. Note that intention to switch is the opposite of loyalty which explains the negative correlation sign. Furthermore, the results in Table 2 suggest that C^1 has a stronger correlation with the customer's intentions than C^2. The statistical significance of these differences were tested with the appropriate test statistic [11]. These tests revealed, after applying a Bonferroni correction [3] for multiple comparisons, that C^1 has a statistically stronger correlation with both customer's intentions than C^2 for company 2. For company 1, only the correlation between C^2 and recommendation is statistically smaller than the correlation between C^1 and recommendation. All these results suggest that C^1 should be preferred over C^2. C^1 might have a stronger correlation with the customer's intentions because in C^2 the incompatibilities are diminished by taking the square, which is also indicated by the smaller standard deviation of C^2. The DST will use C^1 as its compatibility measure.

5 The Decision Support Tool

The goal of the DST is to illustrate the individual impact of three different constructs, i.e. satisfaction, performance and compatibility, on the customer's intentions. The DST should allow managers to quickly evaluate the current situation regarding customer loyalty and word-of-mouth and identify the appropriate course of action to improve these customer intentions. The impact of the three constructs on the customer's intentions are measured by means of a multinomial logistic regression model [6]. The DST will be discussed for the intention to recommend. The DST for intention to switch is completely analogous.

First the intentions need to be discretized into three categories, i.e. *will not recommend* (Recommendation $\in \{1, 2\} \rightarrow Y = 1$), *might recommend* (Recommendation $= 3 \rightarrow Y = 2$) and *will recommend* (Recommendation $\in \{4, 5\} \rightarrow Y = 3$). Next, product performance is constructed as the average of the dimension performances and *Satisfaction*, *Product Performance* and *Compatibility* are linearly transformed to the $[0, 1]$ interval for a fair comparison. Next, the multinomial logistic regression model can be expressed as the set of the following two equations, with Y denoting *Intentions to Recommend*, X_1 denoting *Satisfaction*, X_2 denoting the *Product Performance* and X_3 denoting the *Compatibility* C^1:

$$ln\left(\frac{P(Y=1)}{P(Y=2)}\right) = \beta_{01} + \beta_{11}X_1 + \beta_{21}X_2 + \beta_{31}X_3 \qquad (7)$$

$$ln\left(\frac{P(Y=3)}{P(Y=2)}\right) = \beta_{03} + \beta_{13}X_1 + \beta_{23}X_2 + \beta_{33}X_3 \qquad (8)$$

Once the model parameters are estimated, the probability that a customer will recommend can be calculated for any given X_1, X_2 and X_3.

The DST creates two views of the current situation. The first view is plotted in Fig. 2 and shows how the probability to recommend increases as one of the three constructs goes from 0 to 1, while the other constructs are kept constant at their sample's average. The solid horizontal line represents the probability a customer will recommend when all three constructs are at their average value and is the

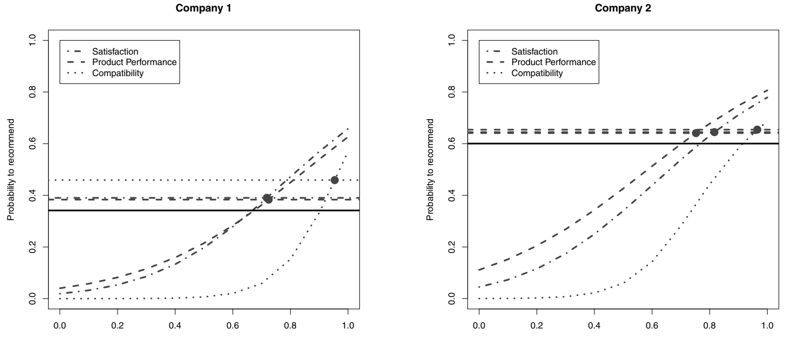


Fig. 2. DST: View 1

first thing the manager should look at, as it gives an immediate impression of the current probability to recommend. While the manager of company 2 might be content with the current probability to recommend of 60%, the manager of company 1 will probably not be satisfied with the current probability to recommend of less than 40%. These results make sense given the fact that company 1 is a newcomer to the market, while company 2 is one of the oldest companies around. While the second company has already built a strong reputation, resulting in customers recommending them, the first company has not.

If the manager of company 1 finds the probability to recommend too low, he can take a look at the marked points for each curve. These points indicate an equally large increase from the sample's average. In this plot, the three points indicate the new probability to recommend if the average score for respectively satisfaction, product performance or compatibility would increase with 0.05. From this plot, one can easily detect the sensitivity of the probability to recommend to any of the three constructs. As for company 1, it becomes clear that the manager should increase effort to improve compatibility, rather than trying to increase satisfaction or product performance. By focussing on compatibility, the effect on the probability to recommend is twice as large in comparison to the effect from satisfaction or product performance.

The second view of the DST, which is shown in Fig. 3 is particularly useful if the manager wants to improve a customer's intention to a specified level. This view shows how well each construct currently performs on average (light grey area) and how much a specific construct must increase to reach a predetermined intention level if the other constructs were kept constant at their average performance (dark grey area). The manager of company 1 can learn from Fig. 3 that the required compatibility improvement to reach a 50% probability to recommend is much smaller than the necessary improvements on satisfaction and product performance. This corresponds to the conclusions from the previous view. However, from the second view it is also easy to detect that compatibility already scores very high on average, indicating that a further increase might become difficult. As for the manager of company 2, who wants to increase the probability

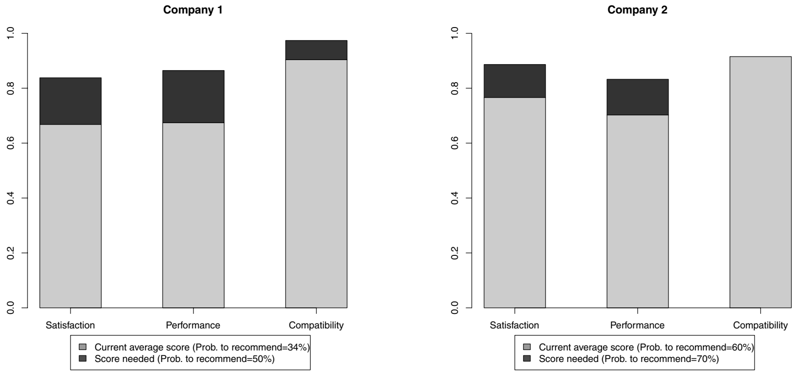


Fig. 3. DST: View 2

to recommend to 70%, it becomes clear that this goal cannot be achieved by only improving compatibility. This is reflected by the fact that there is no dark grey area for compatibility.

Both view 1 and 2 of the DST reveal similar information, but they serve a different purpose. The first view is constructed to allow an immediate evaluation of the current situation regarding the customer's intentions and reveals to which construct this intention is most sensitive. The second view is more useful if a particular intention level wants to be achieved. The second view reveals the absolute improvement necessary and gives a good indication of the relative improvement and the available room for improvement. Also note that none of the two views display exact values on the plots. Without exact values, the manager is not distracted by non-significant differences and attention will be drawn to relative differences which have more practical value. All code to generate the two DST views are written in R [10] and are available at request from the corresponding author.

6 Conclusions

This article offers two main contributions. Firstly, a model-based knowledge discovery approach is described to extract new and useful information from limited customer satisfaction data. This approach translates human behavior theories into a mathematical model in such a way that they can be applied to limited data, resulting in two new constructs, i.e. customer's expectation and expectation-performance compatibility. The expectation construct has been used by other authors, but this article reveals the close link between the mathematics behind Dombi's uninorm and the EDP theory. The compatibility construct is a new construct within the domain of customer's intentions and its validity has been confirmed by various tests in this article.

The second contribution is the development of a decision support tool for evaluating customer loyalty and word-of-mouth. This DST is build upon the knowledge extracted from the limited data in the first part of the article and provides two views. The DST allows a manager to evaluate the current situation regarding the customer's intentions, allows the manager to identify the constructs to which the intentions are most sensitive and gives an indication how much effort will be needed to reach a certain intention level. The two views of the DST are constructed in such a way that they are easy to interpret and offer this information at a glance. Finally, the nature of the DST allows it to be easily integrated into a manager's dashboard, which can be used to monitor the company's evolution.

References

1. Anderson, R.E.: Consumer dissatisfaction: The effect of disconfirmed expectancy on perceived product performance. Journal of Marketing Research 10, 38–44 (1973)
2. Beliakov, G., Pradera, A., Calvo, T.: Aggregation Functions: A Guide for Practitioners, vol. 221. Springer, Heidelberg (2007)

3. Curtin, F., Schulz, P.: Multiple correlations and bonferroni's correction. Biological Psychiatry 44, 775–777 (1998)
4. Depaire, B., Vanhoof, K., Wets, G.: Managerial opportunities of uninorm based importance–performance analysis. WSEAS Transactions on Business and Economics 3(3), 101–108 (2006)
5. Dombi, J.: Basic concepts for the theory of evaluation: The aggregative operator. European Journal of Operational Research 10(3), 282–293 (1982)
6. Menard, S.: Applied Logistic Regression Analysis. In: Quantitative Applications in the Social Sciences. Sage Publications, Inc., Thousand Oaks (2002)
7. Oliver, R.L.: A cognitive model of the antecedents and consequences of satisfaction decisions. Journal of Marketing Research 17(4), 460–469 (1980)
8. Oliver, R.L.: Satisfaction: A Behavioral Perspective on the Consumer. McGraw-Hill, New York (1996)
9. Power, D.: Categorizing decision support systems: A multidimensional approach. In: Mora, M., Forgionne, G., Gupta, J.N. (eds.) Decision Making Support Systems: Achievements, Trends and Challenges for the New Decade, ch. 2, pp. 20–27. Idea Group Publishing, USA (2003)
10. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2007), http://www.R-project.org ISBN 3-900051-07-0
11. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. CRC Press, Boca Raton (1997)
12. Szymanski, D.M., Henard, D.H.: Customer satisfaction: A meta-analysis of the empirical evidence. Journal of the Academy of Marketing Science 29(1), 16–35 (2001)
13. Vanhoof, K., Brijs, T., Wets, G.: An indirect measurement for customer expectation. In: Baets, B.D., Fodor, J. (eds.) Principles of Fuzzy Preference Modelling and Decision Making, pp. 109–122. s.l. Academia Press, New York (2003)
14. Vanhoof, K., Pauwels, P., Dombi, J., Brijs, T., Wets, G.: Penalty–reward analysis with uninorms: A study of customer (dis)satisfaction. In: Ruan, D., Chen, G., Kerre, E.E., Wets, G. (eds.) Intelligent Data Mining. Techniques and Applications, pp. 237–252. Springer, Heidelberg (2005)
15. Yager, R.R.: A model of participatory learning. IEEE Transactions on Systems, Man and Cybernetics 20, 1229–1234 (1990)

An Argumentation-Based BDI Personal Assistant

Federico Schlesinger, Edgardo Ferretti, Marcelo Errecalde, and Guillermo Aguirre

Laboratorio de Investigación y Desarrollo en Inteligencia Computacional
Universidad Nacional de San Luis. San Luis, Argentina
fedest@gmail.com, {ferretti,merreca,gaguirre}@unsl.edu.ar

Abstract. BDI models and argumentation-based approaches are powerful tools that can play a fundamental role in implementing intelligent systems for complex business and industrial problems. Some recent works have started studying the integration of both approaches from a theoretical point of view. However, little effort has been made to analize how these technologies can be effectively integrated in complex systems which require the use of well-known development resources, such as agent development platforms and other programming tools. This work addresses that problem, showing how an argumentation-based reasoning service can be used in different components of a BDI agent implemented with a Jadex platform. All the concepts involved in our proposal are exemplified by designing and implementing a *travel assistant* agent that allows to observe how BDI and argumentation approaches can be effectively integrated in a working system developed with freely available technologies.

1 Introduction

Belief-Desire-Intention (BDI) models and argumentation processes are central concepts in human intelligence, which have gained considerable attention as powerful tools to build artificial intelligent systems. On one hand, BDI is considered as one of the most influential *practical reasoning architectures*, i.e., an architecture to decide what action (or actions) should be carried out. On the other hand, argumentation is a major component of Artificial Intelligence [1] and supports non-monotonic reasoning, providing rationally justifiable conclusions when the problem involves incomplete and potentially inconsistent information.

Both techniques have very solid theoretical foundations [2,3] but their use is not only restricted to the academic area. Recent works have shown the potentials of these technologies to deal with very complex business and industrial problems [4,5]. Likewise, different studies [6,7,8] have recognized the benefits that an integration of BDI and argumentation models could provide, but they tend to focuss on formal aspects of the problem using very specific formal frameworks. Therefore, it is not always straight to understand how these proposals could be implemented in complex practical systems which require the use of more accepted technologies and development tools.

In this work, this problem is addressed by providing a framework and a personal assistant based on this framework, where the integration of both technologies can be easily understood and directly implemented. Other objectives of this work are: (a) the framework should be generic enough to introduce, with little effort, modifications and new

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 701–710, 2010.

implementations of its different processes and components; (b) the framework should provide an adequate support to easily integrate different argumentation and BDI technologies as well as other essential development resources like agent platforms and programming tools and (c) higher priority should be given to freely available technologies.

In [9], the *WADEX* (*W*eb services, *A*rgumentation and (BDI) Ja*dex*) generic framework was proposed to provide support to integrate argumentation-based inference and web service technologies into the design of a BDI system. The present work complements and extends that proposal by focusing on the use of argumentation in two of the three main processes of the *WADEX* framework: the acceptability filter and the selection process. The feasibility of our proposal is shown by designing and implementing a *travel assistant* (TA) agent intended to help a human user that wants to go on a trip on a weekend to an undetermined destination in USA.

The rest of the paper is organized as follows. Section 2 briefly introduces *Defeasible Logic Programming*, the language used for knowledge and preferences representation. In Sect. 3 the *extended* WADEX framework is proposed to integrate argumentation and decision rules in the framework's decision component. Then, an example showing a concrete application of the extended framework is explained in Sect. 4. Finally, in Sect. 5 some general conclusions are drawn and possible future work is discussed.

2 Knowledge and Preferences Representation Language

This section briefly describes Defeasible Logic Programming (DeLP) [10], the concrete formalism used as knowledge and preferences representation language. A DeLP-program, denoted (Π, Δ), consists of two sets: Π, the set of facts and strict rules (certain information) and Δ, the set of defeasible rules (uncertain information). *Facts* are ground literals representing atomic information. *Strict rules* are denoted $L_0 \leftarrow L_1, \ldots, L_n$ and represent firm information, whereas *defeasible rules* are denoted $L_0 \prec L_1, \ldots, L_n$ and represent tentative information. In both cases, the *head* L_0 is a literal and the *body* $\{L_i\}_{i>0}$ is a set of literals. Strong negation ("$\sim$") is allowed in the head of program rules, and hence may be used to represent contradictory knowledge.

In DeLP, to deal with contradictory and dynamic information, *arguments* for conflicting pieces of information are built and then compared to decide which one *prevails*. An *argument* for a literal L, denoted $\langle \mathscr{A}, L \rangle$, is a minimal set of defeasible rules $\mathscr{A} \subseteq \Delta$, such that $\mathscr{A} \cup \Pi$ is non-contradictory and there is a derivation for L from $\mathscr{A} \cup \Pi$. To establish if $\langle \mathscr{A}, L \rangle$ is a non-defeated argument, *argument rebuttals* or *counter-arguments* that could be *defeaters* for $\langle \mathscr{A}, L \rangle$ are considered, *i.e.*, counter-arguments that by some criterion are preferred to $\langle \mathscr{A}, L \rangle$. Since counter-arguments are arguments, defeaters for them may exist, and defeaters for these defeaters, and so on. Thus, a sequence of arguments called *argumentation line* is constructed, where each argument defeats its predecessor in the line (for a detailed explanation of this dialectical process see [10]). The prevailing argument provides a warrant for the information it supports. A literal L is *warranted* from (Π, Δ) if a non-defeated argument $\mathscr{A}$ supporting L exists. Given a query Q there are four possible answers: YES, if Q is warranted; NO, if the complement of Q is warranted; UNDECIDED, if neither Q nor its complement are warranted; and

UNKNOWN, if Q is not in the language of the program. In DeLP, *generalized specificity*[1] is used as default criterion to compare conflicting arguments, but an advantageous feature of DeLP is that the comparison criterion among arguments can be replaced in a modular way. Therefore, in this work, we also use a criterion based on priorities among literals.[2]

As implementation of the DeLP formalism, our work uses a DeLP-Server [11]; a Prolog implementation of a DeLP interpreter that provides a reasoning service for a group of agents. This service runs as an stand-alone program that can interact with the agents using the TCP/IP protocol. A query for a DeLP-Server is a pair $[C,Q]$, where Q is the literal that DeLP will try to warrant and C is the *context* for that query. The context C of a query can be any DeLP-program.

3 The WADEX Framework

The *WADEX* framework [9] conceptually consists of: (a) an *integration archictecture* which describes how the software components implementing these models interact, and (b) a *general control structure* that provides some general guidelines about how the main steps involved in a particular application can be programmed.

The *integration architecture* is depicted in Fig. 1, where the agent platform and the communication channels between the different elements involved are shown. Most of the application logic is encapsulated in a Jadex agent, following its BDI model. This agent runs on top of a deployed JADE platform. There are other agents also running on the same JADE platform, which can be of diverse nature and can communicate through messages with the main Jadex agent, since Jadex provides facilities for sending and receiving messages from plans, whether in a synchronous or asynchronous way. A JADE agent with special characteristics is the Web Services Integration Gateway (WSIG) agent, which allows to extend these advantages of messaging to the use of invoking web services as well.

A web service invocation process starts with a lookup for the service in the *Directory Facilitator* (*DF*), a JADE special agent implementing the *yellow pages* service (*i.e.*, it allows agents in the platform to publish/retrieve descriptions of services that other agents are willing to consume/provide). Once the service has been located, the agent formats a request message in the body of a plan and sends it. Information obtained from the web service invocation can be stored in the agent's belief base. Moreover, our Jadex agent can make queries to a DeLP-Server. The DeLP-Server runs a DeLP program containing part of the facts and decision logic that are supposed to be static. The possibility of including contextual information in the queries allows to have a piece of the DeLP program in the Jadex agent and / or dynamically generate it (or a part of it). In this case, queries are directly made from the executing code of the Jadex agent sending queries (via sockets) to the DeLP-Server, and receiving responses in the same plan without any agent messages involved.[3]

[1] It prefers an argument with either greater information content or with less use of rules.

[2] It prefers an argument built from rules that contain the highest priority literals.

[3] An alternative implementation, *e.g.*, with a JADE agent acting as interface, could have been easily implemented.

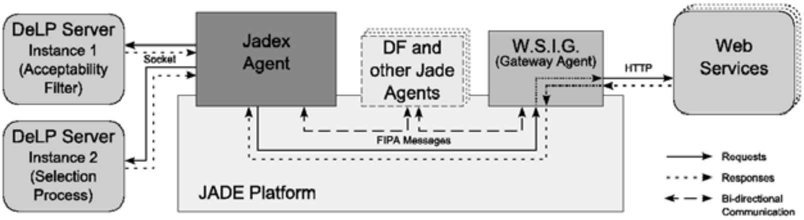


Fig. 1. The integration architecture

As the general application logic is controlled by a (BDI) Jadex agent which acts as initiator of all the interactions requiring distinct web services and inference processes, is important to analyze the steps involved in the *WADEX's general control structure* shown in Fig. 2. It consists of a main cycle with three generic processes: the *acceptability filter* (*acc-filt*), the *selection process* (*select*), and the *execution stage* (*exec*).

The *acceptability filter*, is the process which considers the whole set of available alternatives and discards those that do not satisfy essential requirements to be considered acceptable options. The *selection process* takes as input the options that survive the acceptability filter and compares them in order to decide which alternative will be selected. Three approaches can be used to face this issue: (a) to simply delegate this responsibility to the user, (b) making decision in an automatic way, or (c) using an hybrid approach mixing manual and automatic decisions. This latter approach is the one used in this work. Finally, the *execution stage* involves deciding *how* to accomplish the selected option and executing this plan to achieve the desired goals. At present, this stage involves to select and execute plans from a library of pre-compiled plans, following the standard approach adopted in most PRS-based BDI systems.

```
Let $\mathscr{A}$ be the set of available alternatives
Let $\mathscr{O}$ be the set of acceptable options
Let $o$ be the selected option
While true do
    1. $\mathscr{O} \leftarrow acc\text{-}filt(\mathscr{A})$
    2. $o \leftarrow select(\mathscr{O})$
    3. $execute(o)$
endWhile
```

Fig. 2. Control structure of a WADEX agent

For example, given the application domain sketched in the introductory section, the TA agent's first task is helping to the user to decide his destination. Once the destination has been decided, it will make all the necessary steps to carry out the trip (*i.e.*, buy tickets, make reservations, etc.). In this way, each one of the above-mentioned stages will be represented by a Jadex goal and thus will have at least one associated plan.

The plan for calculating the acceptable options takes each potential destination and sets up a sub-goal for calculating its acceptability. Checking acceptability for a single city involves using a web service to obtain its weather forecast and consulting a DeLP-program, as it will be explained in Sect. 3.1. The candidate cities are stored in the agent's belief base, in the form of Java objects containing the city name, its coordinates and an associated hash table to store the weather-related information of that city. A default set of cities is specified in the agent's definition but more cities can be dynamically added.

When the set of acceptable alternatives has been determined, an argumentation-based decision process compares the acceptable alternatives among each other, as described in Sect. 3.2. Both processes, filtering acceptable cities and comparing among acceptable cities to select one, have to reflect the user's restrictions and preferences. These are captured through an screen with options that is presented to the user when the agent is loaded, and then translated into two DeLP programs that are saved into two different files. After the files generation, the agent starts two DeLP-Server instances, one loaded with each program. These DeLP-programs will be discussed later in Sect. 4.

The necessary steps to travel to the selected city can depend on many factors. Our approach consists of having a set of plans with different courses of action to achieve this goal, and use meta-level reasoning to choose the most adequate. A central aspect of BDI architectures is the possibility of reconsidering intentions. In our example, it is assumed that once the agent has selected a city, the necessary steps for traveling to that city take a considerable amount of time, during which some conditions might change and make the agent discard the selected city and choose a different one. We consider here two possible situations:

- The selected city ceases to be an acceptable option.
- A new option arrives and is considered to be a better option than the current one.

The first case could happen if for example the weather forecast changes. When a change is detected, the agent issues a sub-goal to recheck the acceptability of that city. If the city has turned unacceptable, the goal to perform the necessary steps to travel to the selected city must be dropped, and the main plan has to start over, recheck acceptability for all the cities and select a new one. The second situation might arise for example, if the agent receives information of a sale in flight tickets for a given destination different from the chosen one. In this case, the next step would be to issue a goal to check the acceptability of the new option. If the new option is acceptable, the agent issues the goal to select a city from a set of acceptable cities, but this time only the currently selected option and the new one are passed as parameters to the goal, to select among these two. If the new option is chosen over the current one, the goal for performing steps to travel to the selected city must be dropped. The main plan has to be repeated, but this time there is no need to check acceptability for the other cities and selecting one, as the selected city will now be the newly arrived option. So the main plan now begins from the moment of starting to carry out the trip (*i.e.*, the third abstract step).

This behavior can be easily implemented in Jadex. First, a belief in the belief base, called `selected_city` is available, and stores the city that has been selected. Jadex allows the use of a drop condition in goals (*i.e.*, a condition that tells the system to discard the goal when it holds true). We specify a drop condition for the goal of performing the steps to travel, stating that the goal must be dropped when the city passed as a parameter differs from the one stored in `selected_city`. In case of a situation where the selected option ceases to be acceptable, `selected_city` is set to `null`. In case of a new option that is determined to be better over the current one, the new city is set in the `selected_city` belief. In both cases the goal to travel is dropped.

The three abstract steps in the main plan are inside a loop that is interrupted when the goal to prepare the trip successfully finishes. If that goal fails (*e.g.*, because it is dropped), the loop starts over. The first two steps (calculating acceptable options and selecting one) are only performed when the value of the `selected_city` belief is `null`; otherwise, they are skipped, and the main plan issues a sub-goal to perform the necessary steps to travel to the city stored in the `selected_city` belief.

3.1 Filtering Acceptable Cities

At present, the process which considers the whole set of cities and discards those that do not satisfy essential requirements to be acceptable options, only uses a web service to obtain its weather forecast. The web service invoked by our application corresponds to the National Weather Service of the United States' National Oceanic and Atmospheric Administration (http://www.nws.noaa.gov/forecasts/xml/). This web service is relatively straightforward to use (*e.g.*, invocations are not context-dependant), and provides extensive information in a well structured manner. Among the set of functions available from this service, we only make use of the one called `NDFDgen`, which receives as parameters two real numbers representing latitude and longitude of the geographical point, two dates stating the period of time for which the forecast is required, and a series of weather parameters indicating the required information. We only request the most common weather parameters, such as minimum and maximum temperature, probability of rain, etc.

The agent has a goal that states the need of weather forecast information for a city (given as a parameter to the goal). When the service has been located, the plan sets up the request for the web service, taking the latitude and longitude from the parameter, and sends it to the Gateway Agent. After getting forecast information from the web service, the argumentation process takes place to determine the acceptability of a city. Our application relies on a DeLP program that runs in an instance of a DeLP-Server, and encapsulates the logic to decide whether a city is an acceptable option or not. As will be shown in Sect. 4, the acceptability of a city `x` in our DeLP program, will be given by a warranted argument supporting the conclusion `acc(x)`.

3.2 City Selection

After determining the set of acceptable alternatives, the next step in the agent's control cycle is to select the destination city. At this stage, a device called *decision rule* [12] is used to perform the decision process.

A decision rule is denoted ($D \overset{\mathscr{O}}{\Leftarrow} P$, *not* T), where P contains literals representing preconditions, T contains literals representing constraints, $\mathscr{O}$ represents the set of acceptable options, and D denotes those alternatives that this rule will decide to adopt from $\mathscr{O}$. Decision rules rely on an argumentation formalism (DeLP in this case) to make them applicable. Given a DeLP-program *KB* representing the agent's knowledge base, a rule $(D \overset{\mathcal{O}}{\Leftarrow} P, \mathit{not}\ T)$ is applicable with respect to KB, if every precondition $p_i \in P$ is warranted from KB and every constraint $t_j \in T$ fails to be warranted from KB.

In Sect. 4 it will be exemplified how argumentation is used in the acceptability filter and in the selection process, with two different aims. In the former case, argumentation provides reasons in favor or against an alternative to determine if it is acceptable depending on its own properties, while in the latter a pair of alternatives is compared among each other to select one of them, depending on their properties.

4 Application Domain: Travel Assistant Agent

Let us suppose that John (our tourist) is in New York, and the weekend of November 12^{th} wants to visit another city of USA. New york is the only city he has visited, and he has no particular preference for visiting any other specific city. Since he is leaving USA on November 14^{th} from JFK airport, he has only two days to travel. This fact, maybe a reason for considering only New York's nearby cities. Besides, John has no much extra money to spend in expensive flight tickets; therefore, if there is an offer of a cheap flight for a distant city, this might change his mind about travelling only to nearby cities. John is not a hard-to-please person, but as he has little time for travelling he really wants to visit a place with nice weather to be able of visiting as much as possible from that place. As discussed in Sect. 3, this situation could be modeled by two DeLP-programs. Figure 3(a) shows an excerption of the DeLP-program used to determine the acceptability of the cities, while Fig. 3(b) presents the DeLP-program which states John's preferences.

In Fig. 3(a), rules (1) and (2) are defeasible rules stating that, by default, a city is acceptable if it is a nearby city, and is not recommendable if it is distant.

Rules (3)−(5) exemplify more complex considerations: a distant city can be acceptable if there is a cheap flight to that city, but a city with bad weather is not acceptable, whether it is a nearby or a distant city with a cheap flight to it. The definition of what constitutes a *nearby* or a *distant* city is given by rules (6) and (7) (in this case, the limit was fixed to 500 Km). Next, rules (8)−(12) define the predicate `bad_weather`, exemplifying a way in which the agent could deal with partial and potentially contradictory information. Bad weather is first defined in terms of the forecast (information obtained from the web service) using strict rules. However, forecast information might not be available for some reason, like the date for the trip being very distant from the current date. The last two rules in the extract define bad weather in terms of climate: for example, one might suppose that there is a high probability of very low temperatures in Boston during winter, or that Miami usually has good weather. These last rules are defeasible, because forecast information could contradict them (*e.g.*, the forecast could announce a tropical storm heading to Miami).

Figure 3(b) shows a DeLP-program representing the preferences expressed by John from the interaction with the screen with options (discussed in Sect. 3).[4] As it can be observed, there are six defeasible rules, two for each comparison literal: `nearer_city`,

[4] For further details about how this translation is perfomed and its related properties from a classical decision theory point of view, please refer to [12].

```
[...]
(1)  acc(X) -< nearby_city(X).
(2)  ~acc(X) -< distant_city(X).
(3)  acc(X) -< distant_city(X), travel_date(Y), origin_city(Z), cheap_flight(Z,X,Y).
(4)  ~acc(X) -< nearby_city(X), travel_date(Y), bad_weather(X,Y).
(5)  ~acc(X) -< distant_city(X), travel_date(Y), origin_city(Z),
cheap_flight(Z,X,Y), bad_weather(X,Y).
[...]
(6)  nearby_city(X) <- origin_city(Z), distance(Z,X,K), K < 500.
(7)  distant_city(X) <- origin_city(Z), distance(Z,X,K), K >= 500.
[...]
(8)  bad_weather(X,Y) <- forecast(X,Y,rain).
(9)  bad_weather(X,Y) <- forecast(X,Y,snow).
(10) bad_weather(X,Y) <- forecast(X,Y,min_temp(K)), K < 0.
(11) bad_weather(X,Y) -< bad_climate(X,Y).
(12) ~bad_weather(X,Y) -< good_climate(X,Y).
[...]
```

(a) DeLP-program for determining a city's acceptability

```
(13) better(X,Y) -< nearer_city(X,Y)
(14) ~better(X,Y) -< nearer_city(Y,X)
(15) better(X,Y) -< cheaper_flight(X,Y)
(16) ~better(X,Y) -< cheaper_flight(Y,X)
(17) better(X,Y) -< better_weather(X,Y)
(18) ~better(X,Y) -< better_weather(Y,X)
```

(b) DeLP-program stating John's preferences

Fig. 3. DeLP-programs representing the state of the situation

cheaper_flight and better_weather. These literals represent John's preference criteria, and the defeasible rules that use them, provide reason to support whether a city X must be (or not) considered a better destination than another city Y. As mentioned in Sect. 2, the DeLP-Server loaded with this DeLP-program, uses a literal-based preference criterion where arguments that build their conclusions based on higher priority literals are stronger than those based on lower priority literals. In this case, the priority assigned to these comparison literals is: nearer_city < cheaper_flight < better_weather, stating that John in principle prefers nearer cities to visit, but if a city far from New York has a cheap fligt this allows it to be considered as eligible despite its distance. However, a city that is almost certain to have good weather is John's most preferred criterion since he has only two days for sightseeing.

The DeLP-programs of Figs. 3(a) and 3(b) contain only the decision logic. Any other information must be sent with each query, as *contextual information* generated from static and dynamic knowledge stored in the belief base. From the current situation, a first context is built for the queries sent to the DeLP-Server loaded with the program of Fig. 3(a). After obtaining the set of acceptable cities, a new context based on the previous context and on the set of actual acceptable alternatives is determined, to be used to compare alternatives via queries to the DeLP-Server loaded with the program of Fig. 3(b). The results of these queries serve to make applicable the decision rule which conforms the decision component of the agent, to choose the option to be scheduled.

Given the situation described at the beginning of this section, an example of contextual information might include the facts shown in Fig. 4(a). With this context, the set of acceptable alternatives $\mathscr{O} = \{Miami, Washington, Chicago\}$ will be used as input to the selection process. *Boston* is judged not acceptable, even when it is a nearby city, since bad weather is assumed from the climate information due to the lack of weather forecast

```
origin_city(new_york).
travel_date(f12_12_09).
forecast(miami,f12_12_09, min_temp(20)).
forecast(washington,f12_12_09,min_temp(12)).
forecast(chicago,f12_12_09,min_temp(5)).
forecast(los_angeles,f12_12_09,min_temp(18)).
bad_climate(boston, Y) <- winter(Y).
cheap_flight(new_york, miami, f12_12_09).
cheap_flight(new_york, chicago, f12_12_09).
distance(new_york, boston, 300).
distance(new_york, miami, 1747).
distance(new_york, washington, 328).
distance(new_york, chicago, 1140).
distance(new_york, los_angeles, 3921).
```

(a)

```
cheaper_flight(miami,chicago).
cheaper_flight(chicago,washington).
nearer_city(chicago,miami).
nearer_city(washington,miami).
nearer_city(washington,chicago).
better_weather(miami,chicago).
better_weather(miami,washington).
better_weather(washington,chicago).
```

(b)

Fig. 4. Contextual information for: (a) the acceptability filter and (b) the decision process

for that city. *Miami* and *Chicago* are distant cities but the agent is aware of cheap flight offers from New York to them, which is not the case with *Los Angeles*.

As mentioned above in Sect. 3.2 at this stage the decision rules device is used to decide which will finally be the option to be scheduled. These rules are based on the preference criteria provided to the Jadex agent (Fig. 3(b)) and from contextual information referring only to preferences (see Fig. 4(b)).[5] In fact, the decision rule used in this application is the following one:[6]

$$(19)\ \{W\} \overset{B}{\Leftarrow} \{better(W,Y)\}, not\ \{better(Z,W)\}$$

From the context of Fig. 4(b), using rule (17) two undefeated arguments $\mathscr{A}_1$ and $\mathscr{A}_2$ can be built supporting $better(miami, chicago)$ and $better(miami, washington)$. These arguments will be used to warrant the restriction of rule (19), thus making *chicago* and *washington* not eligible. Moreover, argument $\mathscr{A}_1$ will also be generated to warrant the precondition of rule (19) to support *miami* as eligible.

$$\mathscr{A}_1 = \langle\{better(miami,chicago) \mathrel{-\!\!\prec} better\_weather(miami,chicago)\}, better(miami,chicago)\rangle$$
$$\mathscr{A}_2 = \langle\{better(miami,washington) \mathrel{-\!\!\prec} better\_weather(miami,washington)\}, better(miami,washington)\rangle$$

Although, arguments $\mathscr{A}_3$ and $\mathscr{A}_4$ can be built to support $\sim better(miami, chicago)$ and $\sim better(miami, washington)$, they are defeated by arguments $\mathscr{A}_1$ and $\mathscr{A}_2$, respectively, as they are based on higher preference comparison literals. Hence, *miami* is chosen.

$$\mathscr{A}_3 = \langle\{\sim better(miami,chicago) \mathrel{-\!\!\prec} nearer\_city(chicago,miami)\}, \sim better(miami,chicago)\rangle$$
$$\mathscr{A}_4 = \langle\{\sim better(miami,washington) \mathrel{-\!\!\prec} nearer\_city(washington,miami)\}, \sim better(miami,washington)\rangle$$

5 Conclusions

In this work, it was presented a proposal to effectively integrate BDI and argumentation approaches in a working system to solve real-world problems. All the concepts involved

[5] It is important to note that the contextual information of Fig. 4(b) is derived from the original contextual information of Fig. 4(a) which depicts the whole situtation.

[6] As indicated in [12], there is another rule to account for the case of alternatives with the same properties. However, due to space constraints we present only the one used in our example.

were exemplified by designing a *travel assistant* agent implemented as an instance of the WADEX framework. One of the main results of this work, was to experimentally determine the importance of combining the flexibility and robustness of BDI systems, and the strength of defeasible argumentation inference to manage incomplete and potentially inconsistent information, to solve a real problem. Besides, this was accomplished by using only freely available resources and other wide-spread technologies such as Web Services and FIPA-compliant agent development platforms.

The implemented system showed that the WADEX framework is generic enough to introduce argumentation-based reasoning in different components and processes of the BDI agent in a relatively direct way. Our future works aim to take advantage of this flexibility and extend the present work with explanation capabilities in order to justify to the user the decisions made by the system. Another possible extension is to implement in the selection stage a voting-based mechanism [13] that takes into account the preferences of the agent.

References

1. Rahwan, I., Simari, G. (eds.): Argumentation in Artificial Intelligence. Springer, Heidelberg (2009)
2. Bratman, M.E., Israel, D.J., Pollack, M.E.: Plans and resource-bounded practical reasoning. Computational Intelligence 4(4), 349–355 (1988)
3. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–357 (1995)
4. Benfield, S.S., Hendrickson, J., Galanti, D.: Making a strong business case for multiagent technology. In: Proceedings of AAMAS '06, pp. 10–15. ACM, New York (2006)
5. Evertsz, R., Fletcher, M., Jones, R., Jarvis, J., Brusey, J., Dance, S.: Implementing Industrial Multi-agent Systems Using JACK. In: Dastani, M.M., Dix, J., El Fallah-Seghrouchni, A. (eds.) PROMAS 2003. LNCS (LNAI), vol. 3067, pp. 18–48. Springer, Heidelberg (2004)
6. Boella, G., van der Torre, L.: BDI and BOID Argumentation: Some examples and ideas for formalization. In: Procs. of IJCAI - Computational Models of Natural Argument (2003)
7. Rahwan, I., Amgoud, L.: An Argumentation-based Approach for Practical Reasoning. In: Proc. of AAMAS '06, pp. 347–354. ACM, New York (2006)
8. Rotstein, N.D., García, A.J., Simari, G.R.: Reasoning from desires to intentions: a dialectical framework. In: Proceedings of AAAI'07, pp. 136–141. AAAI Press, Menlo Park (2007)
9. Schlesinger, F., Errecalde, M., Aguirre, G.: An approach to integrate web services and argumentation into a BDI system. Technical report, Universidad Nacional de San Luis (2009)
10. García, A.J., Simari, G.R.: Defeasible Logic Programming: An Argumentative Approach. Theory and Practice of Logic Programming 4(2), 95–138 (2004)
11. García, A.J., Rotstein, N.D., Tucat, M., Simari, G.R.: An argumentative reasoning service for deliberative agents. In: Zhang, Z., Siekmann, J.H. (eds.) KSEM 2007. LNCS (LNAI), vol. 4798, pp. 128–139. Springer, Heidelberg (2007)
12. Ferretti, E., Errecalde, M.L., García, A.J., Simari, G.R.: Decision rules and arguments in defeasible decision making. In: 2nd Intl. Conference on Computational Models of Arguments (COMMA). Frontiers in Artificial Intelligence and Applications, pp. 171–182. IOS Press, Amsterdam (2008)
13. Sen, S., Haynes, T., Arora, N.: Satisfying user preferences while negotiating meetings. Int. J. Hum. Comput. Stud. 47(3), 407–427 (1997)

Challenges of Distributed Model-Based Diagnosis*

Franz Wotawa and Jörg Weber**

Technische Universität Graz, Institute for Software Technology,
Inffeldgasse 16b/2, A-8010 Graz, Austria
{wotawa,jweber}@ist.tugraz.at
http://www.ist.tugraz.at/

Abstract. The importance of distributed systems is increasing. More and more systems are built using multi-agent or service-oriented architectures. The size of the resulting systems also increases, which makes the diagnosis task more difficult because of the underlying complexity. As a consequence distributed diagnosis has become more and more important but usually it is tied to specific modeling concepts or based on particular algorithms whose correctness and completeness is not proven. Therefore, we focus in this paper on a general theory for distributed diagnosis, which provides a framework for checking the correctness and completeness of distributed diagnosis algorithms. Moreover, we present a simple algorithm and show that correctness and completeness can only be guaranteed under certain assumptions. The theory is of importance for industry to ensure the correctness of diagnosis systems in the distributed case.

1 Introduction

Model-based diagnosis [1,2] is a methodology which aims at detecting faults in systems and at locating the failed components. It employs models which capture the correct and maybe also the faulty behavior of the system. By combining the model with observations of the actual system behavior, a troubleshooter can apply generic algorithms in order to derive a set of possible diagnoses. So far, different applications for hardware and also for software diagnosis have been reported.

Most of the model-based diagnosis applications employ a single model and a single diagnosis engine. While this appears to be natural in the case of non-distributed systems, it is interesting that even the diagnosis of distributed systems is very often based on one centralized model and one diagnosis engine. Unfortunately, the centralized reasoning schema has a number of disadvantages, as discussed in [3]. In particular a centralized diagnoser has a (1) high spatial complexity, (2) weak robustness, and (3) poor scalability. The first disadvantage is due to the fact that the models of distributed (sub-)systems have to be combined in order to come up with a centralized model. The weak robustness results from the fact that a fault in a centralized diagnoser makes the whole diagnostic reasoning crash. The poor scalability occurs because any change in a sub-system causes

* The work described in the paper was partially funded by the Austrian Science Fund (FWF) under contract number P20199-N15.

** Authors are listed in reverse alphabetical order.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 711–720, 2010.

changing the global diagnosis model and, moreover, instead of computing diagnoses locally, the computation has to be done at the global level, often taking into account more components than necessary. The latter issue is particularly severe, as the computation of the diagnoses is, in general, exponential in the number of components.

Because of these disadvantages and the spreading use of distributed systems (e.g., web-services [4] or mobile robots [5,6]), the interest in distributed diagnosis is growing.

In Figure 1 we depict the architecture of a spatially distributed diagnosis system. It comprises two or more local diagnosis systems, each of which has a diagnosis engine DE_i, a model SD_i of the underlying (sub-)system, and a set of observations OBS_i. The latter represents grounded facts that are obtained from sensor information after filtering and symbol grounding. It is worth noting that, due to sensing failures and sensor noise, the sensed observations might not reflect the real state of the world. However, it is the only information available for diagnosis. Another part of the discussed distributed diagnosis architecture is the global diagnosis engine. This engine is used for combining the local diagnoses in some way and maybe also for distributing observations among the local diagnosis engines if necessary.

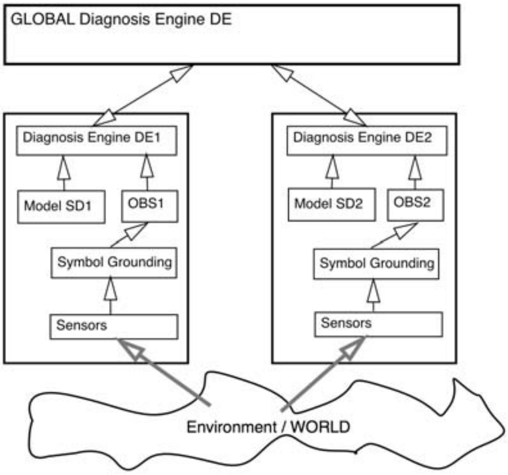


Fig. 1. The distributed diagnosis architecture

Given the distributed diagnosis architecture we are interested in the question whether it is possible to compute local diagnoses and to combine them to a global diagnosis afterwards, without relying on any kind of global model. At a first glance one could think that this is easily possible; however, a closer look reveals that such a distributed diagnosis may not deliver the same results as a diagnosis system which is based on a global information. Consider for example a service-oriented application where two agents are sending the same request and waiting for a response of the same type. I.e., there should be two responses. If only one response arrives but cannot be uniquely assigned to one of the requests, there is no contradiction and therefore no diagnoses at the local level. However, if it can be derived from the global model that there should be two responses but only one was received, then there is a contradiction at the global level, which triggers the computation of diagnoses. Hence, distributed diagnosis might not deliver the same results as a diagnosis system based on global information. We will discuss this issue and give further examples in this paper.

Previous work on distributed diagnosis has mainly be relied on event-based [7], chronicle-based [8], or finite state machine models. In [9] the authors extend their previous work [10] to the distributed case. Other similar work include [11]. In [12] discusses the challenge of designing a distributed diagnosis system. All of those works employ local models, but they also utilize a global diagnoser which acts as a ”mediator” between the local diagnosers in order to determine the global diagnoses. Those global diagnosers typically need some model of how the sub-systems are connected, e.g., the connected neighbours of each sub-system are known.

The previous works rely on specific modelling languages, which has the disadvantage that the general applicability is limited. Another drawback of those works is that the correctness and completeness of the proposed global diagnosis algorithms is neither proven nor defined.

By contrast, we use logical models for our work on principles of distributed diagnosis, which provides more generality than using specific modelling languages. We achieve a broader applicability of our work by directly extending the model-based diagnosis theory of Reiter [1]. Our main contribution is a formal definition of distributed diagnosis and of the correctness and completeness of distributed diagnosis algorithms. The latter is particularly important for practice, because for practical applications we desire certain guarantees wrt to the obtained diagnostic results. Our formalization allows us to discuss limitations of simple distributed diagnosis algorithms which are based on purely local models. Moreover, we provide such an algorithm, and we show that the correctness and completeness of this algorithm is only ensured if certain conditions hold.

Due to the generality of our work we hope that our principles will also be adapted to more application-specific modelling languages. We expect that a deeper discussion of the correctness and completeness of distributed diagnosis algorithms can improve their acceptance by practitioners.

The paper is organized as follows. First, we introduce the basic definitions of model-based diagnosis and extend them to the distributed case. Afterwards, we outline a simple algorithm that allows for computing global diagnoses from the local diagnoses. This simple algorithm relies on purely local models. Then we introduce definitions for the correctness and completeness of distributed diagnosis algorithms, and we show that such algorithms are only correct and complete under certain assumptions. Finally, we conclude the paper.

2 Basic Definitions

Before extending the theory of model-based diagnosis to the distributed case we briefly introduce and explain the basic definitions of Reiter [1]. For diagnosis algorithms we refer the interested reader to Greiner et al. [13] or de Kleer and Williams [2]. According to Reiter [1] a diagnosis problem is characterized by a system description SD, a set of components $COMP$, and a set of observations OBS where the elements of SD and OBS are first order logical formulae. The set of components comprises the unique identifiers of the components where the behavior of the components and the structure of the system is given in SD. In the model-based diagnosis theory only elements of $COMP$ are considered for computing diagnoses and can therefore be either correct (not abnormal) or faulty (abnormal).

Example 1. We illustrate modeling for model-based diagnosis using an example from mobile robotics. We do not model the whole behavior of a robot but only the part which handles the activity "passing a door". The robot can only pass a door if this door is not blocked by an obstacle, which can be detected by a sensor. We identify three components which may exhibit an abnormal behavior. Note that the component names have the subscipt 1, indicating that they refer to the robot R_1 (in a later example, there will be a second robot R_2). The component env_1 refers to the environent (from the point of view of R_1). It is considered as abnormal in the case of unexpected occurrences. In our example, we only consider the unexpected condition that the door is blocked. The component $sensing_1$ represents the sensors of the robot as well as the sensor processing software. If $sensing_1$ is faulty, then the perceptions of the robot can be incorrect. Finally, the abstract component $mobility_1$ refers to the capability of the robot to move. E.g., if the robot's drive is broken, then $mobility_1$ is abnormal.

Hence, $COMP = \{env_1, sensing_1, mobility_1\}$. In the system model SD, we model the correct behavior only. Hereby we use the predicate $real$, stating what really happens in the environment, the predicate $sense$, indicating what the robot senses, and $exec$, referring to the action execution of the robot:

$$SD = \left\{ \begin{array}{l} \neg Ab(env_1) \rightarrow \neg real(door\_blocked), \\ \neg Ab(sensing_1) \rightarrow \big(sense(door\_blocked) \leftrightarrow real(door\_blocked)\big), \\ \neg Ab(sensing_1) \rightarrow \big(sense(passing\_door) \leftrightarrow real(passing\_door)\big), \\ \neg Ab(mobility_1) \rightarrow exec(move\_to\_door), \\ exec(move\_to\_door) \wedge \neg real(door\_blocked) \rightarrow real(passing\_door), \end{array} \right\}$$

We use this example together with the observations that the robot should pass the door but fails, i.e., $OBS = \{exec(move\_to\_door), \neg sense(passing\_door)\}$. □

A diagnosis is defined as set of components that have to be assumed faulty in order to explain the given observations. Explanation in model-based diagnosis (MBD) is defined using satisfiability of a logical formula comprising SD, OBS, and the components' correctness assumptions.

Definition 1 (Diagnosis). *Given a diagnosis problem $(SD, COMP, OBS)$. A set $\Delta \subseteq COMP$ is a diagnosis iff the system description together with the observations and the assumption that all components except the one in Δ are correct is consistent, i.e., the formula $SD \cup OBS \cup \{\neg Ab(C) | C \in COMP \setminus \Delta\} \cup \{Ab(C) | C \in \Delta\}$ is satisfiable.*

Note that in the original paper from Reiter diagnoses have to be minimal. In our case we define a diagnosis Δ to be minimal if there is not subset $\Delta' \subset \Delta$ which is itself a diagnosis. In practice usually only minimal and in most cases only single diagnoses, i.e., diagnoses comprising only one component, are of particular interest. Therefore, from here on we assume that all computed diagnosis are minimal ones unless otherwise stated.

Example 2. (Ex. 1 cont.) The two diagnoses for Example 1 are $\{env_1\}$ and $\{sensing_1\}$. There are no more other minimal diagnoses. $\{mobility_1\}$ is not a diagnosis because assuming that the action is not executed in the right way does not lead to a satisfiable formula. □

In the MBD theory it is more convenient not to compute diagnosis directly but first to compute conflicts from which diagnosis can be derived using the notation of hitting sets. The algorithm from Greiner et al. [13] makes use of conflicts and hitting sets. Formally, a conflict is defined as follows:

Definition 2 (Conflict). *Given a diagnosis problem* $(SD, COMP, OBS)$*. A set* $CO \subseteq COMP$ *is a conflict iff* $SD \cup OBS \cup \{\neg Ab(C) | C \in CO\}$ *is inconsistent.*

Again a conflict is said to be minimal if no subset is itself a conflict. A hitting-set is defined over a set of sets.

Definition 3 (Hitting Set). *Given a set of sets* F*. A set* $H \subseteq \bigcup_{x \in F} x$ *is a hitting set iff the intersection of* H *with all elements of* F *is not empty, i.e.,* $\forall_{x \in F} x \cap H \neq \emptyset$*. A hitting set is said to be minimal if no proper subset is itself a hitting set.*

Example 3. Let F be the set $\{\{a, b\}, \{a, c, d\}\}$. The sets $\{a\}$, $\{b, c\}$, and $\{b, d\}$ are minimal hitting sets. $\{a, b\}$ is also a hitting set but not a minimal one. The set $\{b\}$ is not a hitting set, because the intersection of $\{b\}$ with $\{a, c, d\} \in F$ is empty.

The relationship between diagnosis and conflicts can accordingly to Reiter be expressed using hitting sets. In particular Reiter proved that every minimal diagnosis is a minimal hitting set of the sets of conflicts. Hence, diagnoses can be easily computed from conflicts via first computing conflicts and then computing the hitting sets from the conflicts.

Example 4. (Ex. 1 cont.) There is one conflict for Example 1. Assuming that both env_1 and $sensing_1$ are correct allows us to derive an inconsistency. Hence, $\{env_1, sensing_1\}$ is a conflict from which the two hitting sets $\{env_1\}$ and $\{sensing_1\}$ can be extracted. These hitting sets are exactly the two minimal diagnoses. □

The idea behind *Distributed Model-based Diagnosis (DMBD)* is to solve the diagnosis problem by deconstructing the overall model into separate local system descriptions, and by combining the local diagnoses afterwards to form the global set of diagnoses. Because every sub-model comprises only a fraction of the overall set of components, the time required for computing the minimal diagnoses can be improved, assuming that the communication time as well as the combination of local diagnoses can be neglected. Remember that the computational complexity of computing the minimal diagnoses for a sub-system is exponential in the number of components in this sub-system.

Formally we define the DMBD problem as follows:

Definition 4 (Distributed Diagnosis Problem (DDP)). *A distributed diagnosis problem is a set of tuples* $(SD_i, COMP_i, OBS_i)$ *with* $i = 1, \ldots, n$ *where* SD_i *is the local system description of subsystem i,* $COMP_i$ *is the set of components of the local system, and* OBS_i *are the observations of subsystem i. The following property must be fulfilled for a DDP:* $\forall\, i \neq j : COMP_i \cap COMP_j = \emptyset$.

The property that the components used in each local system are pairwise distinct is important because it ensures that the overall global diagnosis problem can be represented as distinct local diagnosis problems. Note that the properties $SD_i \cap SD_j \neq \emptyset$ and $OBS_i \cap OBS_j \neq \emptyset$ for $i, j \in \{1, \ldots, n\}$ are neither necessary nor desirable. For example, a system might comprise several different instances of the same component, hence the same logical formulae are used to model at least the components' behavior. This also holds for observations where the same observation might be used in different local subsystems.

We now define a diagnosis for a DDP considering the corresponding global diagnosis problem $(SD_G, COMP_G, OBS_G)$ with $SD_G = \bigcup_{i=1}^{n} SD_i$, $COMP_G = \bigcup_{i=1}^{n} COMP_i$, and $OBS_G = \bigcup_{i=1}^{n} OBS_i$.

Definition 5 (Diagnosis DDP). *Given a distributed diagnosis problem (DDP), and let* $(SD_G, COMP_G, OBS_G)$ *be the corresponding global diagnosis system. A set* $\Delta \subseteq COMP_G$ *is a diagnosis for the DDP iff* $SD_G \cup OBS_G \cup \{\neg Ab(C) | C \in COMP_G \setminus \Delta\} \cup \{Ab(C) | C \in \Delta\}$ *is satisfiable.*

It is worth noting that this definition of a solution for a given DDP is as general as possible and does not consider how to compute a diagnosis for the global system. Hence, the definition is not a constructive one. We will tackle the problem of constructively constructing global diagnoses from local ones in the next chapter of this paper. Moreover, it is also worth stating that the above definition is not only of interest for systems that are decomposed into subsystems for the purpose of improving the time required to compute the diagnoses. The same definition can be applied to systems where local diagnosis are relevant without the necessity of computing a global diagnosis. For example, in the robotics domain where multiple mobile autonomous system interact in order to fulfill a given goal, or in the case of web services.

Note that $SD_G \cup OBS_G$ itself has to be consistent. Otherwise, no global diagnoses can be computed because no correctness assumption would lead to a satisfiable logical formula. As a consequence we assume $SD_G \cup OBS_G$ to be consistent.

3 An Algorithm for Distributed Diagnosis

In order to take full advantage of the distribution of a system it would be desirable to diagnose the local sub-systems independently (and concurrently). The results of the local diagnosers have to be combined in order to come up with the global diagnoses. This basic structure of an distributed algorithm has been published and used previously, e.g., in Cordier et al. [11]. However, previous approaches require the model to be formalized as discrete event systems or similar approaches. In this paper, we choose a more general approach by using first order logic for formalizing the models and extend

Reiter's theory of diagnosis to the distributed case. As a consequence the algorithm and in particular the combination of local diagnoses have to be adapted.

Example 5. We illustrate how local diagnoses can be combined. Assume 2 sets of local diagnosis $\Delta_1^S = \{\{a\},\{b,c\},\{b,d\}\}$ and $\Delta_2^S = \{\{e\},\{f,g\}\}$ delivered from 2 diagnosis engines DE_1 and DE_2 respectively. Since, all of the diagnoses are computed from the local models we have to combine them without losing any information. The global diagnosis set would be $\Delta_G^S = \{\{a,e\},\{a,f,g\},\{b,c,e\},\{b,c,f,g\},\{b,d,e\}, \{b,d,f,g\}\}$. □

Formally, the combination of diagnosis sets can be expressed as follows:

$$\Delta_1^S \oplus \Delta_2^S = \{x \cup y | x \in \Delta_1^S \wedge y \in \Delta_2^S\}$$

In case of more than two sets the defined combination of diagnosis sets can be applied recursively. Note that there is no need of checking whether diagnosis sets can be eliminated because of the minimality criterion because in our definition of the DDP we ensured that the components of each local system has to be pairwise disjunctive. Otherwise, the combination of diagnosis sets is more complex.

The following algorithm for distributed diagnosis uses the $\oplus$ operator for combining local diagnosis. The local diagnosis are computed using a function **Diagnosis**. Such a function can easily be provided using the diagnosis algorithm published in Greiner et al. [13]. The algorithm iterates over the local systems and computes diagnosis given the local model and the local observations.

Algorithm **DistrDiagnosis**
Inputs: A DDP $\{(SD_1, COMP_1, OBS_1), \ldots, (SD_n, COMP_n, OBS_n)\}$
Outputs: A set of global diagnoses Δ_G^S

```
1. Let $\Delta_G^S$ be $\{\{\}\}$.
2. For $i = 1$ to $n$ do:
   (a) $\Delta_i^S = \mathbf{Diagnosis}(SD_i, COMP_i, OBS_i)$
   (b) $\Delta_G^S = \Delta_G^S \oplus \Delta_i^S$
3. Return $\Delta_G^S$.
```

The overall complexity of the algorithm is $O(n \cdot 2^{max\{|COMP_1|,\ldots,|COMP_n|\}})$, because each sub-system i requires $O(2^{COMP_i})$ computation time. The complexity of the corresponding global diagnosis is $O(2^{|COMP_G|})$. Hence, distributed diagnosis improves the overall computation time required. However, the complexity result raises a question. How can it be possible to save time? We will give an answer to this question and show that the complexity improvement is because of lack of correctness and completeness. Before giving and proving a theorem stating the incorrectness and incompleteness of the simple **DistrDiagnosis** algorithm, we first formally define correctness and completeness in our case.

Definition 6 (Correctness). *An algorithm* **DD** *for a DDP* $ddp = \{(SD_1, COMP_1, OBS_1), \ldots, (SD_n, COMP_n, OBS_n)\}$ *is correct iff all diagnoses obtained from the algorithm are also diagnoses of the corresponding global diagnosis system, i.e.:*

$$correct(DD) \leftrightarrow \\ (\forall \Delta \in DD(ddp) \rightarrow \Delta \in \mathbf{Diagnosis}(SD_G, COMP_G, OBS_G)).$$

Definition 7 (Completeness). *An algorithm* **DD** *for a DDP* $ddp = \{(SD_1, COMP_1, OBS_1), \ldots, (SD_n, COMP_n, OBS_n)\}$ *is complete iff all diagnoses obtained from the corresponding global diagnosis system are computed using* **DD***, i.e.:*

$$correct(DD) \leftrightarrow \\ (\forall \Delta \in \mathbf{Diagnosis}(SD_G, COMP_G, OBS_G) \rightarrow \Delta \in DD(ddp)).$$

We now show that our simple algorithm for distributed diagnosis is neither correct nor complete.

Theorem 1. *The algorithm* $\mathbf{DistrDiagnosis}$ *is neither correct nor complete.*

Proof. We show that the proposed algorithm for distributed diagnosis is not correct and complete by giving counterexamples. For this purpose we extend Example 1. We assume a second robot R_2, which is supposed to move to the door. In this case the other robot represent an obstacle for the first one. We model this behavior as follows:

$$SD_2 = \left\{ \neg Ab(mobility_2) \rightarrow real(door\_blocked) \right\}$$

In the extension the component set for the second robot only comprises one component, i.e., $COMP_2 = \{mobility_2\}$. The system description of the first robot and its set of components is SD_1 and OBS_1 respectively. Moreover, we assume that $OBS = OBS_1 = OBS_2$, i.e., all robots share the same observations.

Because of SD_2 and OBS the empty set $\{\}$ is the only minimal local diagnosis for the second robot.

Given the diagnosis of the first robot, i.e., $\{env_1\}$ and $\{sensing_1\}$ (see Example 2), we are able to compute the following global diagnoses using algorithm $\mathbf{DistrDiagnosis}$: $\{env_1\}$ and $\{sensing_1\}$. Unfortunately $\{env_1\}$ is not a diagnosis of the corresponding global diagnosis problem but $\{sensing_1, mobility_2\}$ is. Hence, the computed diagnoses are not necessarily correct and since $\mathbf{DistrDiagnosis}$ does not compute the global diagnosis $\{sensing, mobility_2\}$ the algorithm is also not complete. □

What we have learned so far is that a simple algorithm for distributed diagnosis is neither complete nor correct. However, in certain situations $\mathbf{DistrDiagnosis}$ is both correct and complete as stated in the next theorem.

Theorem 2. *Given a DDP* $\{(SD_1, COMP_1, OBS_1), \ldots, (SD_n, COMP_n, OBS_n)\}$. *If for all* $i, j \in \{1, \ldots, n\}$ *the grounded literals of* SD_i *and* SD_j *have an empty intersection, then* $\mathbf{DistrDiagnosis}$ *is correct and complete.*

Proof. In case of an empty intersection of the ground literals in the local system description, the corresponding global diagnosis problem is simple a combination of all local system descriptions and observations, and there is no logical inference between the models of the local system descriptions. Therefore, any diagnosis of the global diagnosis problem is a union of the local diagnoses and vice versa. Hence, to obtain all global diagnoses, we have to combine the local diagnoses using the $\oplus$ operator.

Note that the requirement on the empty intersection of grounded literals, although formulated using the introduced distributed diagnosis theory, has to be interpreted in a stronger way. It is not only required that the observations have no intersection, it is also required that the corresponding physical quantities at the level of the environment like the amount of light at a certain position or the amount of current flowing through a wire for each local diagnosis system are distinct. Hence, no local system is allowed to measure the same quantities using their sensors.

4 Conclusion

The spreading use of distributed systems has led to an increasing demand for distributed diagnosis systems. This is due to the fact that a centralized diagnosis approach has severe disadvantages. The most important one is the computational complexity, which may be significantly improved by employing distributed diagnosis principles.

We provided a definition of distributed diagnosis, and we also formalized the correctness and completeness of distributed diagnosis algorithms. Our work is based directly upon Reiter's general theory of diagnosis, and thus is more general and broader applicable than past works. We also provided a simple algorithm for computing global diagnosis by combining the local diagnoses, and we showed that correctness and completeness of this algorithm is only guaranteed if there are no interactions between the local systems. We expect that our work will have an impact on future applications of distributed diagnosis, where algorithms with proven correctness and completeness are required. Those applications may use specific modelling languages, but still benefit from the principles we proposed in this paper.

The main direction for future research is to find a general distributed diagnosis algorithm which is based on our extension of Reiter's theory of diagnosis and which is guaranteed to be correct and complete without any restriction. Such an algorithm should minimize the overall computational complexity and hence make optimal use of the distributed nature of the system. Moreover, we plan to use the general distributed diagnosis algorithm in our robots to diagnose their subsystems, i.e., the control software, the sensing subsystem, and the actuators.

References

1. Reiter, R.: A theory of diagnosis from first principles. Artificial Intelligence 32(1), 57–95 (1987)
2. de Kleer, J., Williams, B.C.: Diagnosing multiple faults. Artificial Intelligence 32(1), 97–130 (1987)
3. Su, R., Wonham, M.W., Kurien, J., Koutsoukos, X.: Distributed diagnosis for qualitative systems. In: Proc. of the 6th International Workshop on Discrete Event Systems (WODES). IEEE, Los Alamitos (2002)
4. Ardissono, L., Console, L., Goy, A., Petrone, G., Picardi, C., Segnan, M., Dupré, D.: Cooperative model-based diagnosis of web services. In: Proc. 16th Intl. Workshop on Principles of Diagnosis (DX), pp. 125–130 (2005)
5. Daigle, M., Koutsoukos, X., Biswas, G.: Distributed diagnosis of coupled mobile robots. In: Proc. of the IEEE International Conference on Robotics and Automation, Orlando, Florida (2006)

6. Kalech, M., Kaminka, G.: On the design of coordination diagnosis algorithms for teams of situated agents. Artificial Intelligence 171(8-9), 491–513 (2007)
7. Lamperti, G., Zanella, M.: Diagnosis of Active Systems. Kluwer Academic Publishers, Dordrecht (2003)
8. Guillou, X.L., Cordier, M., Robin, S., Rozé, L.: Chronicles for on-line diagnosis of distributed systems. In: Proc. of the European Conference on Artificial Intelligence (ECAI), pp. 194–198. IOS Press, Amsterdam (2008)
9. Baroni, P., Lamperti, G., Pogliano, P., Zanella, M.: Diagnosis of a class of distributed discrete-event systems. IEEE Trans. Systems Man Cybernet. Part A: Systems and Humans 30(6), 731–752 (2000)
10. Baroni, P., Lamperti, G., Pogliano, P., Zanella, M.: Diagnosis of large active systems. Artificial Intelligence 110, 135–183 (1999)
11. Cordier, M., Grastien, A.: Exploiting independence in a decentralised and incremental approach of diagnosis. In: Proc. 17th Intl. Workshop on Principles of Diagnosis, DX (2006)
12. Ribot, P., Pencolé, Y., Combacau, M.: Design requirements for the diagnosability of distributed discrete event systems. In: Proc. 19th Intl. Workshop on Principles of Diagnosis (DX), Blue Mountains, Australia (2008)
13. Greiner, R., Smith, B.A., Wilkerson, R.W.: A correction to the algorithm in Reiter's theory of diagnosis. Artificial Intelligence 41(1), 79–88 (1989)

A Distributed Algorithm for the Multi-Robot Task Allocation Problem

Stefano Giordani[1], Marin Lujak[1], and Francesco Martinelli[2]

[1] Dip. Ingegneria dell'Impresa - University of Rome "Tor Vergata", Italy
[2] Dip. Informatica Sistemi e Produzione - University of Rome "Tor Vergata", Italy

Abstract. In this work we address the Multi-Robot Task Allocation Problem (MRTA). We assume that the decision making environment is decentralized with as many decision makers (agents) as the robots in the system. To solve this problem, we developed a distributed version of the Hungarian Method for the assignment problem. The robots autonomously perform different substeps of the Hungarian algorithm on the base of the individual and the information received through the messages from the other robots in the system. It is assumed that each robot agent has an information regarding its distance from the targets in the environment. The inter-robot communication is performed over a connected dynamic communication network and the solution to the assignment problem is reached without any common coordinator or a shared memory of the system. The algorithm comes up with a global optimum solution in $O(n^3)$ cumulative time ($O(n^2)$ for each robot), with $O(n^3)$ number of messages exchanged among the n robots.

Keywords: Multi robot task allocation, assignment problem, distributed algorithm.

1 Introduction

In this work, we consider the Multi-Robot Task Allocation problem (MRTA) which objective is to find the assignment of n robots to a set of n tasks based on the optimization of some global objective function (see, e.g., Gerkey and Mataric, 2003). MRTA corresponds to the (linear sum) assignment problem for which the first developed algorithm was the Hungarian method (see Kuhn, 1955).

The Hungarian method or Kuhn-Munkres algorithm is a well known iterative algorithm which maintains dual feasibility during calculation and searches for a primal solution satisfying complementary slackness conditions. If the primal solution is feasible, the solution is optimal. If the primal solution is not feasible, the method performs a modification of the dual feasible solution after which a new iteration starts. Hungarian method can be implemented using the alternating trees so that its worst case time complexity is limited by $O(n^3)$ (see, e.g., Papadimitriou and Steiglitz 1982).

We assume that the decision making environment is decentralized with as many decision makers (agents) as the robots in the system. To solve this problem, we developed a distributed version of the Hungarian Method. The robots

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 721–730, 2010.

autonomously perform different sub-steps of the Hungarian algorithm on the base of the individual and the information received through the messages from the other robots in the system. It is assumed that each robot agent has the information regarding its distance from the targets in the environment or, more generally, the assignment costs to the same. The inter-robot communication is performed over a connected dynamic communication network. The algorithm comes up with a global optimum solution in $O(n^3)$ cumulative time ($O(n^2)$ for each robot), with $O(n^3)$ number of messages exchanged among the n robots. There is no shared memory among the processors and the solution to the assignment problem is reached without any common coordinator of the system. The tasks are memoryless and are indeed considered as objects without any calculation or memory capacities.

The outline of the paper is as follows. In Section 2, we introduce related work. In Section 3, some basic definitions are stated together with the formulation of the problem. In Section 4, the distributed allocation algorithm is presented. In Section 5 we present the simulation results. We close the paper with the possible directions of the future work and the conclusion in Section 6.

2 Related Work

There are several main approaches to the Assignment problem (see, e.g., Burkard and Çela, 1999).

The classical centralized assignment methods find a matching solution through the iterative improvement of some cost function: in primal simplex methods it is a primal cost, and in Hungarian, dual simplex and relaxation methods it is a dual cost (see Bertsekas, 1992).

The Auction algorithms can improve as well as worsen both the primal and the dual cost through the intermediate iterations, although at the end the optimal assignment is found (see Bertsekas, 1991). Bertsekas in this work introduces the auction algorithm in which the agents bid for the tasks in iterative manner, and in each iteration, the bidding increment is always at least equal to ϵ (ϵ-complementary slackness). If $\epsilon < \frac{1}{n}$ the algorithm finds the optimal solution, and it runs in $O(n^3 \cdot \max\{c_{ij}\})$ time. Zavlanos et al. (2008) provide a distributed version of the auction algorithm proposed by Bertsekas for the networked systems with the lack of global information due to the limited communication capabilities of the agents. Updated prices, necessary for accurate bidding can be obtained in a multi-hop fashion only by local exchange of information between adjacent agents. No shared memory is available and the agents are required to store locally all the pricing information. This approach calculates the optimal solution in $O(\Delta \cdot n^3 \cdot \max\{c_{ij}\})$ time, with $\Delta \leq n-1$ being the maximum network diameter of the communication network.

There are also many parallel algorithms based on the Hungarian method. For a good survey see, e.g., Burkard and Çela, 1999, Bertsekas et al., 1995.

Another way of seeing the assignment problem is through the Game theoretical formulation (see, e.g., Arslan et al., 2007).

In the robotic community, there are many heuristic methods developed for the MRTA problem. Smith and Bullo (2007) describe two algorithms, namely ETSP, and Grid algorithm, for the task assignment in the sparse and dense environments. The agents have a full knowledge of the tasks in the environment and, in the ETSP algorithm approach the closest ones. If the closest task is occupied, they pre-compute an optimal tour through the n remaining tasks, and search for the first non-occupied task based on certain criteria. Gerkey and Mataric (2003, 2004) describe certain relevant heuristic algorithms for the MRTA problem through the prism of three important factors: computation and communication requirements, and the manner in which tasks are considered for (re)assignment, i.e. whether all or just a part of the tasks are considered for (re)assignment in each iteration. The problem with these approaches, as the authors state, is that there is no characterization of the solution quality that can be expected of the algorithms. Kwok et al. (2002) apply the classic combinatorial methods, such as Hungarian and Gabow algorithms, to the assignment problem in the context of a set of mobile robots which need to be moved to a desired matrix of grid points to have a complete surveillance of the desired area.

Top-down centralized methods and, in general, centralized control, are highly complex and costly in the cases with increased number of controlled components, in our case robots, and are influenced by resource limitations, performance bottlenecks, and critical failures (see, e.g., Sugar and Kumar, 2000). On the contrary, the coordination system in a bottom-up multi-agent modular approach distributes computational resources and capabilities across a network of interconnected agents, and therefore reduces the complexity and increases the robustness of the same. MAS does not suffer from the "single point of failure" problem associated with centralized systems (see e.g. Wooldridge, 2002). Moreover, in the case of an unknown dynamic environment or the unknown moving obstacles situation as is the case in the factories with mobile production robots and stochastic demand situation (see, e.g., Giordani, Lujak, and Martinelli, 2009), local planning is preferred over global planning. The latter is expensive if there are many agents in the system or if there exists varying demand with multiple combinations of production machines capable of satisfying it. Some of the further advantages of the decentralized implementation of this algorithm in respect to the centralized one are: (see, e.g., Lueth and Laengle 1994) modularity: the robots are by nature independent; decentralized knowledge bases: each robot uses a local knowledge base that stores relevant information for the robot and is capable of exchanging this information with other robots; fault-tolerance and redundancy: in case of an error on a singular robot, the multi-robot system continues to function, and extendibility: new robots can be added to the original system without any change in the system architecture.

3 Definitions and Problem Formulation

We consider a system consisting of a set R of n collaborative mobile robot agents positioned in an environment with a set T of n target locations (tasks) and a given $n \times n$ matrix of the distances (or costs) c_{ij} between the robots that are represented by the index j, and the tasks represented by i. Each robot can be assigned to maximally one task, and each task can be performed by not more than one robot. The problem consists of finding a feasible assignment of minimum total cost (distance) between the robots and the tasks, i.e., determining the minimum weight perfect matching of the weighted bipartite graph $G = (T \bigcup R, E)$ with weight c_{ij} on edge $(i, j) \in E$. The mathematical formulation of the problem is:

$$\min \sum_{i,j} c_{ij} x_{ij} \tag{1}$$

subject to

$$\sum_{i=1}^{n} x_{ij} = 1, \quad \forall\, j \ , \tag{2}$$

and

$$\sum_{j=1}^{n} x_{ij} = 1, \quad \forall\, i \ , \tag{3}$$

$$x_{ij} \geq 0 \ . \tag{4}$$

Although fractional solutions exist, there always exists an optimal integer solution which corresponds to a real matching (see, e.g., Papadimitriou and Steiglitz, 1982). In particular, $x_{ij} = 1$ if the robot j is matched to the task i; otherwise $x_{ij} = 0$.

Denoting the above problem formulation as the primal problem, it is possible to define the dual problem as follows. Let α_i be a dual variable related to task i, and β_j the same related to robot j. Both variables are real and unrestricted in sign. The objective of the dual problem is

$$\max\{\sum_j \beta_j - \sum_i \alpha_i\} \tag{5}$$

subject to

$$\beta_j - \alpha_i \leq c_{ij} \quad , \forall\, i \ , \ \forall\, j \ . \tag{6}$$

It is possible to give an economic interpretation to the dual problem and the dual variables: α_i is the price that each robot j will pay if it gets matched with the task i, while β_j is the robot j's utility for the matching with a certain task. The constraints (6) of the dual problem state that the utility β_j of robot j cannot be greater than the total cost $(c_{ij} + \alpha_i)$ of allocation (matching) to task i. The dual objective function (5) to be maximized is the difference between the sum of the utilities of the robots and the sum of the prices of the tasks, i.e., the total net profit of the robots. On the base of the duality in linear programming, $\sum_j \beta_j - \sum_i \alpha_i \leq \sum_{ij} c_{ij} \cdot x_{ij}$; therefore, the total net profit of the robots cannot be greater than the total assignment cost that the robots have to pay for moving to the tasks, and only at the optimum, those two are equal. Obviously, at the optimum, each robot j will be matched to task i for which the utility of the robot is $\beta_j = c_{ij} + \alpha_i$, i.e., exactly equal to the total cost that the robot j would have for matching with task i.

4 Distributed Allocation Algorithm

It is assumed that each robot agent has an information regarding its position and can receive the information regarding the position of all the tasks in the environment through the coordinates on a map of the environment. The inter-robot communication is performed over a connected dynamic communication network and the solution to the assignment problem is reached without any common coordinator or a shared memory of the system.

To solve the problem in this context, we developed a distributed version of a Hungarian Method with the augmenting paths from the graph theory that we recall briefly in the following.

We refer to the centralized (Hungarian) method implemented by means of so-called alternating trees (see, e.g., Burkard and Çela, 1999). The algorithm is iterative and, maintaining a feasible dual solution, considers the admissible bipartite graph $\bar{G} = (T \bigcup R, \bar{E})$ where $\bar{E} = \{(i, j) : c_{ij} + \alpha_i - \beta_j = 0\}$. The algorithm searches for a matching of the maximal cardinality in the graph $\bar{G}$. If the matching is perfect, i.e., every robot is matched with a task, then the matching represents the optimal solution and the algorithm stops. If the matching is not perfect, the algorithm updates the dual variables so as to increase the dual objective function such that at least one new admissible edge is added to $\bar{G}$, and continues with a new iteration.

Given $\bar{G}$, let $M \subseteq \bar{E}$ be the current maximal matching in $\bar{G}$. The edges belonging to M are called matched edges, and the ones in $\bar{E} \backslash M$ are free edges. Given $\bar{G}$ and the current maximal matching $M \subseteq \bar{E}$, the algorithm iteratively improves the matching along augmenting paths over alternating trees in $\bar{G}$. An alternating tree is a subgraph of $\bar{G}$ rooted at some free task vertex t connected to all its adjacent robot vertices through free edges of $\bar{E}$, and with each robot vertex connected to a task vertex through a matched edge. When the above mentioned tree reaches a task leaf vertex which is adjacent to some free robot vertex r, an augmenting path is found, i.e., a path that lets the augmentation of the cardinality of matching exchanging the free and matched edges. When all the possible augmentation steps are done, the dual variables are updated and the new admissible edges are added. Then a new iteration begins searching for the maximum matching on the new admissible graph (see, e.g., Burkard and Çela, 1999).

In our implementation we maintain the forest F^1 of all the alternating trees rooted in free task vertices. Moreover, we maintain a forest F^2 of the alternating trees in $\bar{G}$ rooted in robot vertices containing all the robot-task vertices not contained in F^1. Clearly, the latter are also not connected with vertices in F^1.

Initially, the forests are set up in the following way. Forest F^1 is composed of n isolated task vertices. Forest F^2, in the similar way, is made of n isolated robot vertices. Dual variables are assumed to be $\alpha_i = 0$ for all the tasks i, and $\beta_j = \min_i c_{ij}$, for all the robots j, and are updated as follows. Let $minSlack_j = \min_{i \in F^1} (c_{ij} + \alpha_i - \beta_j)$ and n_j be the task corresponding to the $\arg\min$, for each robot vertex $j \in F^2$, and $\delta = \min_{j \in F^2} minSlack_j$. New values of the dual variables are:

$$\alpha_i := \alpha_i - \delta \ \ , \forall \ \text{ task vertex } \ \ i \ \in F^1$$
$$\beta_j := \beta_j - \delta \ \ , \forall \ \text{ robot vertex } \ \ j \ \in F^1$$

After updating the dual variables, even though there might be more than one new admissible edge, w.l.o.g., we can add the new edges one at a time and after adding only one edge, search for a new matching on the augmented admissible graph. The new edge connects the two forests. In more detail, the edge starts at the task vertex t^* of F^1 and ends on the robot vertex r^* of F^2. If r^* is free (unmatched), then there is an augmenting path $(t, \ldots, t^*, r^*)$ starting from the root task vertex of F^1, connected with t^* through the alternating path, and ending in r^*. By exchanging the free and matched edges in this augmenting path we get the augmented matching. Since task vertex t is no more free, all the tree of F^1 rooted in t moves to F^2 and is connected to root r^* over t^*. If r^* is not free, i.e., it is matched, there is no augmenting path and the two forests of alternating trees are updated by disconnecting robot r^* with all its subtree from forest F^2 and connecting it with all its subtree to the task vertex t^* in forest F^1.

4.1 Algorithm Details

The robots interactively execute the steps of the algorithm based on the local information and the one received through the messages from their neighbors in a communication graph. The structure of the communication graph is made of the forests F^1 and F^2 restricted to robots. The interconnected robots recalculate the structure of the forests each time the forests change. Therefore, the connection graph is updated each time the update of the forests F^1 and F^2 occurs.

Each robot keeps in its memory 4 pointers relating to the robots on its neighboring positions in the tree, i.e., the father robot, the oldest son, next older brother, and next younger brother, as seen from the Fig. 1. The first two pointers are used to move upwards and downwards in the tree while the two latter ones are used to explore a branch. Each robot in its memory also keeps the data about its position (whether it is in F^1 or F^2), its matched task M_j (if there is none, $M_j = 0$), and the father task vertex. Each robot j memorizes also the column vector j of the matrix of the distances, that is, its vector of distances from its position to the positions of all the tasks, its dual variable β_j, and the $minSlack_j$ along with the task n_j which obtains the $minSlack_j$.

Through autonomous calculations and the communication with the neighbors, robots get and share the information about α_i of all the tasks i, δ value for the dual variables' update, the tasks which are in F^1, the new edge between the forests F^1 and F^2, and the root robots $r(F^1)$ and $r(F^2)$ of forests F^1 and F^2 respectively (the first robot vertex in a depth-first search of the forest). In this way, there is no common coordinator or a shared memory of the robots' system. In detail, the robots, depending on the position in the forests, (e.g., whether they are in the augmenting path, in forest F^1 or F^2, etc.) change their roles,

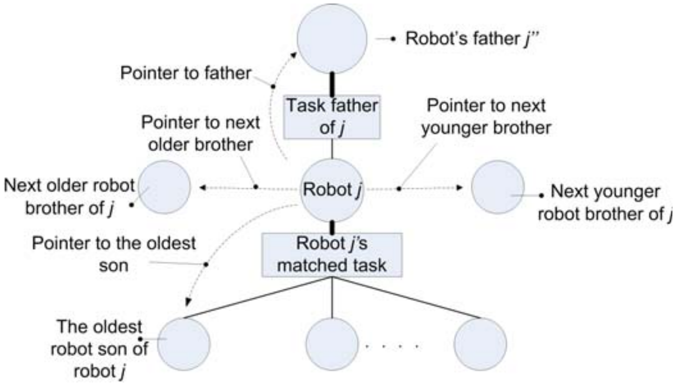

Robot's father j"
Pointer to father
Task father of j
Pointer to next younger brother
Pointer to next older brother
Next older robot brother of j
Robot j
Next younger robot brother of j
Pointer to the oldest son
Robot j's matched task
The oldest robot son of robot j
. . . .

Fig. 1. Four pointers of each robot in the tree

and accordingly do some of the steps of the distributed algorithm. In the following we describe the robots actions according to their roles.

Initialization (done by each robot j):

- Let $M_j := 0$ (robot j is initially unmatched); let $\alpha_i := 0$ for all the tasks i, and $\beta_j := \min_i\{c_{ij}\}$.
- Initialize the pointers in such a way that robot j is connected only with its younger $(j-1)$ and older brother robot $(j+1)$ and has no father and no sons. This means that the forests are initialized as described above.
- Each robot positioned in F^2 calculates $[minSlack_j, n_j]$ without exchanging any message with the others.

The root robots $r(F^1)$ and $r(F^2)$ start the message exchange based on depth-first search (DFS) asking if there is an unmatched robot. In the contact with the first unmatched robot, the same informs $r(F^2)$ to start a new iteration. In the following we describe the actions of the robots in respect to their roles during one iteration.

- All the robots $j \in F^2$ participate in calculating $[\delta, (t^*, r^*)]$. The calculation of δ starts from the root robot $r(F^2)$ and follows via the exchange of messages through the depth-first search of F^2. When the values of $[\delta, (t^*, r^*)]$ are found, the information is transmitted to all the robots in F^2 through message exchange based on depth-first search (DFS) starting from the root robot of F^2. The same DFS message passing applies to inform the robots in F^1.
- All the robots j do the following: for all $t_i \in F^1$ set $\alpha_i := \alpha_i - \delta$; moreover, if j belongs to F^1, set $\beta_j := \beta_j - \delta$. No messages are exchanged in this phase.
- The father robot of t^* (if any), returns the information about himself and about his oldest son to all the robots through the DFS message passing.

- Root robot $r(F^2)$ informs r^* in F^2 to start the next phase of augmenting or non augmenting the path.
- If r^* is not matched, then it initiates augmenting. Over the message passing from r^* and the robots in the augmenting path, robot r^* orders the swapping of the free and the matched edges along the augmenting path. It sends the messages to the robots in F^2 with the information about the new connection with the subtree of task t^* in F^1. The information contains t^*, and all the objects in the tree going backwards in F^1 up to the first task without father along the augmenting path $(t, \ldots, t^*, r^*)$. The robots simultaneously update the matching in respect to the messages received from r^* and update the pointers to the adjacent robots so that the tree of t^* in F^1 rooted in t moves to F^2 and gets connected to root r^* over t^*. The robots through the DFS message passing get the updated information regarding the position of the tasks in respect to the forests. Each robot positioned in F^2 calculates the new $[minSlack_j, n_j]$ without exchanging any messages with the others.
- If r^* is already matched, then it does not augment the matching. It sends the message to $r(F^2)$ to inform, through the DFS message passing, all the robots in F^2 to decrease minimum slack by δ; r^* disconnects from tree F^2 attaching itself together with its subtree to F^1 over t^*. Each robot in F^2 updates the new minimum slack for each task in the subtree routed at r^* in F^1, which results in $O(n^2)$ messages exchanged.

Regarding the robustness of the method, if the robot during the execution of the algorithm stops responding, it is considered erroneous and is eliminated from the further calculations. In the case where the robot was unmatched in the forest F^2, the calculation continues without any modifications, ignoring the robot in question. Otherwise, the algorithm starts from the beginning excluding the same.

5 Simulation Results

The simulation was performed in the C language and executed on a PC with an Intel Core Duo 1.6 GHz CPU and 1 GB of RAM. For each possible n varying from 10 to 250 we considered the average of 10 different instances consisting of sparse random cost matrices with the sparsity varying from 2 to 10% and the cost $c_{ij} \in (0, 100)$.

From the experiments it turns out that the cumulative computational time of the decentralized method, due to the exchange of messages among the robots, is from one to two orders higher than the computational time of the centralized algorithm, but is still limited by the $O(n^3)$ time (see Fig. 2, left). The number of messages exchanged among the robots is limited as well by $O(n^3)$ (see Fig. 2, center). Even if the decentralized method is cumulatively heavier, the computational time performed by each robot (in the order of $O(n^2)$) results minor than the time required by the standard (centralized) Hungarian algorithm (see Fig. 2, right).

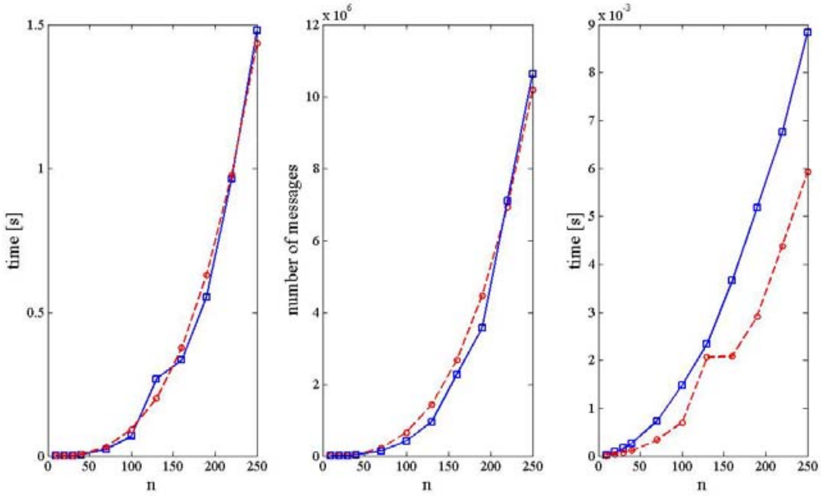


Fig. 2. Left: cumulative time of the decentralized algorithm (solid) vs. n^3 best fitting curve (dashed); center: total number of messages (solid) vs. n^3 best fitting curve (dashed); right: computational time of the (centralized) Hungarian algorithm (solid) vs. computational time of each robot in the decentralized method (dashed).

6 Conclusion

A decentralized implementation of the Hungarian algorithm has been presented to solve an allocation problem for a set of n robots which must execute a given number of tasks in the framework of a decentralized architecture where a centralized controller and a shared memory are not available. The robots must operate autonomously and the decentralized implementation is possible on the basis of an exchange of messages. We have assumed an equal number of robots and tasks but the more general case can be easily handled by introducing fictitious robots/tasks with fictitious costs; also the case where robots can execute only a subset of tasks can be handled with a proper choice of the matrix costs. The cumulative execution time of the decentralized algorithm and the total number of messages exchanged are both in the order of $O(n^3)$, resulting in an average computational payload $O(n^2)$ for each robot. A decentralized implementation is intrinsically robust: how robot failures can be handled is introduced and will be extended in the future research.

References

Arslan, G., Marden, J.R., Shamma, J.S.: Autonomous vehicle-target assignment: A game-theoretical formulation. Journal of Dynamic Systems, Measurement, and Control 129, 584–596 (2007)

Bertsekas, D.P.: Linear Network Optimization: Algorithms and Codes. MIT Press, Cambridge (1991)

Bertsekas, D.P.: Auction algorithms for network flow problems: A tutorial introduction. Computational Optimization and Applications 1(1) (1992)

Bertsekas, D.P., Castanon, D.A., Eckstein, J., Zenios, S.: Parallel computing in network optimization. In: Network Models - Handbooks in Operations Research and Management Science, vol. 7, pp. 330–399. Elsevier, Amsterdam (1995)

Burkard, R.E., Çela, E.: Linear assignment problems and extensions. Handbook of Combinatorial Optimization 4(1), 221–300 (1999)

Gerkey, B.P., Mataric, M.J.: A framework for studying multi-robot task allocation. In: Shultz, A.C., et al. (eds.) Multi-Robot Systems: From Swarms to Intelligent Automata, The Netherlands, vol. 2, pp. 15–26. Kluwer Academic Publishers, Dordrecht (2003)

Gerkey, B.P., Mataric, M.J.: A formal analysis and taxonomy of task allocation in multi-robot systems. International Journal of Robotics Research 23(9), 939–954 (2004)

Giordani, S., Lujak, M., Martinelli, F.: A Decentralized Scheduling Policy for a Dynamically Reconfigurable Production System. In: Mařík, V., Strasser, T., Zoitl, A. (eds.) Holonic and Multi-Agent Systems for Manufacturing. LNCS (LNAI), vol. 5696, pp. 102–113. Springer, Heidelberg (2009)

Kuhn, H.W.: The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly 2, 83–97 (1955)

Kwok, K.S., Driessen, B.J., Phillips, C.A., Tovey, C.A.: Analyzing the multiple-target-multiple-agent scenario using optimal assignment algorithms. Journal of Intelligent and Robotic Systems 35(1), 111–122 (2002)

Lawler, E.L.: Combinatorial Optimization: Networks and Matroids. Holt, Rinehart, and Winston (1976)

Lueth, T.C., Laengle, T.: Task description, decomposition and allocation in a distributed autonomous multi-agent robot system. In: IEEE/RSJ IROS, pp. 1516–1523 (1994)

Papadimitriou, C.H., Steiglitz, K.: Combinatorial optimization: algorithms and complexity. Prentice-Hall, Inc., Englewood Cliffs (1982)

Smith, S.L., Bullo, F.: Target assignment for robotic networks: Asymptotic performance under limited communication. In: American Control Conference, ACC'07, pp. 1155–1160 (2007)

Sugar, T., Kumar, V.: Control and Coordination of Multiple Mobile Robots in Manipulation and Material Handling Tasks. In: The Sixth International Symposium on Experimental Robotics, vol. VI, pp. 15–24. Springer, Heidelberg (2000)

Wooldridge, M.: Introduction to Multi-Agent Systems. John Wiley and Sons, Chichester (2002)

Zavlanos, M.M., Spesivtsev, L., Pappas, G.J.: A distributed auction algorithm for the assignment problem. In: Proc. of 47th IEEE Conf. on Decision and Control, Cancun, Mexico, pp. 1212–1217 (2008)

Comparison between Deterministic and Meta-heuristic Methods Applied to Ancillary Services Dispatch

Zita A. Vale, Carlos Ramos, Pedro Faria, João P. Soares, Bruno Canizes,
Joaquim Teixeira, and Hussein M. Khodr

GECAD – Knowledge Engineering and Decision-Support Research Center
Institute of Engineering – Polytechnic of Porto (ISEP/IPP)
Rua Dr. António Bernardino de Almeida, 431
4200-072 Porto
Portugal
{zav,csr,pnf,japs,bmrc,jra,hmk}@isep.ipp.pt

Abstract. This paper proposes two meta-heuristics (Genetic Algorithm and Evolutionary Particle Swarm Optimization) for solving a 15 bid-based case of Ancillary Services Dispatch in an Electricity Market. A Linear Programming approach is also included for comparison purposes. A test case based on the dispatch of Regulation Down, Regulation Up, Spinning Reserve and Non-Spinning Reserve services is used to demonstrate that the use of meta-heuristics is suitable for solving this kind of optimization problem. Faster execution times and lower computational resources requirements are the most relevant advantages of the used meta-heuristics when compared with the Linear Programming approach.

Keywords: Artificial Intelligence Techniques, Ancillary Services Dispatch, Electricity Markets, Evolutionary Particle Swarm Optimization, Genetic Algorithm, Linear Programming.

1 Introduction

In electricity markets, Ancillary Services (AS) can be seen as a set of products separated from the energy transactions. Adequate procedures and methodologies to determine AS needs as well as the ways they are obtained and priced are required in a competitive environment. AS are essential for ensuring the reliable operation and security of electric power systems [1]. According to this definition, many services such as voltage support, losses, black start capacity, regulation, load shedding and reserves with varying levels of response time are considered as AS. Modern meta-heuristics algorithms like Genetic Algorithms and Evolutionary Particle Swarm Optimization are considered effective tools for optimization problems [2]. Therefore Artificial Intelligence techniques can be applied to solve the dispatch of AS. In this paper the optimization process of an AS dispatch for a 15 bid-based case is presented using Linear Programming (LP), Genetic Algorithm (GA) and Evolutionary Particle

N. García -Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 731–741, 2010.

Swarm Optimization (EPSO). The aim is to demonstrate the effectiveness ot the use of Artificial Intelligence techniques to solve the problem of AS dispatch.

2 Problem Description

Electricity markets have as the underlying negotiated product the electrical energy required by the demand side and the energy transactions occur in the scope of the energy market. AS services are important to achieve targeted objectives on power system security, frequency stability, as well as voltage level and stability. The Independent System Operator (ISO) is responsible for the secure operation of the power system, having to determine the required AS. For this reason, Ancillary Services Market (ASM) clearing methods are required and this can be done using optimization methods that consider technical and economic issues. The formulation presented in this paper considers Regulation Down, Regulation Up, as well as Spin Reserves and Non-spin Reserves services. These services are defined as follows:

- Regulation: Generating units that are already operating and synchronized with the ISO controlled grid, so that the active power produced can be incremented or decreased instantaneously through automatic generation control. Regulation is used to maintain real-time balance on the system;
- Spin Reserves: Are available nearly instantaneously and are able to stabilize system frequency upon the outage of a large generating unit in the system. Their response time is from a few seconds to about 5 minutes or less and requires no notification;
- Non-spin Reserves: Their time frame is from several minutes to half an hour or more. While they can be exercised quickly and reliably, they will often require some manual intervention for their activation. These reserves are needed not only to stabilize system frequency, but also to deal with energy balancing within a control area.

Generators can provide power up to their maximum capacity. This capacity can be split into energy and reserves. They are the same original physical product but, on the other hand, they are distinct commercial products. Producers can sell both products in order to maximize their profits subject to the units' capacities. From the point of view of the market the goal is to get the required quantities at the minimum possible cost. This requires allocating plants efficiently for the energy and reserve markets. ASM clearing mechanisms aim at dispatching the required AS at minimum cost. For the case study presented in this paper, the results obtained by a Linear Programming approach are compared with the results obtained with the Genetic Algorithm and the Evolutionary Particle Swarm Optimization approaches, the three methods having been implemented in MATLAB.

The Ancillary Services Dispatch Optimization can be formulated by (1) to (3) for Regulation Down and by (4) to (9) for Regulation Up, Spin Reserves and Non-spin Reserves.

Regulation down dispatch is done independently from the other services and does not have an effect on the maximum available capacity of market players [3-5]:

$$MIN\sum_{i=1}^{N}\left(PR_{i,k}+p\cdot PE_{i,k}\right)\cdot X_{i,k} \qquad (k=1) \tag{1}$$

$$\text{Subject to: } \sum_{i=1}^{N} X_{i,1} = Q_1 \tag{2}$$

$$0 \le X_{i,1} \le C_{i,1} \; and \; X_{i,1} \, integer \tag{3}$$

$$MIN\sum_{k=2}^{4}\sum_{i=1}^{N}(PR_{i,k}+p.PE_{i,k})*X_{i,k} \tag{4}$$

$$\text{Subject to: } \sum_{i=1}^{N} X_{i,2} = Q_2 \tag{5}$$

$$\sum_{i=1}^{N} X_{i,3} = Q_3 \tag{6}$$

$$\sum_{i=1}^{N} X_{i,4} = Q_4 \tag{7}$$

$$\sum_{k=2}^{4} X_{i,k} \le Cm\acute{a}x_i \qquad \text{I=1,...,N} \tag{8}$$

$$0 \le X_{i,k} \le C_{i,k} \text{ and } X_{i,k} \text{ integer} \tag{9}$$

where N is the total number of bids, i is the bid index (i=1,2,….,N), k is the ancillary service index (1 for regulation down, 2 for regulation up, 3 for spin, and 4 for non-spin), Q_k is the total capacity requirement for ancillary service k, $Cmax_i$ is the maximum capacity of bid i, C_{ik} is the capacity of bid i for ancillary service k, $X_{i,k}$ is the accepted capacity of bid i for ancillary service k, $PR_{i,k}$ is the capacity reserve price of bid i for ancillary service k, $PE_{i,k}$ is the energy reserve price of bid i for ancillary service k, and p is the estimated probability of using reserve energy acquired for AS.

3 Methodology Approach

The nature of the formulated problem is combinatorial. Several solution techniques have been proposed such as heuristics [6], dynamic programming [7], mixed-integer linear programming [8], Lagrangian relaxation [9], simulated annealing [10] and evolution-inspired approaches [11]. In this section, three different approaches are presented: Linear Programming, Genetic Algorithm, and Evolutionary Particle Swarm Optimization.

A) Linear Programming (LP)

If the quantity required by the System Operator (SO) for regulation down is satisfied, then the global process continues, otherwise there is no feasible solution. In order to process the other services, firstly all the prices are combined and then all repeated combinations are discarded; secondly the combinations that do not satisfy the constraints are also discarded. Next, for each combination J, the optimization method is applied and the results (solutions) are stored until the total unique combination number (CN) is achieved. Finally, when there are no more combinations to dispatch, the process displays the best stored solution combined with regulation down dispatch. This mathematical model has been implemented using MATLAB Optimization Toolbox for Linear Programming.

As Linear programming guarantees the optimal solution it has been chosen to solve AS dispatch as a comparison reference with GA and EPSO. Figure 1 represents a Linear Programming simultaneous optimization process to solve ASM clearing price.

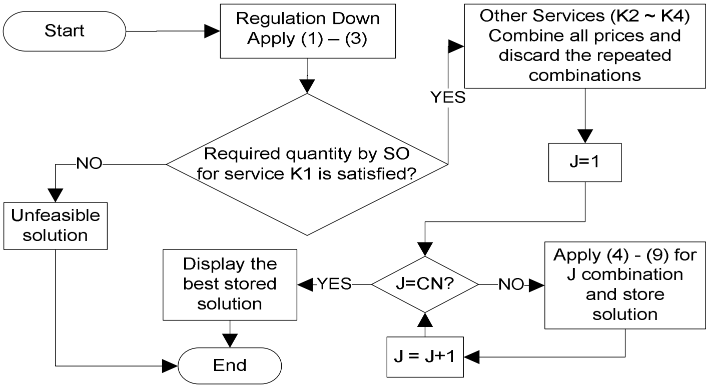


Fig. 1. Linear Programming process flowchart

B) Genetic Algorithm (GA)

The GA works with a set of codified solutions that constitutes a population. Such population is able to evolve by application of selection, mutation, reproduction and crossover as can be seen in Figure 2. The strongest individuals (solutions) survive during the optimization process or generations.

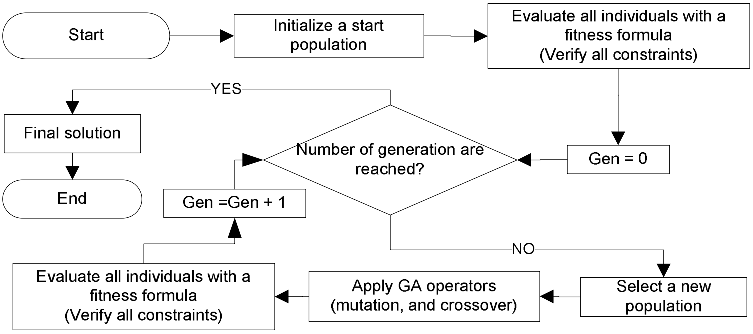


Fig. 2. Genetic Algorithm process flowchart

Firstly, an initial population (also known as start population) is randomly generated. The select operation creates an intermediate population that picks up the best individuals among the initial population by means of a fitness evaluation function (cost). To obtain an evolved population it is necessary to apply some operators (crossover and mutation) that allow to obtain a new generation. This paper presents a GA implementation in MATLAB, using the Genetic Algorithm Optimization Toolbox (GAOT) for MATLAB 5. GAOT implements simulated evolution in the MATLAB environment using both binary and real representations [12]. This implementation is very flexible in what concerns the genetic operators, selection function, termination functions as well as evolution functions that can be used. The undertaken implementation is described in [12] and adopted in this work. Each individual of the GA population represents the quantity accepted for each bid i. Each individual has 45 values (45 bids) for regulation up, spin reserve and non-spin reserve dispatch and 15 values (15 bids) for regulation down dispatch. The initial population represents the number of randomly generated initial solutions. The crossover, mutation, select operator and number of generations used in this algorithm are discussed in the Case Study section.

The implementation of the GA approach consists of the following: the Regulation Down Dispatch and the other 3 services use the parameters of table 2. Population size is 200 in both cases and each individual within the population has 15 values for the Regulation Down Dispatch and 15 for each of the 3 other services. The meaning of each value is the quantity accepted for each bid.

C) Evolutionary Particle Swarm Optimization (EPSO)

When compared with GA, the classical PSO has no selection and mutation operator [13, 14]. On the other hand, Evolutionary Particle Swarm Optimization (EPSO) introduces a new approach: mutation of the strategic parameters and selection by stochastic tournament of particles passing to the next generation. The movement rule is defined by three strategic parameters (see Figure 3): inertia, memory and cooperation [15, 16].

Comparing with other adaptive evolutionary methods, EPSO is different in its adaptive recombination operator. While usually the adaptive operator in other evolutionary algorithms is the mutation operator, EPSO relies on evolving weights in the movement equation, instead of an explicit random factor. Therefore, EPSO is less dependent on externally defined parameters by the user, with values that are depend on the type of the problem. EPSO has proven to be efficient, accurate and robust, and with successful applications to power system problems [17, 18]. Given a population as a set of particles, the general of EPSO algorithm can be explained as follows:

Replication: Each particle is replicated one ($r = 1$) time.

Mutation: The strategic parameters w of each particle are mutated according to:

$$*w_i = w_i + [LogN(0,1) \times \delta] \tag{10}$$

In this paper the lognormal mutation to the particle's weight was used where δ is the learning parameter, externally fixed. Note that the Strategic parameters are randomly initialized between 0 and 1. Reproduction: Each particle generates one descendent according to the movement equation:

$$*v_i = w_i.inertia(v_i) + w_i.memory(b_i - X_i) + w_i.coop(*bG - X_i) \tag{11}$$

$$*bG = bG + w_i.deviation\,N(0,1) \tag{12}$$

Finally, the basic particle movement rule is:

$$*X_i = X_i + *v_i \tag{13}$$

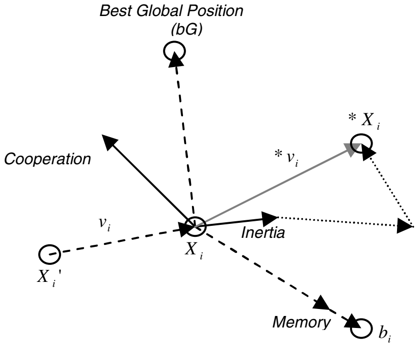


Fig. 3. Spatial representation of EPSO movement rule

Evaluation: Each particle has its fitness value, according to its current position in the search space.

Selection: By stochastic tournament, the best particle of each group of *r+1* descendents (population + *r* replicas) of each individual of previous generation is selected to form the new generation. Figure 4 presents the EPSO algorithm flowchart.

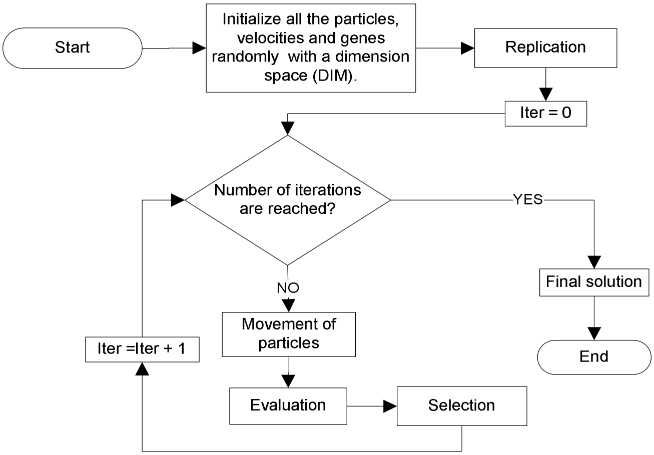

Start
Initialize all the particles, velocities and genes randomly with a dimension space (DIM).
Replication
Iter = 0
Number of iterations are reached?
YES
Final solution
End
NO
Movement of particles
Evaluation
Selection
Iter =Iter + 1

Fig. 4. Evolutionary Particle Swarm Optimization process flowchart

4 Case Study

This section presents the application of LP, GA and EPSO approaches in order to perform the dispatch of the active power reserves to a case study. Table 1 presents the data for this case study. These data include the AS bids for a set of 15 players and the values of the required power reserve for each service, which are displayed in the last row of Table 1. Quantities (*Qt*) are expressed in MW, total capacity requirement (*Rq*) in MW, reserve prices (*Pr*) in mu/MW and energy prices (*Pe*) in mu/MWh, where *m.u.* stands for monetary units.

Table 1. Ancillary service bids and requirements

Bids	Ancillary Services												
	Max. Power	Regulation Down			Regulation Up			Spin Reserve			Non-Spin Reserve		
		Qt	PR	PE	Qt	PR	PE	Qt	PR	PE	Qt	PR	PE
	(MW)	(MW)	(mu/MW)	(mu/ MWh)	(MW)	(mu/ MW)	(mu/ MWh)	(MW)	(mu/ MW)	(mu/ MWh)	(MW)	(mu/ MW)	(mu/ MWh)
1	50	30	4.2	38.8	30	5.0	38.8	40	6.0	34.6	45	6.0	34.6
2	45	30	3.3	44.0	35	4.0	44.0	35	5.0	40.6	35	5.0	40.6
3	40	25	3.5	42.8	30	4.2	42.8	35	4.5	41.8	40	5.0	41.8
4	50	30	4.2	39.6	30	5.0	39.6	30	6.2	36.8	35	6.6	36.8
5	30	30	3.7	41.3	35	4.0	41.3	30	5.0	38.0	30	6.0	38.0
6	45	25	3.9	43.4	20	4.6	43.4	40	5.0	39.4	40	5.0	39.4
7	50	30	4.0	42.4	20	4.5	42.4	30	5.3	38.0	30	5.5	38.0
8	55	40	3.8	42.5	35	4.6	42.5	40	5.6	37.6	45	6.0	37.6
9	55	45	4.2	38.6	30	5.5	38.6	50	6.3	34.4	40	6.3	34.4
10	50	40	4.0	41.5	40	5.0	41.5	40	6.0	37.4	30	6.0	37.4
11	60	35	3.8	42.8	40	4.0	42.8	45	5.0	39.4	45	5.0	39.4
12	50	30	3.2	43.6	30	4.3	43.6	30	4.8	42.0	35	5.0	42.0
13	65	30	4.2	38.5	35	5.3	38.5	45	5.3	34.9	50	5.4	34.9
14	50	30	4.4	39.6	30	5.2	39.6	30	6.2	35.2	35	6.6	35.2
15	45	25	3.9	43.4	20	4.6	43.4	40	5.0	39.4	40	5.0	39.4
Rq	**587**	**141**			**235**			**117**			**94**		

Table 2 depicts the crossover, mutation, select operator and number of generations that have been used.

Table 2. Tuning parameters GA

ATTRIBUTE	DESCRIPTION	USED VALUE REGULATION DOWN	USED VALUE REGULATION UP, SPIN RESERVE AND NON-SPIN RESERVE
ε	Epsilon indicates the threshold to find the best solution	10–10	10–10
Crossover operator	Arithmetic Crossover	25 crossover operations per generation	12 crossover operations per generation
Mutation operator	Boundary Mutation	10 mutations per generation	60 mutations per generation
Select Operator	Normalized geometric distribution	Probability of 1.00	Probability of 1.00
Select Operator 2	Roulette method	-	-
Initial population	Population size	200	200
Number of generations	Number of generations applied to start population	200	200

Table 3 presents more detailed information for the EPSO algorithm, including the number of iterations and the used mutation method, among others

Table 3. Tuning parameters EPSO

Attribute	Description	Used value Regulation Down	Used value Regulation up, spin reserve and non-spin reserve
Number of particles	Number of particles	40	40
Search space	Dimension of values for each particle	15	45
Number of iterations	Number of iterations applied to swarm	60	60
τ	Tau indicates the mutation factor value	0.6	0.6
Mutation method	Lognormal mutation	$^*w_i = w_i + [LogN(0,1)\times\delta]$	$^*w_i = w_i + [LogN(0,1)\times\delta]$

Table 4 presents a synthesis of the obtained results for LP, GA and EPSO approaches. These results consider 100% probability for the effective use of the reserve service (p=1). *Total* represents the capacity requirement, *MP* refers to the market clearing price and *Final Cost* is the reserve cost plus energy cost. It can be seen that LP guarantees a better solution than GA and EPSO; however these two last methods achieve a very reasonable solution. The cost differences for GA and EPSO are 464.9 and 910.3 *m.u* respectively, when compared with LP.

Table 4. Ancillary services dispatch results

	Regulation Down			**Regulation Up**			**Spin Reserve**			**Non-Spin Reserve**			**Total Cost (mu)**		
	LP	**GA**	**EPSO**	**LP**	**GA**	**EPSO**	**LP**	**GA**	**EPSO**	**LP**	**GA**	**EPSO**	**LP**	**GA**	**EPSO**
Total (MW)	141	141	141	235	235	235	117	117	117	94	94	94	587	587	587
MP (mu/MW)	4.4	4.2	4.4	5.3	5.3	5.5	6.3	6.2	6.3	6.0	6.6	6.6			
Reserve Cost (mu)	620.4	592.2	620.4	1245.5	1245.5	1292.5	737.1	725.4	737.1	564	620.4	620.4	3167.0	3183.5	3270.4
Final Cost (mu)													**25528.4**	**25993.3**	**26438.7**

All cases have been tested on a PC compatible with 2 processors Intel Xeon X5450 3.0 GHz, each one with 4 Cores, 4GB of Random-Access-Memory (RAM) and Windows Server 2008 Operating System. Figure 4 shows that EPSO is the faster method with 67% less computational time. Figure 5 shows that EPSO is also the method with lower consumed resources (RAM) with a reduction of 98% in comparison with LP. The LP process performs 432 price combinations with 2797 iterations under 7 seconds.

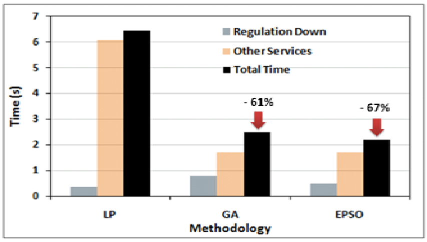


Fig. 4. Time used by methodologies

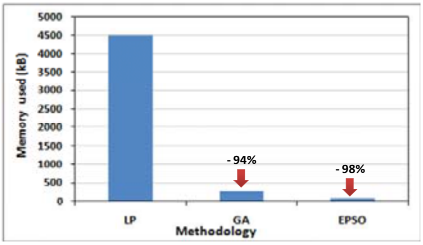


Fig. 5. Memory used by methodologies

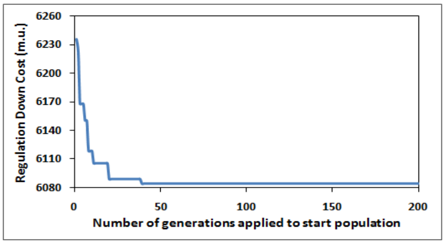


Fig. 6. Regulation Down Costs in m.u. using GA

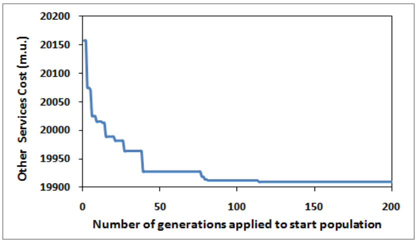


Fig. 7. Other Services Costs in m.u. using GA

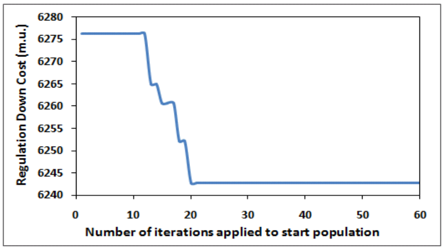


Fig. 8. Regulation Down costs in m.u. using EPSO

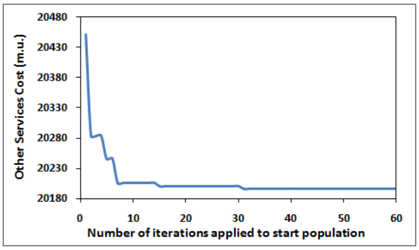


Fig. 9. Other Services costs in m.u. using EPSO

When comparing the results shown in Figures 6 to 9, it can be concluded that the number of generations needed by the EPSO approach to achieve fitness cost stabilization is lower than GA. This comparison can be made by analyzing figure 6 along with figure 8 (stabilization after 40 generations in GA and after 20 iterations in EPSO). Comparing figure 7 with figure 9 the stabilization is obtained after 120 generations in GA and after 33 iterations in EPSO.

5 Conclusions

Ancillary services are an important component of electricity markets as they represent a business opportunity for market players.

This paper presents a case-study that compares the results obtained with ancillary services market simulation using three methodologies: Linear Programming, Genetic algorithm and Evolutionary Particle Swarm Optimization .

LP guarantees always the best solution while EPSO is faster and consumes fewer resources, positioning this methodology as a levelheaded approach to solve this kind of problems. GA presents satisfactory solutions and performance, being less effective in terms of resources used and computation time when compared with EPSO.

Acknowledgments

The authors would like to acknowledge FCT, FEDER, POCTI, POSI, POCI, POSC and PTDC for their support to R&D Projects and GECAD Unit.

References

[1] Motamedi, A., Fotuhi-Firuzabad, M.: Ancillary Service Markets Design. In: 2007 Large Engineering Systems Conference on Power Engineering (LESCOPE07), Montreal, Quebec, Canada, October 10-12, pp. 316–320 (2007)

[2] Burade, P.G., Helonde, J.B.: A Novel Approach for Optimal Power Dispatch Using Artificial Intelligence (AI) Methods. In: International Conference on Control Automation, Communication and Energy Conservation, ICCACEC 2009, June 4-6 (2009)

[3] Papalexopoulos, A., Singh, H.: On the Various Design Options for Ancillary Services Markets. In: 34th Annual Hawaii International Conference on System Sciences, Maui, Hawaii, Janaury 2002, pp. 798–805 (2002)

[4] Pereira, A.J.C., Vale, Z.A., Machado e Moura, A., Dias Pinto, J.A.: Provision and Costs of Ancillary Services in a Restructured Electricity Market. In: International Conference on Renewable Energy and Power Quality (ICREPQ'04), March 31-April 02 (2004)

[5] Vale, Z., Ramos, C., Faria, P., Soares, J.P., Canizes, B.R., Khodr, H.M.: Ancillary Service Market Simulation. In: IEEE T&D Asia, Seoul, Korea (October 2009)

[6] Li, C., Johnson, R.B., Svoboda, A.J.: A new unit commitment method. IEEE Trans. Power Syst. 12(1), 113–119 (1997)

[7] Ouyang, Z., Shahidehpour, S.M.: An intelligent dynamic-programming for unit commitment application. IEEE Trans. Power Syst. 6(3), 1203–1209 (1991)

[8] Medina, J., Quintana, V.H., Conejo, A.J.: A clipping-off interior point technique for medium-term hydro-thermal coordination. IEEE Trans. Power Syst. 14(1), 266–273 (1999)

[9] Ongsakul, W., Petcharaks, N.: Unit commitment by enhanced adaptive Lagrangian relaxation. IEEE Trans. Power Syst. 19(1), 620–628 (2004)

[10] Purushothama, G.K., Jenkins, L.: Simulated annealing with local search—A hybrid algorithm for unit commitment. IEEE Trans. Power Syst. 18(1), 273–278 (2003)

[11] Rajan, C.C.A., Mohan, M.R.: An evolutionary programming-based tabu search method for solving the unit commitment problem. IEEE Trans. Power Syst. 19(1), 577–585 (2004)

[12] Houck, C., Joines, J., Kay, M.: A Genetic Algorithm for Function Optimization: A MatLab Implementation, NCSU-IE TR 95-09 (1995)

[13] Chaturvedi, K.T., Pandit, M., Srivastava, L.: Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch. IEEE Transactions on Power Systems 23(3), 1079–1087 (2008)

[14] Wu, J.K., Zhu, J.Q., Chen, G.T., et al.: A Hybrid Method for Optimal Scheduling of Short-Term Electric Power Generation of Cascaded Hydroelectric Plants Based on Particle Swarm Optimization and Chance-Constrained Programming. IEEE Transactions on Power Systems 23(4), 1570–1579 (2008)

[15] Huang, C.M., Wang, F.L.: An RBF network with OLS and EPSO algorithms for real-time power dispatch. IEEE Transactions on Power Systems 22(1), 96–104 (2007)

[16] Lee, T.Y.: Optimal spinning reserve for a wind-thermal power system using EIPSO. IEEE Transactions on Power Systems 22(4), 1612–1621 (2007)

[17] Leite, H., Barros, J., Miranda, V.: Evolutionary algorithm EPSO helping doubly-fed induction generators in ride-through-fault. In: PowerTech, 2009 IEEE Bucharest, June 28-July 2, pp. 1–8 (2009)

[18] Azevedo, F., Vale, Z., Moura Oliveira, P.: A Decision-Support System Based on Particle Swarm Optimization for Multiperiod Hedging in Electricity Markets. IEEE Transactions on Power Systems 22(3) (August 2007)

Domain-Dependent Planning Heuristics for Locating Containers in Maritime Terminals

Mario Rodríguez-Molins, Miguel Á. Salido, and Federico Barber

Instituto de Automática e Informática Industrial
Universidad Politécnica de Valencia.
Valencia, Spain

Abstract. Maritime container terminals are facilities where cargo containers are transshipped between ships and land vehicles. These terminals involve a large number of complex and combinatorial problems. One of the important problems is related to the Container Stacking Problem. A container stack is a type of temporary store where containers await further transport by truck, train or vessel. The main efficiency problem for an individual stack is to ensure easy access to containers at the expected time of transfer. Since stacks are 'last-in, first-out', and the cranes used to relocate containers within the stack are heavily used, the stacks must be maintained in a state that minimizes on-demand relocations. In this paper, we present a set of domain-dependent planning heuristics for finding the best allocation of containers in a yard-bay in order to minimize the number of reshuffles. To this end, the problem is distributed into a set of subproblems in order to be solved more efficiently. The resultant configuration of the yard will satisfy a set of constraints such as balanced bays, minimum distance for dangerous containers, etc.

1 Introduction

Maritime container terminals are the most important locations for transshipment and intermodal container transfers. Containers are an ISO standardized metal box of 8-ft wide by 8-ft high and 20-ft or 40-ft lengths. The container capacity is often expressed in twenty-foot equivalent unit (TEU). Containers are made out of steel and can be stacked on top of each other. Loading and offloading containers on the stack is performed by cranes. In order to access a container which is not at the top of its pile, those above it must be relocated. This reduces the productivity of the cranes. Maximizing the efficiency of this process leads to several requirements. First, each incoming container should be allocated a place in the stack which should be free and supported at the time of arrival. Second, each outgoing container should be easily accessible, and preferably close to its unloading position, at the time of its departure. In addition, there exist a set of hard/soft constraints regarding the container locations, for example, small differences in height of adjacent yard-bays, dangerous containers must be allocated separately by maintaining a minimum distance and so on.

Since the allocation of positions to containers is currently done more or less manually, this has convinced port operators that it should be possible to achieve significant improvements of lead times, storage utilization and throughput using appropriate and improved techniques.

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 742–751, 2010.

The overall goal collaboration between our group at the Technical University of Valencia (UPV) and the maritime container terminal MSC (Mediterranean Shipping Company S.A) is to offer assistance to help in planning and scheduling tasks such as the allocation of spaces to outbound containers, to identify bottlenecks, to determine the consequences of changes, to provide support in the resolution of incidents, to provide alternative planning of vessel arrivals, etc.

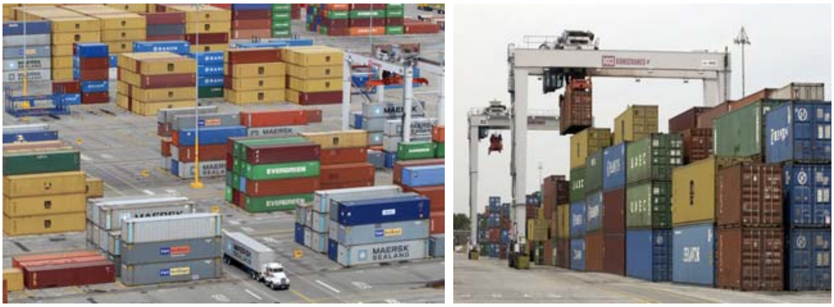


Fig. 1. A container yard (left) and gantry cranes (right) (Photos by Stephen Berend)

Figure 1 left shows a container yard. A yard consists of several blocks, and each block consists of 20-30 yard-bays [3]. Each yard-bay contains several (usually 6) rows. Each row has a maximum allowed tier (usually tier 4 or tier 5 for full containers). Figure 1 right shows a gantry crane that is able to move a container within a stacking area or to another location on the terminal. For safety reasons, it is usually prohibited to move the gantry crane while carrying a container [4], therefore these movements only take place in the same yard-bay.

When an outside truck delivers an outbound container to a yard, a transfer crane picks it up and stacks it in a yard-bay. During the ship loading operation, a transfer crane picks up the container and transfers it to a truck that delivers it to a quay crane.

In container terminals, the loading operation for export containers is pre-planned by load planners. For load planning, a containership agent usually transfers a load profile (an outline of a load plan) to a terminal operating company several days before a ship's arrival. The load profile specifies only the container group. In order to have an efficient load sequence, storage layout of export containers must have a good configuration. The main focus of this paper is optimally reallocating outgoing containers for the final storage layout from which a load planner can construct an efficient load sequence list. In this way, the objective is therefore to plan the movement of the cranes so as to minimize the number of reshuffles of containers in a complete yard. To this end, the yard is decomposed in yard-bays, so that the problem is distributed into a set of subproblems. Thus, each yard-bay generates a subproblem, but containers of different yard-bays must satisfy a set of constraints, so that subproblems will be sequentially solved taken into account the set of constraints with previously solved subproblems.

2 Problem Description (The Container Stacking Problem)

The Container Stacking Problem can be viewed as a modification of the *Blocks World* planning domain [9], which is a well-known domain in the planning community. This domain consists of a finite number of blocks stacked into towers on a table large enough to hold them all. The *Blocks World* planning problem is to turn an initial state of the blocks into a goal state, by moving one block at a time from the top of a tower onto another tower (or on a table). The optimal *Blocks World* planning problem is to do so in a minimal number of moves.

This problem is closed to the Container Stacking Problem, but there are some important differences:

- The number of towers is limited to 6 because a yard-bay contains usually 6 rows.
- The height of a tower is also limited to 4 or 5 tiers depending on the employed cranes.
- There exist a set of constraints that involve different rows such as balanced adjacent rows, dangerous containers located in different rows, etc.
- The main difference is in the problem goal specification. In the *Blocks World* domain the goal is to get the blocks arranged in a certain layout, specifying the final position of each block. In the container stacking problem the goal state is not defined as accurately, so many different layouts can be a solution for a problem. The goal is that the most immediate containers to load are in the top of the towers, without indicating which containers must be in each tower.

We can model our problem by using the standard encoding language for classical planning tasks called *PDDL* (Planning Domain Definition Language) [1]. Following this standard, a planning task is defined by means of two text files. On the one hand, the **domain file**, which contains the common features for all problems of this type. These features are: *Object types*, *Propositions* and *Actions*. On the other hand, the **problem file** which describes the particular characteristics of each problem:

- *Objects*: the rows available in the yard-bay (usually 6) and the containers stored in them.
- *Initial state*: the initial layout of the containers in the yard.
- *The goal specification*: the selected containers to be allocated at the top of the stacks or under other selected containers.
- *The metric function*: the function to optimize. In our case, we want to minimize the number of relocation movements (reshuffles).

More information about the domain file and problem file can be found in [5].

3 A Domain-Independent and Domain-Dependent Planner: New Heuristics

Since the Container Stacking Problem can be formalized in *PDDL* format (Section 2), we can use a general domain independent planner to solve our problem instances.

The plan, which is returned by the planner, is a totally ordered sequence of actions or movements which must be carried out by the crane to achieve the objective.

We have initially used a local search domain-independent planner called *Simplanner* [8]. This planner has several interesting properties for the container stacking problem: It is an anytime planning algorithm [10]; it is complete, so it will always find a solution if exists; and it is optimal, so that it guarantees finding the optimal plan if there is time enough for computation. Moreover, it follows an enforced hill-climbing [2] approach with some modifications: it applies a best-first search strategy to escape from plateaux. This search is guided by a combination of two heuristic functions and it allows the planner to escape from a local minimum very efficiently. If a plateau exit node is found within a search limit imposed, the hill-climbing search is resumed from the exit node. Otherwise, a new local search iteration is started from the best open node.

Simplanner was firstly used to solve individual subproblems (yard-bays). To improve the solutions obtained by *Simplanner* we have developed set of domain-dependent heuristics to guide the search in order to accelerate and guide the search toward a optimal or sub-optimal solutions.

The first heuristic (called H_1) was developed to efficiently solve each individual subproblems. H_1 computes an estimator of the number of container movements that must be carried out to reach a goal state (see Algorithm 1). The essential part of this algorithm is to count the number of containers located on the selected ones, but also keeps track of the containers that are held by the crane distinguishing between whether they are selected containers or not. When the crane is holding a selected container, the value h has a smaller increase since, although this state is not a solution, this container will be at the top of some row in the next movement. Some improvements for this heuristic are showed in [7].

The former heuristic (H_1) was developed to solve an individual subproblem (yard-bay). It was evaluated and compared against *Simplaner* in [7]. However, both *Simplaner* and H_1 were unable to solve a complete yard (with 20 yard-bays) due to the fact that they only solve individual subproblems. In this paper, we include new constraints that involve several yard-bays. These constraints are:

- Balancing contiguous yard-bays: rows of adjacent yard-bays must be balanced, that is, the difference between the number of containers of row j in yard-bay i and row j in yard-bay $i-1$ must be lower than 3. Figure 2 left shows an example of non-balanced yard-bays (rows in dotted points).
- dangerous containers: two dangerous containers must maintain a minimum security distance. Figure 2 right shows an example of two dangerous containers that does not satisfy the security distance constraint.

These constraints interrelate the yard-bays so the problem must be solved as a complete problem. However, it is a combinatorial problem and it is not possible to find an optimal or sub-optimal solution in a reasonable time. Following the previous philosophy of solving each subproblem independently (each yard-bay separately), we can distribute the problem into subproblems and solve them sequentially taken into account related yard-bays. Thus a solution to the first yard-bay is taken into account to solve the second yard-bay. A solution to the second yard-bay is taken into account

Algorithm 1. Pseudo-code of the domain-dependent heuristic H_1

```
Data: $b$: state of the yard-bay;
Result: $h$: heuristic value of $b$;
$h = 0$;
// Container hold by the crane
if $\exists x-container/\mathtt{Holding}(x) \in b$ then
    if $\mathtt{GoalContainer}(x)$ then
        $h = 0.1$;
    else
        $h = 0.5$;
    end
end
// Increasing the $\Delta h$ value
for $r \leftarrow 1$ to $\mathtt{numRows}(b)$ do
    $\Delta h = 0$;
    for $x-container/\mathtt{At}(x,r) \wedge \mathtt{GoalContainer}(x) \in b$ do
        if $\nexists y-container/\mathtt{GoalContainer}(y) \wedge \mathtt{On}(y,x) \in b$ then
            $\Delta h = \max(\Delta h, \mathtt{NumContainersOn}(x))$;
        end
    end
    $h+ = \Delta h$;
end
```

to solve the third yard-bay. Furthermore, if there exist a dangerous container in a first bay, its location is taken into account to solve a dangerous container located in the third yard-bay (if it exists); and so on. Taken into account this distributed and synchronous model, we present two different heuristics to manage these type of constraints.

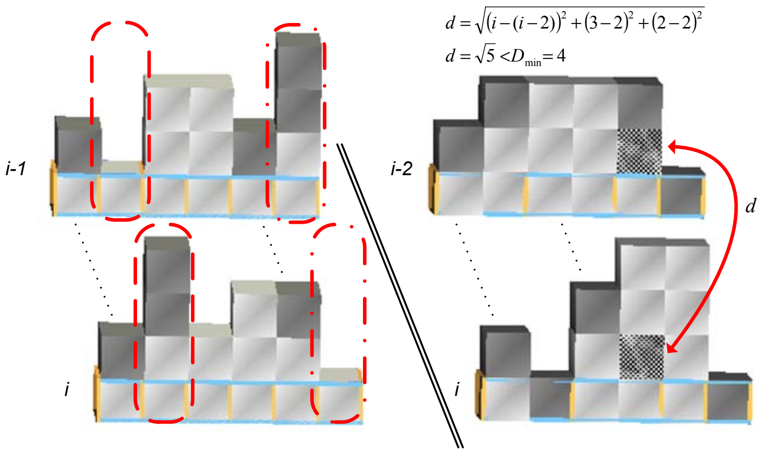

$d = \sqrt{(i-(i-2))^2 + (3-2)^2 + (2-2)^2}$
$d = \sqrt{5} < D_{\min} = 4$
i-1
i
i-2
i
d

Fig. 2. (Left) Non-balanced yard-bays. (Right) Proximity of two dangerous containers.

3.1 H_2: Balancing Contiguous Yard-Bays

As we have introduced before, in this section we present an extension for the heuristic H_1 (Algorithm 1) to include the balancing of continuous yard-bays as a requirement. It is considered that there is a *sink* when a difference higher than two containers exists between two adjacent rows in contiguous yard-bays. This heuristic is an extension of the *balanced heuristic* presented in [6], which avoids *sinks* in the same *yard-bay* (horizontal balance) both before and after the outbound containers have been removed from the yard. However in this case a sink represents a constraint between two subproblems. Thus, we also consider that there is a *sink* when a difference of two exists between the same row r in two contiguous *yard-bays* (vertical balance).

This heuristic is showed in Algorithm 3. It uses the Function *HeightsWithoutGoals* (Algorithm 2) in order to calculate for the *yard-bay* b the height for each stack where the first no-goal container is. These values are employed to get the difference of height between two adjacent stacks once the goal containers have been removed from the yard. Heights of each row are stored as soon as the planner gets the final solution plan for one *yard-bay*.

Firstly, we apply the heuristic presented in [6] on the *yard-bay* b. We obtain the differences between the row r and $r - 1$ to calculate the value of h by using the heights calculated in Algorithm 2 and the real heights of the actual *yard-bay*. When this value is zero (the *yard-bay* b is horizontally balanced), then we introduce our heuristic to balance it with respect to the last *yard-bay* b_l. To do so, we must also calculate the `heights'` through the Algorithm 2 over b_l and use the real heights of it in order to obtain the differences between the row r situated in b and b_l. When these differences are higher than 2 we increase h proportionally. After that process, b will be balanced horizontally with respect to their rows, and vertically with respect to the last *yard-bay*. Repeating this process for each *yard-bay* in the *block*, this will be completely balanced.

Algorithm 2. Function `HeightsWithoutGoals` to calculate heights of each row without taking into account the goal containers at the top

```
Data: $b$: state of the yard-bay;
Result: MinHeight, heights calculated;
for $r \leftarrow 1$ to numRows($b$) do
    MinHeight[$r, b$] = Height[$r, b$];
    // Decrease till the first no goal-container
    while MinHeight[$r, b$] $> 0 \wedge$ GoalContainer(MinHeight[$r, b$], $r$) $\in b$ do
        MinHeight[$r, b$] $- -$;
    end
end
```

3.2 H_3: Dangerous Containers

Within a *block*, there are different types of containers depending on the goods they transport, being some of them dangerous. If they do not satisfy certain restrictions, it

Algorithm 3. Pseudo-code to balance two adjacent *yard-bays*

```
Data: b: state of the actual yard-bay; h: Initial heuristic; $b_l$: last yard-bay;
Result: h: heuristic value of b
// Getting the balance horizontally
HeightsWithoutGoals($b$);
$h+ = $ BalBeforeAfter($b$);
// This heuristic will be executed after a partial solution
if $h == 0 \wedge b \neq 1$ then
    $\Delta h = 0$;
    HeightsWithoutGoals($b_l$);
    // Balancing with containers which are not objective
    for $r \leftarrow 1$ to numRows($b$) do
        $\Delta h = $ Abs(MinHeight$[r, b_l] -$ MinHeight$[r, b]$);
        if $\Delta h > 2$ then
            $h = h + (\Delta h - 2)/2$;
        end
    end
    if $h < 1$ then
        // Balancing with containers which are objective
        for $r \leftarrow 1$ to numRows($b$) do
            $\Delta h = $ Abs(Height$[r, b_l] -$ Height$[r, b]$);
            if $\Delta h > 2$ then
                $h = h + (\Delta h - 2)/2$;
            end
        end
    end
end
```

may become a hazard situation for the *yard* since e.g. if one of them explodes and they are not enough far between them, it will set off a chain of explosions.

With this added objective, the next heuristic (Algorithm 4) ensures a minimum distance (D_{min}) between every two dangerous containers (C_d) in the *yard*. D_{min} is set as one parameter for the planner and the distance is calculated as the Euclidean distance, considering each container located in a 3-dimensional space (X,Y,Z) where X is the number of yard-bays, Y is the number of rows and Z is the tier.

Generally, in container terminals, at most, there is only one dangerous container in two contiguous yard-bays, so that we take into account this assumption in the development of this heuristic.

This heuristic increase h when a dangerous container C_{d1} exists in a yard-bay b. Thereby, for each dangerous container C_{d2} allocated within a range of the last D_{min} *yard-bays* is calculated the Euclidean distance. When for one of those C_{d2} the distance is lower than D_{min} we increase h with the number of containers n on C_{d1} because we indicate that removing those n containers is necessary to move the container C_{d1}.

Algorithm 4. Pseudo-code to avoid locating two dangerous containers closer to a distance D_{min}

```
Data: $B$: whole block; $b$: state of the actual yard-bay; $h$: Initial heuristic; $D_{min}$:
      Minimum distance; $NC$: Number of containers;
Result: $h$: heuristic value of $b$;
$nBay = \texttt{NumBay}(b)$;
if $nBay > 1 \wedge h < NC \wedge \exists C_{d1} \in b$ then
    $\Delta h = 0$;
    $L_1 = \texttt{Location}(C_{d1})$;
    foreach $b_l \in Y/\texttt{NumBay}(b_l) \in \{\max(nBay - D_{min} + 1, 1), nBay - 1\}$ do
        if $\exists C_{d2} \in b_l$ then
            $L_2 = \texttt{Location}(C_{d2})$;
            $dist = \texttt{EuclideanDistance}(L_1, L_2)$;
            if $dist < D_{min}$ then
                $\Delta h = \Delta h + \texttt{NumContainersOn}(C_{d1})$;
            end
        end
    end
    $h+ = \Delta h$;
end
```

4 Evaluation

In this section, we evaluate the behaviour of the heuristics presented in this paper. The experiments were performed on random instances. A random instance of a yard-bay is characterized by the tuple $< n, s >$, where n is the number of containers in a yard-bay and s is the number of selected containers in the yard-bay. Each instance is a random configuration of all containers distributed along six stacks with 4 tiers. They are solved on a personal computer equipped with a Core 2 Quad Q9950 2.84Ghz with 3.25Gb RAM.

First, we present a comparison between our basic domain dependent heuristic H_1 against a domain independent one (Simplanner). Thus, Table 1 presents the average running time (in milliseconds) to achieve a first solution as well as the best solution found (number of reshuffles) in 10 seconds in both Simplaner and heuristic H_1 in problems $< n, 4 >$. Thus, we fixed the number of selected containers to 4 and we increased the number of containers n from 15 to 20. It can be observed that our new domain-dependent heuristic is able to find a solution in a few milliseconds, meanwhile the domain-independent heuristic (Simplaner) needs some more time for finding the first solution. Furthermore, due to the fact that our tool is an anytime planner, we evaluate the best solution found in a given time. It can be observed, in Table 1, that heuristic H_1 is able to find better solutions than Simplaner.

Table 2 shows the performance of the heuristics for solving the whole *block* of yard-bays. These experiments were performed in *blocks* of 20 *yard-bays* and each of them are instances $< 15, 4 >$. This evaluation was carried out in a yard with 3 blocks of 20 yard-bays. Thus, the results showed in Table 2 represent the average number of reshuffles, the average number of sinks generated along the block and the average number of unsatisfied dangerous containers.

Table 1. Average number of reshuffles and running time of $Simplaner$ and H_1 in problems $< n, 4 >$

Instance	**A Domain Independent Heuristic (Simplaner)**		**Our New Domain Dependent Heuristic** (H_1)	
	Running time for the first solution	Best Solution in 10 seconds	Running time for the first solution	Best Solution in 10 seconds
$< 15, 4 >$	180	2.80	6	2.55
$< 17, 4 >$	320	4.05	10	3.60
$< 19, 4 >$	533	4.70	15	4.15
$< 20, 4 >$	1210	6.11	40	5.23

In this table, it can be observed that H_1 still outperforms *Simplaner* in the average number of reshuffles. However, due to the fact that they do not take into account the balancing constraints, Simplaner generated an average of 27 sinks in the block of yard-bay and H_1 generated and average of 32.67 sinks. In both cases, the average number of unfeasible constraints for dangerous containers was 7.67.

Taking into account that H_4 is a junction of H_2 and H_3, both H_2 and H_3 solved their problems, that is, H_2 obtained its solutions with no sinks and H_3 obtained its solutions by satisfying all dangerous constraints. Furthermore, H_4 was able to solve its problems by satisfying both types of constraints. However we could state that balancing problem is harder than the problem related to dangerous containers because H_2 needs more reshuffles to obtain a solution plan than H_3. Moreover, we observe with H_2, H_3 and H_4 ensure the established requirements however the average reshuffles is increased with respect to H_1.

Table 2. Evaluation with blocks of 20 *yard-bays* each one being a $< 15, 4 >$ problem

	Simplaner	H_1	H_2	H_3	H_4
Avg. Reshuffles	3.92	3.60	5.68	4.30	6.53
Avg. Num. Sinks	27.00	32.67	0	33.33	0
Avg. Num. Dangerous	7.67	7.67	8.00	0	0

5 Conclusions

This paper presents a set of heuristics for solving the container stacking problem form the Artificial Intelligence point of view. We have developed a domain-dependent planning tool for finding an appropriate configuration of containers in a *bay*. Thus, given a set of outgoing containers, our planner minimizes the number of necessary reshuffles of containers in order to allocate all selected containers at the top of the stacks by satisfying both balancing constraints and dangerous container constraints. This approximation has also be extended to manage *blocks* of yard-bays into consideration. However, as the problems involve a larger number of constraints, the solution becomes harder and the number of reshuffles increases.

Due to the fact that a solution of a yard-bay influences on the solution of the following yard-bay, the order of solving the yard-bays will determine the minimal number of reshuffles. Thus, in further works, we will focus our attention on new manners to manage the block of yard-bays. Furthermore, we will continue on the development of more complex domain-dependent planning heuristics to manage new hard and soft constraints. An example of soft constraints in to minimize the distance travelled by the crane in reshuffling operations.

Acknowledgment

This work has been partially supported by the research projects TIN2007-67943-C02-01 (Min. de Educacion y Ciencia, Spain-FEDER), P19/08 (Min. de Fomento, Spain-FEDER).

References

1. Ghallab, M., Howe, A., Knoblock, C., McDermott, D., Ram, A., Veloso, M., Weld, D., Wilkins, D.: PDDL - the planning domain definition language. In: AIPS-98 Planning Committee (1998)
2. Hoffman, J., Nebel, B.: The FF planning system: Fast planning generation through heuristic search. Journal of Artificial Intelligence Research 14, 253–302 (2001)
3. Kim, K.W., Park, Y.M., Ryu, K.R.: Deriving decision rules to locate export containers in container yards. European Journal of Operational Research 124, 89–101 (2000)
4. Lee, Y., Hsu, N.-Y.: An optimization model for the container pre-marshalling problem. Computers & Operations Research 34(11), 3295–3313 (2007)
5. Salido, M., Sapena, O., Barber, F.: The container stacking problem: an artificial intelligence planning-based approach. In: The International Workshop on Harbour, Maritime and Multimodal Logistics Modelling and Simulation (2009)
6. Salido, M., Sapena, O., Rodriguez, M., Barber, F.: Heuristicas dependientes del dominio en terminales de contenedores. In: PSCS'09 Workshop for Planning, Scheduling and Constraint Satisfaction (2009)
7. Salido, M., Sapena, O., Rodriguez, M., Barber, F.: A planning tool for minimizing reshuffles in containers terminals. In: ICTAI 2009: 21st International Conference on Tools with Artificial Intelligence (2009)
8. Sapena, O., Onainda, E.: Domain independent on-line planning for strips domains. In: Garijo, F.J., Riquelme, J.-C., Toro, M. (eds.) IBERAMIA 2002. LNCS (LNAI), vol. 2527, pp. 825–834. Springer, Heidelberg (2002)
9. Slaney, J., Thibaux, S.: Blocks world revisited. Artificial Intelligence 125, 119–153 (2001)
10. Zilberstein, S., Russell, S.J.: Optimal composition of real-time systems. Artificial Intelligence 82(1-2), 181–213 (1996)

Robust Solutions in Changing Constraint Satisfaction Problems

Laura Climent, Miguel Á. Salido, and Federico Barber

Instituto de Automática e Informática Industrial
Universidad Politécnica de Valencia,
Valencia, Spain

Abstract. Constraint programming is a successful technology for solving combinatorial problems modeled as constraint satisfaction problems (CSPs). An important extension of constraint technology involves problems that undergo changes that may invalidate the current solution. These problems are called Dynamic Constraint Satisfaction Problems (DynCSP). Many works on dynamic problems sought methods for finding new solutions. In this paper, we focus our attention on studying the robustness of solutions in DynCSPs. Thus, most robust solutions will be able to absorb changes in the constraints of the problem. To this end, we label each constraint with two parameters that measure the degree of dynamism and the quantity of change. Thus, we randomly generate a set of more restricted CSPs by using these labels. The solutions that satisfy more random CSPs will have a higher probability of remain valid under constraints changes of the original CSP.

1 Introduction

Nowadays many real problems can be modeled as constraint satisfaction problems and are solved using constraint programming techniques. Much effort has been spent to increase the efficiency of the constraint satisfaction algorithms: filtering, learning and distributed techniques, improved backtracking, use of efficient representations and heuristics, etc. This effort resulted in the design of constraint reasoning tools which were used to solve numerous real problems. However, all these techniques assume that the set of variables and constraints which compose the CSP is completely known and fixed. This is a strong limitation when dealing with real situations where the CSP under consideration may evolve because of the environment, the user and other agents. Related to the environment is the evolution of the set of tasks to be performed and/or of their execution conditions in scheduling applications. In addition, the user can change the requirements in the framework of an interactive design. There are other agents in the framework of a distributed system responsible of the evolution of the CSP [7].

Real world is dynamic in its nature, so techniques attempting to model the real world should take this dynamism in consideration. Dynamic Constraint Satisfaction Problem [2] is an extension to a static CSP that models addition and retraction of constraints and hence it is more appropriate for handling dynamic real-world problems. It is indeed easy to see that all the possible modifications in constraints or domains of a CSP can be expressed in terms of constraint additions or removals [7].

N. García-Pedrajas et al. (Eds.): IEA/AIE 2010, Part I, LNAI 6096, pp. 752–761, 2010.

Several techniques have been proposed to solve DynCSPs, including: searching for stable solutions that are valid after small problem changes [8]; searching for a new solution that minimizes the number of changes from the original solution [5]; reusing the original solution to produce a new solution [7]; or solving DynCSPs by identifying stable features [9].

- Searching for stable solutions that are valid after small problem changes has the objective of exploring methods for finding solutions that are more likely to remain valid after changes because this fact means that these are stable solutions [8]. These changes alter temporarily the set of valid assignments, then changes may occur repeatedly, and different changes may occur with different frequencies. Nevertheless, the probabilities of change, for example the probability of value loss, are not known a priori. An assignment is evaluated in terms of the product of the probabilities associated with constraints that are violated. The strategy is to penalize values that are invalid when there are changes in the problem. In addition, this idea is incorporated into a hill-climbing procedure. When the first solution is found, a value in the deletion set was selected at random and deleted from the domain of its associated variable. If the current solution is invalidated by this deletion, then hill-climbing continues until it finds another solution.
- Searching for a new solution that minimizes the number of changes from the original solution, considers the stability like close solutions in case of changes [5]. Its aim is to be able to constrain a problem in a way that a solution has the properties of robustness that the problem requires. These models are called super models. Super solutions are useful tools for this purpose. A solution is a super solution if it is possible to repair the solution with only a few changes when a small number of variables lose their values [4]. Given a solution, a variable is repairable if there exists at least one other value for that variable compatible with the rest of the solution and with this new value all the constraints are satisfied without changing the rest of the solution. That is to say any solution of the reformulated CSP is a super solution of the original. The algorithm for finding super-solutions, based on the Maintain Arc-Consistency algorithm, returns both the super solutions and the repair solutions. Super solutions are good for characterizing robustness; nevertheless with this method we do not obtain robust solutions.
- Reusing the original solution to produce a new solution consist of producing the new solution by local changes on the previous one [7]. The main idea is that it is possible to enter a new task *t* if there exists for *t* a location such that all the tasks whose location is incompatible with the location of t can be removed and entered again one after another without modifying the location of the new task t. In order to develop this technique, we must explore the search space.
- Solving dynamic constraint satisfaction problems by identifying stable features proposes different strategies that can be classified in two main classes: Efficient methods for solving the new problem using information about the affected parts of the assignment and methods for finding stable solutions [9]. The way to find the stable solutions is using the fact that the solutions remain valid after changes or by means of a fixed number of assignment changes, being still a valid solution after a change in the initial problem.

As can be deduced from the above techniques there is not a well established definition of robustness and stability of solutions in a CSP. Some researchers deal about stability and others about robustness. What's the difference between stable and robust? It's the first question that comes to mind, especially for researchers who work with quantitative models or mathematical theories [6]. In general, a solution is stable in a dynamic system, whether through small perturbations to the solution, we obtain a new solution that is similar to the original one. The perturbations are small differences in the actual state of the system. However the robustness concept is broader than stability concept. Robustness is a measure of feature persistence in systems that compels us to focus on perturbations due to they represent changes in the composition or topology of the system.

By taken into account the general definitions of stability and robustness, we can specify in the definition of the robustness of a solution in a CSP: A solution of a CSP is robust if it is able to remain valid after small modifications of the problems. These modifications in the CSP can be expressed in terms of constraint additions or removals [7]. Thus, in this paper we focus our attention on searching for robust solutions that remain valid after problem changes. In many real life problems many constraints are considered dynamic meanwhile others are considered statics. We assume that the user knows the percentage of dynamism and the quantity of change of the constraints. Thus, each constraint will be labeled in order to study the robustness of the solutions of the original problem. We will generate a set of random and restricted CSPs in order to analyze the robustness of the original solutions. Thus, solutions able to persist more modifications in the original problem will be more robust.

1.1 Definitions

By following standard notations and definitions in the literature [1] we have summarized the basic definitions that we will use in the rest of the paper.

Definition 1. Constraint Satisfaction Problem (CSP) is a triple $P = \langle X, D, C \rangle$ where, X is the finite set of variables $\{x_1, x_2, ..., x_n\}$. D is a set of domains $D = D_1$xD_2x...xD_n such that for each variable $x_i \in X$ there is a finite set of values that variable can take. C is a finite set of constraints $C = \{C_1, C_2, ..., C_m\}$ which restrict the values that the variables can simultaneously take.

Definition 2. Dynamic Constraint: is a constraint that undergo changes. We define two parameters to fix the dynamism of a constraint: l_i and d_i. The first one measures the dynamism likelihood and the second the magnitude of change.

Definition 3. Stability of a solution is the ability of a solution to share as many values as possible with a new solution if a change occurs [4].

Definition 4. Robustness of a solution is the measure of persistence of the solution after modifications in the original CSP. Thus a solution of a CSP is robust if it is able to remain valid after small modifications of the problem.

2 Studying Robustness of Solutions

Our aim is to find solutions to problems that remain valid under changes produced in the constraints. Many real problems, such as planning and scheduling problems, can be modeled as a CSP and the constraints are established knowing that they are probably to be modified. For instance, in scheduling problems, if a worker starts working at time $T1_{begin}$ and his first task must be finished before 50 minutes, then the constraint is $(T1_{end} - T1_{begin} < 50)$. However, the worker could arrive late to work due to many factors. Thus, we can assign a high probability of change to this constraint and make a simulation of a possible change of the constraint. For example, the modification in the constraint could be $(T1_{end} - T1_{begin} < 40)$. If something is wrong the task can be delayed some more minutes and the obtained solution for the modified problem will be more robust than a solution of the original problem due to the fact that if the worker arrives late ($<$10 minutes) the solution remains valid.

In order to model the dynamism of the constraints, we define two parameters:

- l_i: This parameter represents the dynamism likelihood. That is, this parameter measures the probability of change of the constraints. Its value in ranged between zero and one. The minimum value ($l_i = 0$) means that the constraint is static so there is no probability that it changes (no dynamic constraint). The maximum values ($l_i \approx 1$) mean that the constraint is very dynamic so the probability that it changes is very high.
- d_i: This metrics indicates or represents the amount of change in one constraint. Without loss of generality we consider four integer values that this parameter can take: 1, 2, 3 and 4. The number 1 means that the constraint would change in a small quantity and the number 4 means that the constraint would change in a large quantity.

The change of the constraints from the original ones is always produced in a more restrictive way, because only more restrictive constraints can invalidate the solutions. That is why only this kind of changes is interesting in the analysis of the robustness.

2.1 Restricting Constraints

In order to find robust solutions by restricting constraints, each constraint is labeled with the two above parameters l_i and d_i. We assume constraints in the form $C_i : Ax - k \geq 0 \rightarrow a_1x_1 + a_2x_2 + ... + a_nx_n - k \geq 0$. We carry out a given number of simulations n and for each simulation we generate different random probabilities p_i. Thus, if the probability of change l_i is higher than p_i, a new constraint is generated (see equation (1)), else the constraint C_i does not change. Carrying out this process n new $CSP's$ will be generated. After solving the initial CSP, a set of solutions are obtained. Then, each solution is checked to analyzed if it is feasible for the new $CSP's$. The solution with the highest score will be the most robust solution for the original CSP.

$$C_i' : Ax - (k + (d_i * k/4)) \geq 0 \tag{1}$$

Max-Robust solver Input: A CSP(P), $P = \langle X, D, C\rangle$: $C = \{C_1, C_2, ..., C_m\}$, a number of simulations n, a set of labels $L = \{l_1, l_2, ..., l_m\}$ and a set of labels $D = \{d_1, d_2, ..., d_m\}$.

Output: solutions Sol and its level of robustness

```
$S_i \leftarrow$ Solve CSP(P)
$label_i = 0$
for every simulation $n$
    for every constraint $C_i$ and labels $l_i$ and $d_i$
        Generate a random probability $p_i$
        if $(l_i \geq p_i)$
            Generate a new constraint $C'_i$ based on $d_i$
        else
            $C'_i = C_i$
    endfor
    $C'_i = \{C'_1, C'_2, ..., C'_m\}$
    Compose a $CSP'(P')$ $P' = \langle X, D, C'\rangle$
    for every solution $Sol_i \in S_i$
        if $(Sol_i$ Satisfies $CSP'(P'))$
            $label_i = label_i + 1;$
        endif
    endfor
endfor
return $Sol_i$, $label_i$
end
```

3 Example

Following we present an example to clarify the robustness method presented in the previous section. For simplicity, we consider a CSP with only two variables x_1 and x_2 with domains $D_1 : \{3..7\}$ and $D_2 : \{2..6\}$ (see Figure 1). The constraints with the dynamism parameters are:

$$C_1(0.2, 2) : x_1 + x_2 \geq 6$$
$$C_2(0.8, 1) : x_1 + x_2 \leq 12$$
$$C_3(0.4, 4) : x_1 - x_2 \leq 4$$
$$C_4(0.2, 4) : -x_1 + x_2 \leq 2$$

The first number between the parenthesis represents the dynamism likelihood l_i and the second the quantity of change d_i.

Just us it is showed up here, in the example we use linear constraints. Nevertheless all kind of convex constraints can be used.

Figure 1 left shows a graphic representation of the original problem. It can be observed that the Cartesian Product of the domains generates 25 different instantiations. The original constraints removes the non-valid instantiations $(x_1 = 3, x_2 = 2)$, $(x_1 = 7, x_2 = 2)$, $(x_1 = 3, x_2 = 6)$,$(x_1 = 7, x_2 = 6)$. Thus, the rest of 21 instantiations are solutions of the CSP. If a CSP solver gives us a subset of solutions, which one is more robust? To answer this question, we will make use of the constraint labels.

In Figure 1 (left) we present the four constraints with its corresponding labels: C_1 $(0.2, 4), C_2(0.8, 3), C_3(0.4, 2), C_4(0.2, 4)$. For example, if we consider one simulation with a random probabilities of $p_1 = 0.67$, $p_2 = 0.25$, $p_3 = 0.32$ and $p_4 = 0.49$, the resultant new constraints are:

$C_1(0.2, 2) : x_1 + x_2 \geq 6 \rightarrow C_1' : x_1 + x_2 \geq 6$,
$C_2(0.8, 1) : x_1 + x_2 \leq 12 \rightarrow C_2' : x_1 + x_2 \leq 9$, (see equation (1))
$C_3(0.4, 4) : -x_1 + x_2 \leq 2 \rightarrow C_3' : x_1 + x_2 \leq 0$, (see equation (1))
$C_4(0.2, 4) : x_1 - x_2 \leq 4 \rightarrow C_4' : x_1 - x_2 \leq 4$.

The resultant $CSP'(P')$ is showed in Figure 1 (right). It can be observed that for this simulation with these random probabilities, the constraints that change are C_2 and C_3 due to their parameters l_i are higher to their corresponding random likelihood for this simulation. This means that they are dynamic constraints and a tightest CSP is possible. However, the constraints C_1 and C_4 do not change due to their parameters l_i are lower to their corresponding random likelihood.

The new constraints remove some unfeasible instantiations, $(x_1 = 7, x_2 = 5)$, $(x_1 = 7, x_2 = 4)$, $(x_1 = 7, x_2 = 3)$, $(x_1 = 6, x_2 = 6)$,$(x_1 = 6, x_2 = 5)$, $(x_1 = 6, x_2 = 4)$, $(x_1 = 5, x_2 = 6)$, $(x_1 = 5, x_2 = 5)$, $(x_1 = 4, x_2 = 6)$, $(x_1 = 4, x_2 = 5)$, $(x_1 = 3, x_2 = 5)$ and $(x_1 = 3, x_2 = 4)$. These instantiations that are solutions for the original CSP are candidates that remain invalid if constraints C_2 and C_3 change. As can be seen in Figure 1 (right), the solution space has been reduced and the set of solutions $S' \subset S$ can be considered more robust than the set of solutions $S \backslash S'$.

The main idea is to produce a high number of simulations with different random probabilities p_i, counting how many $CSPs'$ each original solution satisfies. The original solution that satisfies more restricted $CSPs'$ will be more robust for the original problem.

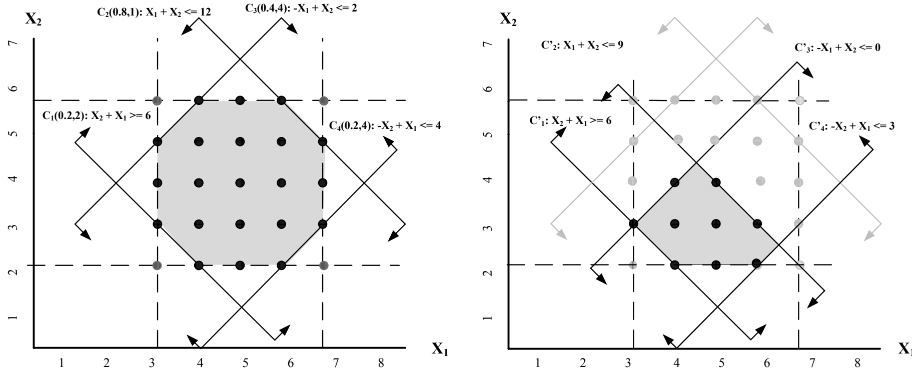


Fig. 1. Example of CSP (left). Resultant CSP when constraints are modified by its labels (right).

4 A Tool for Finding Robust Solutions

As we have pointed out in previous sections, our aim is to find robust solutions by restricting a CSP with different simulations. To this end, our tool is able to generate random CSP or to load a custom CSP previously created by the user. In any case the CSP (variables, domains and constraints) is showed (see Figure 2). Then, the user selects an upper bound of desired solutions for the original CSP. These solutions can be selected from a sample of a larger number of solutions. Then, the user selects the number of simulations and the tool obtains the robustness of each original solution.

Let's analyze the main characteristics of the developed tool. The interface of the tool is divided in five basic parts and two windows to visualize the original CSP and the obtained results (see Figure 2).

The first and second part is aimed to generate/load a CSP to solve it. The first part is designed to generate random $CSPs$. The user introduces the parameters of the CSP: number of variables, domain size, number of constraints and the constrainedness parameter. The last parameter fixes how much restrictive the random constraints will be. That parameter is very useful because if we generate a CSP which does not have a solution, we can relax the parameter in order to obtain less restrictive constraints for the problem. Pressing *RandomCSP* button, a CSP based on the above parameters is generated. Furthermore, pressing *SaveCSP* button, it is possible to save the random generated problem in a file.

The second part is related to load a problem. If the user desires loading an existing CSP, then the user presses the *LoadCSP* button and select the file to load. Then, the loaded CSP is visualized in the window.

The third part is composed of the elements responsible to solve the CSP. The user introduces an upper bound of desired solutions to obtain. In addition, he can select a number of solutions which are going to be analyzed and showed in the results window. This set of solutions is achieved doing a sampling in all the solutions obtained.

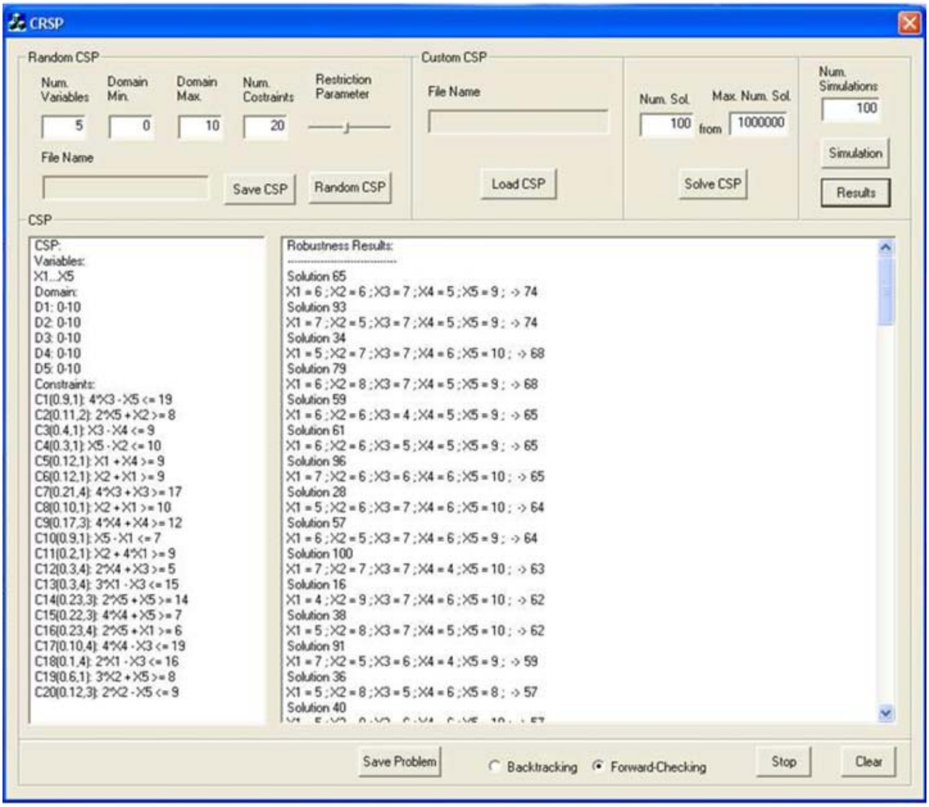
The fourth part is the responsible of the simulation of the new $CSPs'$ based on the original one. First of all the user specifies the number of simulations. Once the user presses the *Simulation* button, the new $CSPs'$ are generated and solved, and a robustness analysis is carried out. Then, the results are showed in the windows, that is, each original solution is labeled with a number that measures how many $CSPs'$ this solution satisfies. Pressing *Results* button, the solutions are arranged, showing the most robust solutions in the first place.

The fifth part, placed on the bottom of the window, has a button called *Save Problem* and allow to save the problem and the robustness results obtained. In addition the user can choose between backtracking and forward-checking search methods for solving the CSPs.

5 Evaluation

In this section we compare the robustness of a solution obtained by our algorithm against a solution obtained by a general CSP solver. To this end, we have randomly generated 50 random instances of problems characterized by the 3-tuple $< n, d, m >$, where n was the number of variables, d the domain size and m the number of binary constraints. For each instance the algorithm performed 100 simulations.

It can be observed in Table 1 that the average number of simulations satisfied by our algorithm (Max-Robust solver) is always higher than the satisfied by a general CSP solver. That means that the solutions provided by our algorithm are always more robust



Num. Variables: 5
Domain Min: 0
Domain Max: 10
Num. Costraints: 20
File Name:
File Name:
Num. Sol: 100 from Max. Num. Sol: 1000000
Num. Simulations: 100
○ Backtracking ● Forward-Checking

```
CSP:
Variables:
X1...X5
Domain:
D1: 0-10
D2: 0-10
D3: 0-10
D4: 0-10
D5: 0-10
Constraints:
C1(0.9,1): 4*X3 - X5 <= 19
C2(0.11,2): 2*X5 + X2 >= 8
C3(0.4,1): X3 - X4 <= 9
C4(0.3,1): X5 - X2 <= 10
C5(0.12,1): X1 + X4 >= 9
C6(0.12,1): X2 + X1 >= 9
C7(0.21,4): 4*X3 + X3 >= 17
C8(0.10,1): X2 + X1 >= 10
C9(0.17,3): 4*X4 + X4 >= 12
C10(0.9,1): X5 - X1 <= 7
C11(0.2,1): X2 + 4*X1 >= 9
C12(0.3,4): 2*X4 + X3 >= 5
C13(0.3,4): 3*X1 - X3 <= 15
C14(0.23,3): 2*X5 + X5 >= 14
C15(0.22,3): 4*X4 + X5 >= 7
C16(0.23,4): 2*X5 + X1 >= 6
C17(0.10,4): 4*X4 - X3 <= 19
C18(0.1,4): 2*X1 - X3 <= 16
C19(0.6,1): 3*X2 + X5 >= 8
C20(0.12,3): 2*X2 - X5 <= 9
```

```
Robustness Results:
------------------------------
Solution 65
X1 = 6 ;X2 = 6 ;X3 = 7 ;X4 = 5 ;X5 = 9 ; -> 74
Solution 93
X1 = 7 ;X2 = 5 ;X3 = 7 ;X4 = 5 ;X5 = 9 ; -> 74
Solution 34
X1 = 5 ;X2 = 7 ;X3 = 7 ;X4 = 6 ;X5 = 10 ; -> 68
Solution 79
X1 = 6 ;X2 = 8 ;X3 = 7 ;X4 = 5 ;X5 = 9 ; -> 68
Solution 59
X1 = 6 ;X2 = 6 ;X3 = 4 ;X4 = 5 ;X5 = 9 ; -> 65
Solution 61
X1 = 6 ;X2 = 6 ;X3 = 5 ;X4 = 5 ;X5 = 9 ; -> 65
Solution 96
X1 = 7 ;X2 = 6 ;X3 = 6 ;X4 = 6 ;X5 = 10 ; -> 65
Solution 28
X1 = 5 ;X2 = 6 ;X3 = 7 ;X4 = 6 ;X5 = 10 ; -> 64
Solution 57
X1 = 6 ;X2 = 5 ;X3 = 7 ;X4 = 6 ;X5 = 9 ; -> 64
Solution 100
X1 = 7 ;X2 = 7 ;X3 = 7 ;X4 = 4 ;X5 = 10 ; -> 63
Solution 16
X1 = 4 ;X2 = 9 ;X3 = 7 ;X4 = 6 ;X5 = 10 ; -> 62
Solution 38
X1 = 5 ;X2 = 8 ;X3 = 7 ;X4 = 5 ;X5 = 10 ; -> 62
Solution 91
X1 = 7 ;X2 = 5 ;X3 = 6 ;X4 = 4 ;X5 = 9 ; -> 59
Solution 36
X1 = 5 ;X2 = 8 ;X3 = 5 ;X4 = 6 ;X5 = 8 ; -> 57
Solution 40
```

Fig. 2. Interface of the tool

than the others ones due to the fact that they are able to satisfy more random problems. Focusing on the dynamism parameters, it can be observed that for low probabilities of change (l) and low magnitudes of change (d), the founded solutions have a high percentage of robustness due to the fact that this kind of CSPs are considered stable. However, if one of the dynamism parameters increases, the robustness percentages decrease, being the lowest when both parameters of dynamism are very high.

6 Some Issues to Improve the Search of Robust Solutions

For all the simulations, once the solution space has been reduced, we can also carry out different techniques to obtain even more robust solutions. Due to the fact that the set of reduced CSPs ($CSP's$) have tighter constraints, filtering techniques such as arc-consistency can reduce the domain size of the variables. Furthermore, we can apply some strategies to find even more robust solutions:

- Carrying out a value ordering. Once filtering techniques such as node-consistency, arc-consistency are carried out and the domains have been reduced by the previous

Table 1. Influence of the dynamism parameters in the solution robustness

		$< 5, 5, 10 >$		$< 10, 5, 20 >$	
		$d < 3$	$d \geq 3$	$d < 3$	$d \geq 3$
$l \leq 0.2$	CSP-solver	72%	53%	48%	28%
	Max-Robust solver	100%	94%	77%	49%
$l \geq 0.5$	CSP-solver	10%	2%	2%	0%
	Max-Robust solver	98%	78%	25%	2%

techniques, a value ordering can be applied to starting the search by the center of the search space. Thus, given a domain $[d_i, D_i]$ of a variable x_i, the search starts by assigning the value $x_i = \frac{d_i+D_i}{2}$, that is, by the center of the interval. For instance, in the example of Figure 1, the search will start by assigning the value $(x_1 = 5, x_2 = 4)$.

- Modeling the problem as a MAX-CSP with the aim of maximizing the number of satisfied constraints. Thus, the original constraints are considered hard constraints meanwhile restricted constraints generated from each original constraint are considered soft constraints. The main goal is to obtain a solutions that maximize the number of soft constraints. If a solution satisfies all hard and soft constraints, then it is considered the most robust solution. Thus, we find for robust solutions in real problems such as planning and scheduling problems [3].

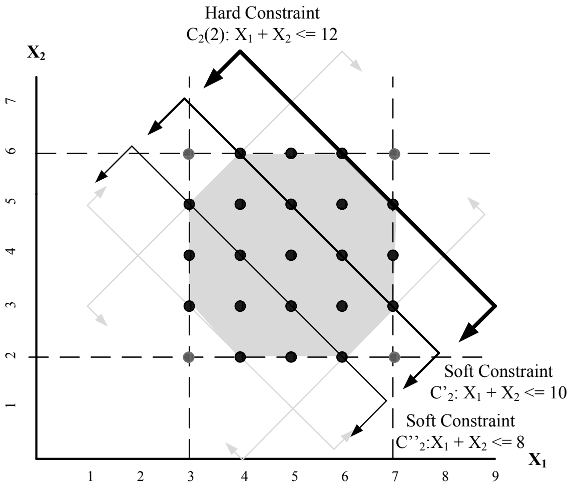


Fig. 3. Hard and soft constraints in a MAX-CSP

7 Conclusions

Many real problems can be modeled as Dynamic Constraint Satisfaction Problems. Several techniques have been developed to manage Dynamic CSPs. Some of them are based on searching for robust solutions that are valid after small problem changes. However, it is a very hard task due to the fact that all solutions must be found to obtain the most robust one based on an objective function. Other techniques are based on searching for a new solution that minimizes the number of changes from the original solution. These techniques are appropriate for dynamic CSPs but the obtained solutions are not robust. Finally some techniques are based on reusing the original solution to search a new solution. In this case, a CSP solver must explore the search space for finding a solution for the new problem.

In this paper, we focus our attention on finding robust solutions by restricting the original CSP, with the aim of obtaining solutions able to absorb changes. To this end, we have developed a tool for simulating a set of $CSPs'$ with tighter constraints, depending on the dynamism parameters associated to the original constraints. Thus, solutions which are valid in most $CSPs'$, are considered more robust solutions for the original CSP. These solutions have a high probability to remain valid after constraints changes in the original CSP.

Acknowledgment

This work is partially supported by the research projects TIN2007-67943-C02-01 (Min. de Educacion y Ciencia, Spain-FEDER), P19/08 (Min. de Fomento, Spain-FEDER), ACOMP/2009/178 Generalitat Valenciana, and by the Technical University of Valencia.

References

1. Bessiere, C.: Constraint propagation. Technical report, CNRS/University of Montpellier (2006)
2. Dechter, R., Dechter, A.: Dynamic constraint networks. In: Proc. of the 7th National Conference on Artificial Intelligence (AAAI-88), pp. 37–42 (1988)
3. Garrido, A., Salido, M., Barber, F.: Heuristic methods for solving job-shop scheduling problems. In: ECAI-2000 Workshop on New Results in Planning, Scheduling and Design (PUK), pp. 36–43 (2000)
4. Hebrand, E., Hnich, B., Walsh, T.: Super CSPs. Technical report (2003)
5. Hebrard, E., Hnich, B., Walsh, T.: Super solutions in constraint programming. In: Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems. CPAIOR-04, pp. 157–172 (2004)
6. Jen, E.: Stable or robust? what's the difference? Complexity 8, 12–18 (2003)
7. Verfaillie, G., Schiex, T.: Solution reuse in dynamic constraint satisfaction problems. In: Proceedings of the 12th National Conference on Artificial Intelligence (AAAI-94), pp. 307–312 (1994)
8. Wallace, R., Freuder, E.: Stable solutions for dynamic constraint satisfaction problems. In: Maher, M.J., Puget, J.-F. (eds.) CP 1998. LNCS, vol. 1520, pp. 447–461. Springer, Heidelberg (1998)
9. Wallace, R., Grimes, D., Freuder, E.: Solving dynamic constraint satisfaction problems by identifying stable features. In: International Joint Conferences on Artificial Intelligence, pp. 621–627 (2009)

Author Index