Studies in Fuzziness and Soft Computing

Mo Jamshidi
Vladik Kreinovich
Janusz Kacprzyk *Editors*

Advance Trends in Soft Computing

Proceedings of WCSC 2013, December 16–18, San Antonio, Texas, USA

Springer

Studies in Fuzziness and Soft Computing

Volume 312

Series Editors

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

For further volumes:
http://www.springer.com/series/2941

About this Series

The series "Studies in Fuzziness and Soft Computing" contains publications on various topics in the area of soft computing, which include fuzzy sets, rough sets, neural networks, evolutionary computation, probabilistic and evidential reasoning, multivalued logic, and related fields. The publications within "Studies in Fuzziness and Soft Computing" are primarily monographs and edited volumes. They cover significant recent developments in the field, both of a foundational and applicable character. An important feature of the series is its short publication time and world-wide distribution. This permits a rapid and broad dissemination of research results.

Mo Jamshidi · Vladik Kreinovich
Janusz Kacprzyk
Editors

Advance Trends in Soft Computing

Proceedings of WCSC 2013,
December 16–18, San Antonio, Texas, USA

Springer

Editors
Mo Jamshidi
The University of Texas
San Antonio, Texas
USA

Janusz Kacprzyk
Polish Academy of Sciences
Systems Research Institute
Warsaw
Poland

Vladik Kreinovich
Department of Computer Science
University of Texas at El Paso
USA

ISSN 1434-9922 ISSN 1860-0808 (electronic)
ISBN 978-3-319-03673-1 ISBN 978-3-319-03674-8 (eBook)
DOI 10.1007/978-3-319-03674-8
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2013954671

Printed on acid-free paper

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

Dedication

Monarch Trophy Studio

ARTWORK APPROVAL FORM

STOP

Please review the attached artwork layout for
CORRECT LOGOS / DATES / SPELLING (Names) Etc.

STOP

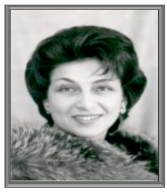
YOUR ORDER IS ON HOLD UNTIL WE RECEIVE YOUR RESPONSE

____ Artwork Approved ____ Make Revisions Indicated & Proceed With Order ____ Make Revisions, Send New Proof
MULTIPLE PROOFS MAY INCUR ADDITIONAL CHARGES

Signature _____ Date _____

PLEASE PRINT, SIGN and FAX TO (210) 341-4906

INVOICE 333185



WCSC 2013

THIRD ANNUAL WORLD CONFERENCE ON
SOFT COMPUTING
SAN ANTONIO, TEXAS, USA
DECEMBER 16-18, 2013

WCSC 2013 is Dedicated to the Scientific Heritage of
Professor Lotfi A. Zadeh, "Father of Fuzzy Logic" and
Creator of Soft Computing Methodologies, and to
Mrs. Fay Zadeh for Her Life-Changing Sacrifices and Dedications
to the Cause of Lotfi Zadeh

Tuesday December 17, 2013 San Antonio, Texas, USA

MO JAMSHIDI
GENERAL CHAIR

VLADIK KREINOVITCH
PROGRAM CHAIR

Preface

This volume of Springer-Verlag series on *Studies in Fuzziness and Soft Computing* represents the proceedings of World Conference on Soft Computing, held from December 16–18, 2013 in San Antonio, Texas, USA. This conference was jointly dedicated to:

Professor **Lotfi A. Zadeh**, "Father of Fuzzy Logic" and creator of Soft Computing Methodologies, and to Mrs. **Fay Zadeh** for her life-long sacrifices and dedications to the causes of Lotfi Zadeh and his career.

We welcomed over 80 attendees from 18 nations (Australia, Azerbaijan, Belgium, Canada, Colombia, Finland, Hungary, India, Iran, Israel, Japan, Mexico, Poland, Russia, Spain, Turkey, Ukraine, USA) who contributed to this conference.

Eight keynote speakers presented recent advances in soft computing and cloud computing, as follows:

1) Prof. Ron Yager, Iona College, USA under the title of "Intelligent Aggregation Methods for Decision Making and Learning".

2) Dr. Leila Meshkat, NASA/CALTECH JPL, USA under the title of "Uncertainty Management for NASA Space Mission".

3) Prof. Saeid Nahavandi, Deakin University, Australia under the title of "Knowledge Management in Future Factories".

4) Prof. Jerry Mendel, USC, USA, under the title of "General Type-2 Fuzzy Sets and Systems: Where Are We Now and Where Are We Heading?"

5) Mr. Eli Karpilovski, Millanox Corp., USA, under the title of "High Performance Cloud".

6) Prof. Lotfi A. Zadeh, University of California at Berkeley, USA under the title of "Bayesianism versus Fuzzy Logic".

7) Prof. Rafik Aiev, Oil Academy, Azerbaijan, under the title of "Combined States-Based Decision Theory with Z-restriction,".

We wish to thank our co-editor Professor Janusz Kacprzyk of Polish Academy of Sciences, Springer-Verlag Series Editor of this volume for facilitating and recommending to print this volume. We wish to thank many members of the organizing and program committee who helped us with the conference. We specially thank

Dr. Leontina Di Cecco of Springer-Verlag Publishers, Candice Contreras of the Electrical and Computer Engineering Department at the University of Texas at San Antonio (UTSA), as well as Ramin and Amin Sahba, Patrick Benavidez, Azima Motaghi, Maryam Ezell, Halid Kaplan and Yunus Yetis of ACE Laboratory, UTSA for helping with both the conference and the volume that will be published based on the Conference.

We wish to thank all the authors for contributing to his volume.

October 15 2013

Mo Jamshidi
San Antonio, TX, USA

Vladik Kreinovich
El Paso, TX, USA

Janusz Kacprzyk
Warsaw, Poland

Contents

Synthesis and Research of Neuro-Fuzzy Model of Ecopyrogenesis Multi-circuit Circulatory System

Yuriy P. Kondratenko[1,2], Oleksiy V. Kozlov[2],
Leonid P. Klymenko[1], and Galyna V. Kondratenko[1,2]

[1] Intelligent Information Systems Department, Ecology Department
Petro Mohyla Black Sea State University, 68-th Desantnykiv str. 10,
54003 Mykolaiv, Ukraine
y_kondrat2002@yahoo.com, rector@chdu.edu.ua,
galvlad09@rambler.ru
[2] Computerized Control Systems Department
Admiral Makarov National University of Shipbuilding, Geroiv Stalingrada ave. 9,
54025 Mykolaiv, Ukraine
oleksiy.kozlov@nuos.edu.ua

Abstract. This paper presents the development of the neuro-fuzzy mathematical model of the ecopyrogenesis (EPG) complex multiloop circulatory system (MCS). The synthesis procedure of the neuro-fuzzy model, including its adaptive-network-based fuzzy inference system for temperature calculating (ANFISTC) training particularities with input variables membership functions of different types is presented. The analysis of computer simulation results in the form of static and dynamic characteristics graphs of the MCS as a temperature control object confirms the high adequacy of the developed model to the real processes. The developed neuro-fuzzy mathematical model gives the opportunity to investigate the behavior of the temperature control object in steady and transient modes, in particular, to synthesize and adjust the temperature controller of the MCS temperature automatic control system (ACS).

Keywords: ecopyrogenesis complex, multiloop circulatory system, neuro-fuzzy mathematical model, adaptive-network-based fuzzy inference system, membership functions.

1 Introduction

The problem of industrial and domestic organic waste recycling is one of the main environmental problems caused by the development and growth of urbanization in many countries of the world. Quite a prospective method of this problem solution is the use of ecopyrogenesis technology, which allows complete utilization of the whole scope of the organic part of solid waste and low-grade coal in the environmentally-friendly and energy-saving modes [1]. EPG technology provides a simplified sorting of organic solid waste into two categories: the first category - dried organic waste, which includes all of the polymer waste, including polyvinylchloride (PVC) but not

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_1,

more than 2%, worn tires, rubber, oil sludge, paper, etc.; the second category - waste with high humidity, which include food waste, shredded wood, paper, cardboard, etc.. The first category of waste is disposed by a multi-loop circulatory pyrolysis (MCP) to obtain from the mass of raw materials up to 60-85% of the liquid fuel of light fractions with characteristics comparable to diesel fuel. The second category of waste is utilized by the method of multi-loop bizonal circulatory gasification (MCG) with obtaining generator gas, which has a calorific value 1100-1250 kcal/m^3 [1]. For realization of the EPG technology specific technological complexes are used, which are, in turn, complicated multi-component technical objects. Automation of such technological complexes allows to significantly increase the operation efficiency and economic parameters.

Stabilization of the set temperature value on the outlet of the MCS is one of the important tasks of automatic control of the EPG process [1]. The possibility of temperature control with high quality indicators allows controlling of the thermal destruction process in terms of various depth of hydrocarbon decay starting with petrol and up to diesel fuel. This allows obtaining of high-quality liquid fractions of alternative fuel on the outlet with the set molecular mass and, in turn, requires a special temperature ACS.

To study the ACS effectiveness at the stage of its design it is reasonable to use the mathematical and computer modeling methods that are quite effective and low-cost, comparing with experimental and other approaches, especially while studying the behavior of thermal power objects and their control systems [2-5]. In particular, development and adjustment of MCS ACS temperature controller requires an availability of an adequate mathematical model. Also MCS temperature ACS quality indicators significantly depend on the accuracy of the synthesized model and its adequacy to the real processes.

Therefore, the aim of this work is development and research of the mathematical model of the EPG complex MCS as a temperature control object.

2 MCS Temperature Control System and Neuro-Fuzzy Mathematical Model Structure

The principle diagram of the output (control) point of the temperature control system of the of the EPG MCS is shown in Fig. 1 [1], where the following indications are used: CB – control block; TS – temperature sensor; LFR – linear flow regulator; OC – output condenser; AF – air fan; SD – servodrive; V – valve; CA – cooling air; CW – cooling water; 1C, 2C, 3C – first, second and third MCS cooling circuits; OW – organic waste; GB – gas burner which heats the reactor; GT – gas tank with liquefied gas.

The MCS task is to cool the gas-vapor mixture, obtained in the process of waste decay in the reactor, to the set temperature in its output point. The MCS system includes three sequentially connected circuits with various cooling types: the 1st – with non-regulated air cooling; the 2nd – with regulated air cooling; the 3rd – with non-regulated water cooling. Thus, the temperature control in the output point on the MCS outlet can be performed due to the flow change of cooling air of the 2nd MCS circuit.

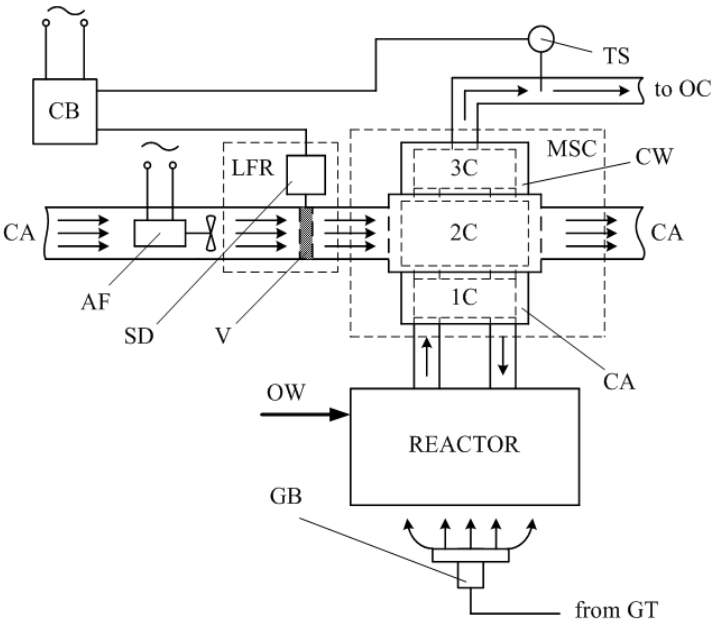


Fig. 1. Principle diagram of the output (control) point of the temperature control system of the of the EPG MCS

The unregulated air fan delivers the constant value of cooling air through the 2nd circuit, which then can be changed using a linear air flow regulator. The air LFR, in turn, is a valve with a servodrive and has the linear characteristic of the dependence of the air consumption from the input voltage. In accordance with it the servodrive rotates the valve to a certain angle, thus changing the consumption of the cooling air.

Control unit contains the set device (SD), the summator, the temperature controller and allows to control the temperature of the MCS output point both in manual and in automatic modes [1].

As already mentioned above, temperature controller synthesis requires an availability of a high-precision MCS mathematical model. The analysis of physical properties and technical characteristics of the thermal power objects as the complicated control objects with significant uncertainties and nonlinearities [5-8] shows reasonability of its mathematical model development on the basis of soft computing and artificial intelligence principles and algorithms [9-13]. The mathematical models and control systems based on fuzzy logic, artificial neural networks etc. are developed and successfully introduced in the following fields: technological processes control, transport control, medical and technical diagnostics, financial management, stock forecast, pattern recognition, etc. [14-19]. Especially effective are neuro-fuzzy hybrid mathematical models and control systems that combine the advantages of fuzzy and neural networks based systems [20-22]. Thus, to consider special features of the MCS multicircuit structure, it is reasonable to develop MCS mathematical model on the basis of the adaptive-network-based fuzzy inference system for temperature calculating and approach, given in [23]. The MCS neuro-fuzzy mathematical model functional structure is presented Fig. 2.

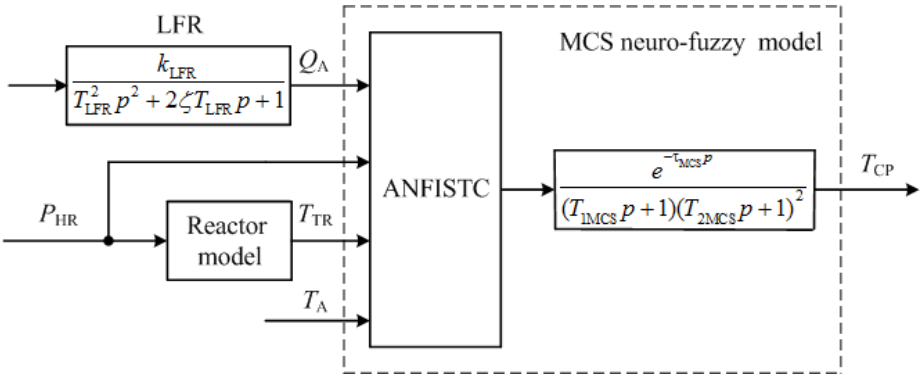


Fig. 2. Functional structure of the MCS neuro-fuzzy mathematical model

The following values are given to the input of this model: power value of the reactor heating installation P_{HR}, waste temperature value at the top of the reactor $T_{TR,}$ the flow of the cooling air of the 2nd circuit of the MCS Q_A and ambient temperature T_A. The control point temperature value of the MCS T_{CP} is formed on the output. The ANFISTC implements the dependence $T_{CP} = f_{ANFISTC}$ (P_{HR}, T_{TR}, Q_A, T_A). Time constants T_{1MCS}, T_{2MCS}, τ_{MCS} and n order are determined from the experimental characteristic of the MCS control point heating transient process using the approach given in [23]. The well tested model developed and presented in [24] was chosen as the mathematical model of the reactor. In turn, the LFR mathematical model represented as an oscillatory link, where u_{LFR} – LFR control input, k_{LFR}, T_{LFR}, ζ – gain constant, time constant and damping coefficient, which are determined by the parameters of the servodrive and gas valve, which are the parts of the linear flow regulator.

Let us consider the synthesis procedure particularities of the adaptive-network-based fuzzy inference system for temperature calculating in detail.

3 Synthesis Procedure of the ANFISTC

Adaptive-network-based fuzzy inference system (ANFIS) is a variant of hybrid neuro-fuzzy networks – neural network of direct signal propagation of particular type [20]. ANFIS implements the Sugeno type fuzzy inference system in the form of a five-layer neural network of the signal forward propagation. The neuro-fuzzy network architecture is isomorphic to the fuzzy knowledge base. In the neuro-fuzzy networks the differentiated implementation of triangular norms (multiplication and probabilistic OR) are used, as well as the smooth membership functions. This allows you to apply for adjustment of neuro-fuzzy networks fast algorithms for neural networks training, based on the method of back-propagation.

Fuzzy rule with serial number r has the following form

$$\begin{array}{l} \text{If } x_1 = a_{1,r} \text{ and} \ldots \text{and } x_n = a_{n,r} \\ \text{then } y = b_{0,r} + b_{1,r} x_1 + \ldots + b_{n,r} x_n \end{array}, \quad (1)$$

where $r = 1,\ldots, m$ – the number of rules; $a_{i,r}$ – fuzzy term with membership function $\mu_r(x_i)$, that is used for the linguistic evaluation of variable x_i in the r-th rule ($r = 1,\ldots, m$; $i = 1,\ldots, n$); $b_{q,r}$ – real numbers in conclusion of r-th rule ($r = 1,\ldots, m$; $q = 1,\ldots, n$).

The functional structure of typical ANFIS with two inputs x_1, x_2 and one output y is presented in Fig 3.

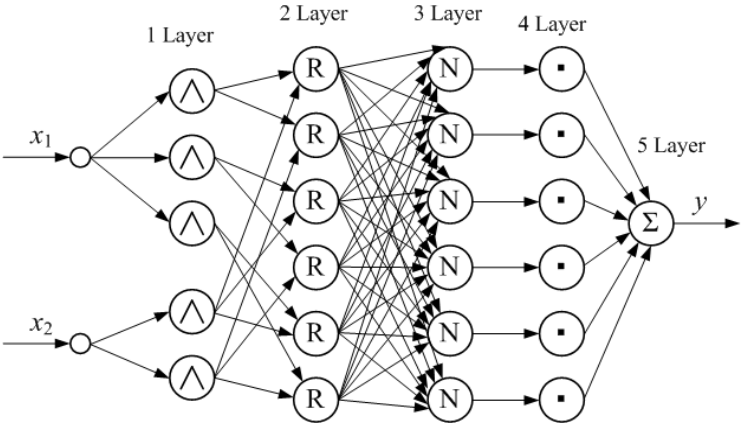


Fig. 3. Functional structure of typical ANFIS

ANFIS-network functions as follows:

Layer 1. Each node of the first layer is one term with certain membership function. Network inputs x_1, x_2,…, x_n are connected only with their terms. Nodes amount is equal to the sum of the terms of all variables. The node output is a membership degree of the input variable value to the corresponding fuzzy term

$$\mu_r(x_i) = \frac{1}{1+\left|\frac{x_i - g}{c}\right|^{2d}}, \tag{2}$$

where c, d, g – adjustable parameters of the membership function.

Layer 2. The number of nodes of the second layer is m. Each node of this layer corresponds to one fuzzy rule. A node of the second layer is connected with the nodes of the first layer, which form the antecedents of the corresponding rule. Therefore, each node of the second layer can receive from 1 to n input signals. The node output is the degree of fulfillment of the rules, which is calculated as the product of the input signals. Let's denote the nodes outputs of this layer as $\tau_r, r = \overline{1,m}$.

Layer 3. The number of nodes of the third layer is also m. Each node in this layer calculates the relative degree of fulfillment of the fuzzy rule

$$\tau_r^* = \frac{\tau_r}{\sum\limits_{j=\overline{1,m}} \tau_j}. \tag{3}$$

Layer 4. The number of nodes of the fourth layer is also m. Each node is connected to one node of the third layer and also with all network inputs (in Fig. 3 the connection with the inputs are not shown). The node of the fourth layer calculates the contribution of a fuzzy rule in the output of the network

$$y_r = \tau_r^* \cdot \left(b_{0,r} + b_{1,r} x_1 + ... + b_{n,r} x_n \right). \tag{4}$$

Layer 5. The only node of this layer sums the contributions of all the rules

$$y = y_1 + ... y_r ... + y_m . \tag{5}$$

In this work the authors' studies have shown, that the shape of input variables linguistic terms membership functions significantly affect the training process of the ANFISTC and MCS model accuracy on the whole. So, to achieve the highest accuracy of the MCS neuro-fuzzy mathematical model, at the stage of its design the ANFISTC synthesis for the different types of linguistic terms membership functions (triangular, trapezoidal, Gaussian 2) of the input variables P_{HR}, T_{TR}, Q_A, T_A is considered. 3 linguistic terms are chosen for the variable P_{HR}: S – small, M – middle and B – big; for the variables T_A and T_{TR} 3 linguistic terms are also chosen for each one: L - Low, M – middle and H – High; for the variable Q_A – 5 terms: Z – Zero, S – small, M – middle, B – big and VB – Very Big. The ANFISTC knowledge base consists of 135 rules, which correspond to all combinations of four input fuzzy variables. In this case the amount of rules coefficients $b_{q,r}$ is 540.

During ANFISTC training, at different types of membership functions the input variables linguistic terms parameters are found, as well as all rules coefficients $b_{q,r}$.

ANFISTC training was conducted with the help of training data with 12400 points and the hybrid training method, which combines the back-propagation method and the least square method. The training data is based on the experimental characteristics of the MCS output point heating transient processes of the real EPG complex in different operation modes.

For the input variables linguistic terms membership functions of triangular type the ANFISTC training process lasted 3147 epochs, and the minimum training error is 0,879.

Linguistic terms membership functions of the input variable Q_A, as well as the coefficients $b_{q,r}$ of some rules are shown in Fig. 4 and Table 1, respectively (for input variables P_{HR} and Q_A the range of values is determined in relative units).

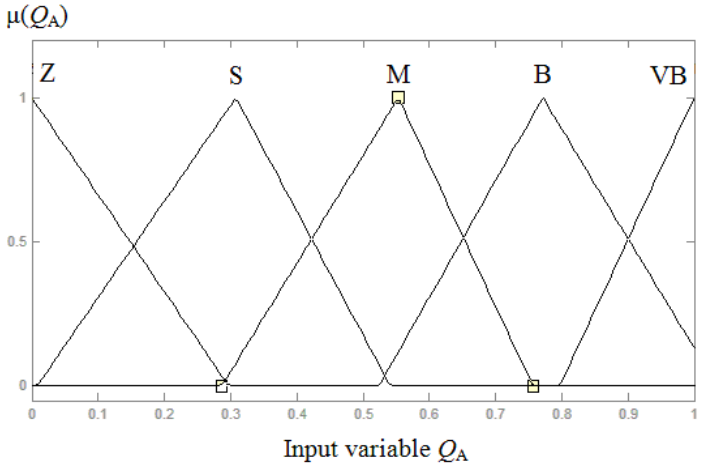


Fig. 4. ANFISTC input variable linguistic terms membership functions of triangular type

Table 1. ANFISTC rule base fragment for the input variables membership functions of triangular type

	№ of rule				
	1	**66**	**73**	**81**	**86**
P_{HR}	S	M	M	M	M
T_A	L	M	M	H	H
T_{TR}	L	M	H	M	H
Q_A	Z	Z	M	Z	Z
$b_{1,r}$	0	2,53	2,39	2,016	3,29
$b_{2,r}$	0	0,232	0,194	0,198	0,643
$b_{3,r}$	0	0,324	0,245	0,267	0,322
$b_{4,r}$	0	0	-8,82	0	0
	№ of rule				
	126	**130**	**131**	**133**	**135**
P_{HR}	B	B	B	B	B
T_A	H	H	H	H	H
T_{TR}	M	M	H	H	H
Q_A	Z	VB	Z	M	VB
$b_{1,r}$	2,426	2,53	4,231	4,01	3,455
$b_{2,r}$	0,314	0,156	0,594	0,447	0,213
$b_{3,r}$	0,315	0,127	0,478	0,331	0,226
$b_{4,r}$	0	-3,23	0	-6,56	-2,79

The characteristic surface of the developed ANFISTC for the input variables membership functions of triangular type at $P_{HR} = P_{HRmax}$ and $T_A = 20$ °C is presented in Fig. 5.

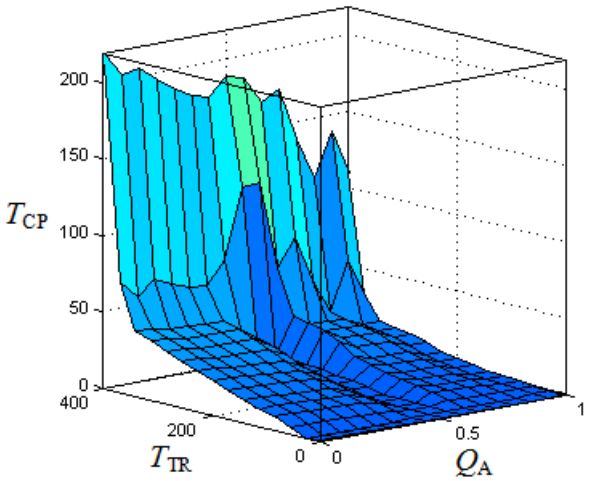


Fig. 5. ANFISTC characteristic surface for the input variables membership functions of triangular type

For the input variables linguistic terms membership functions of trapezoidal type the ANFISTC training process lasted 4725 epochs, and the minimum training error is 0,0934.

Linguistic terms membership functions of the input variable Q_A, as well as the coefficients $b_{q,r}$ of some rules are shown in Fig. 6 and Table 2, respectively.

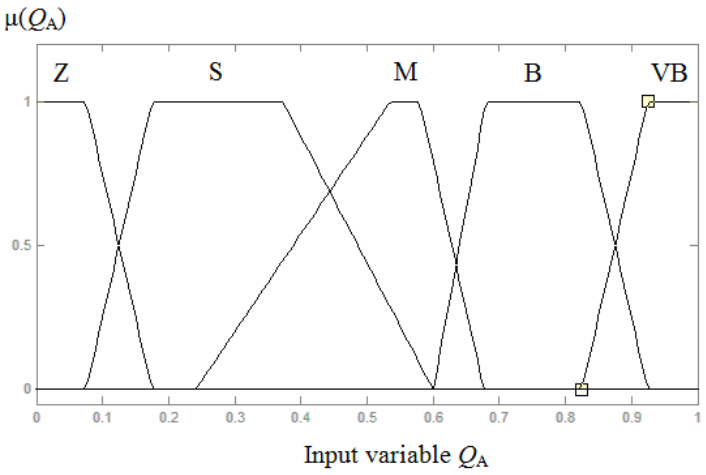


Fig. 6. ANFISTC input variable linguistic terms membership functions of trapezoidal type

The characteristic surface of the developed ANFISTC for the input variables membership functions of trapezoidal type at $P_{HR} = P_{HRmax}$ and $T_A = 20$ °C is presented in Fig. 7.

Table 2. ANFISTC rule base fragment for the input variables membership functions of trapezoidal type

	№ of rule				
	1	**66**	**73**	**81**	**86**
P_{HR}	S	M	M	M	M
T_A	L	M	M	H	H
T_{TR}	L	M	M	M	H
Q_A	Z	Z	M	Z	Z
$b_{1,r}$	0	2,04	2,09	2,13	3,41
$b_{2,r}$	0	0,182	0,204	0,212	0,627
$b_{3,r}$	0	0,224	0,227	0,243	0,354
$b_{4,r}$	0	0	-8,16	0	0
	№ of rule				
	126	**130**	**131**	**133**	**135**
P_{HR}	B	B	B	B	B
T_A	H	H	H	H	H
T_{TR}	M	M	H	H	H
Q_A	Z	VB	Z	M	VB
$b_{1,r}$	2,42	2,03	4,81	4,02	3,85
$b_{2,r}$	0,342	0,142	0,684	0,455	0,265
$b_{3,r}$	0,313	0,103	0,426	0,327	0,22
$b_{4,r}$	0	-3,14	0	-6,04	-2,45

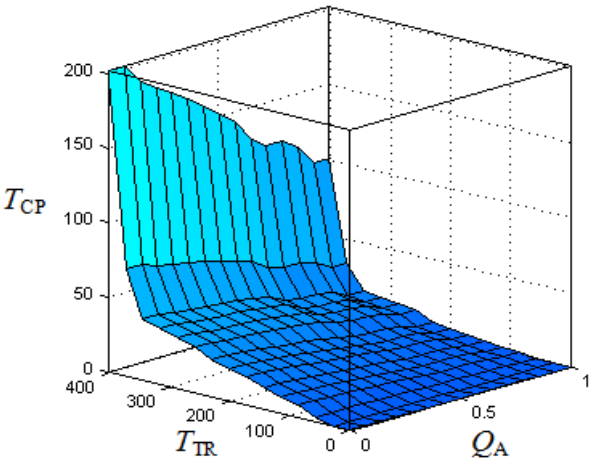


Fig. 7. ANFISTC characteristic surface for the input variables membership functions of trapezoidal type

For the input variables linguistic terms membership functions of Gaussian 2 type the ANFISTC training process lasted 3012 epochs, and the minimum training error is 1,443.

Linguistic terms membership functions of the input variable Q_A, as well as the coefficients $b_{q,r}$ of some rules are shown in Fig. 8 and Table 3, respectively.

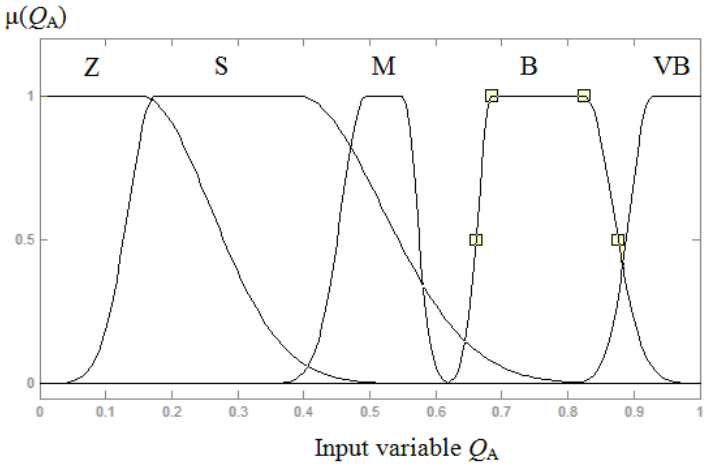


Fig. 8. ANFISTC input variable linguistic terms membership functions of Gaussian 2 type

Table 3. ANFISTC rule base fragment for the input variables membership functions of Gaussian 2 type

	№ of rule				
	1	**66**	**73**	**81**	**86**
P_{HR}	S	M	M	M	M
T_A	L	M	M	H	H
T_{TR}	L	M	H	M	H
Q_A	Z	Z	M	Z	Z
$b_{1,r}$	0	2,12	2,144	2,21	3,523
$b_{2,r}$	0	0,193	0,234	0,223	0,701
$b_{3,r}$	0	0,298	0,267	0,225	0,315
$b_{4,r}$	0	0	-8,02	0	0
	№ of rule				
	126	**130**	**131**	**133**	**135**
P_{HR}	B	B	B	B	B
T_A	H	H	H	H	H
T_{TR}	M	M	H	H	H
Q_A	Z	VB	Z	M	VB
$b_{1,r}$	2,13	2,234	4,176	4,234	3,167
$b_{2,r}$	0,3213	0,243	0,698	0,415	0,215
$b_{3,r}$	0,354	0,096	0,478	0,369	0,224
$b_{4,r}$	0	-3,985	0	-6,435	-2,198

The characteristic surface of the developed ANFISTC for the input variables membership functions of Gaussian 2 type at $P_{HR} = P_{HRmax}$ and $T_A = 20$ °C is presented in Fig. 9.

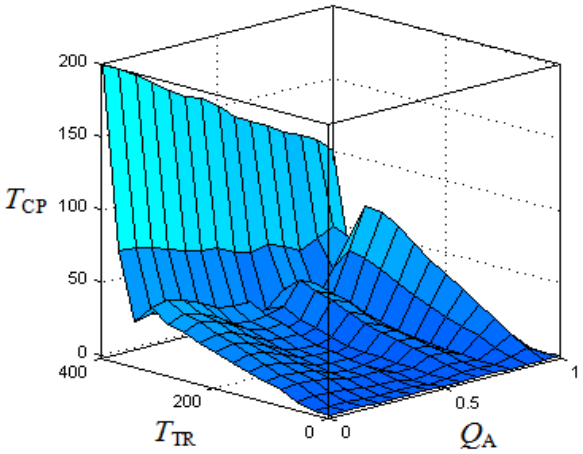


Fig. 9. ANFISTC characteristic surface for the input variables membership functions of Gaussian 2 type

4 Computer Simulation of the MCS as the Temperature Control Object

The computer modeling of the MSC output point heating was carried out for the experimental EPG complex with the following parameters: the displacement volume of the pyrolysis reactor of this complex is 14 liters, the maximum power of gas-burner is 25kW, the nominal power of air fan of the second cooling MCS circuit is 800W. The parameters of transfer function have been defined in the identification process and they are: $T_{1MCS} = 57$ s; $T_{2MCS} = 24$ s; $\tau_{MCS} = 12$ s; $n = 3$. The simulation was carried out at constant heating power of the gas-burner when $P_{HR} = P_{HRmax} = 25$ kW, at boiling point of polymer wastes loaded into the reactor when $T_b = 400°$ C, at constant temperature of ambient $T_A = 20°$ C and the flow of the cooling air of the 2nd circuit of the MCS $Q_A = 0{,}1Q_{Amax}$. Simulation results of the MSC output point heating transient processes, using the developed MCS neuro-fuzzy mathematical model with input variables linguistic terms of different types, are graphically shown in Fig.10.

The detailed graphics of the MSC output point heating transient processes are graphically shown in Fig.11.

After analyzing the simulation results we can confirm that the mathematical model has rather high level of adequacy as the nature of its transient processes with high accuracy repeat the character of real processes of MCS heating. The graphics of transient processes (Fig. 10 and Fig. 11) represent the processes of MCS heating, that proceed very slowly before waste start to boil and is carried mainly out at the cost of light evaporation when waste heating and melting. Then, after the waste boiling, at the cost of their intensive evaporation, the MCS heating rate significantly increases.

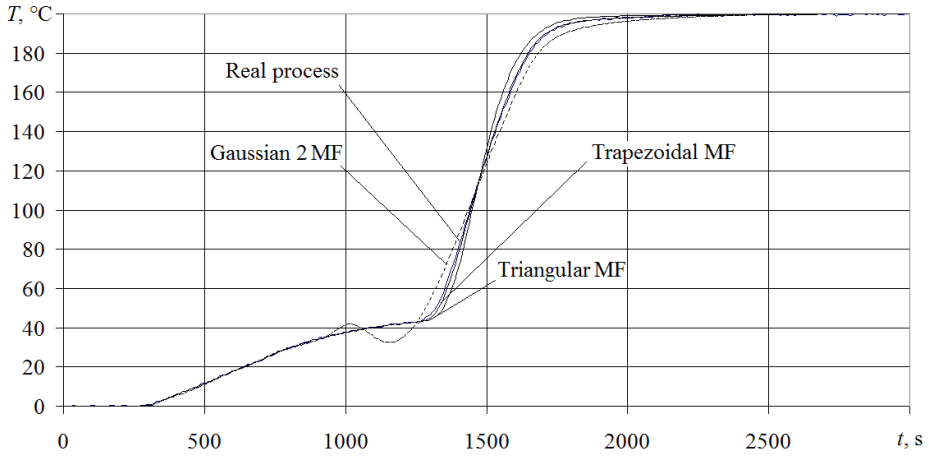


Fig. 10. MSC output point heating transient processes

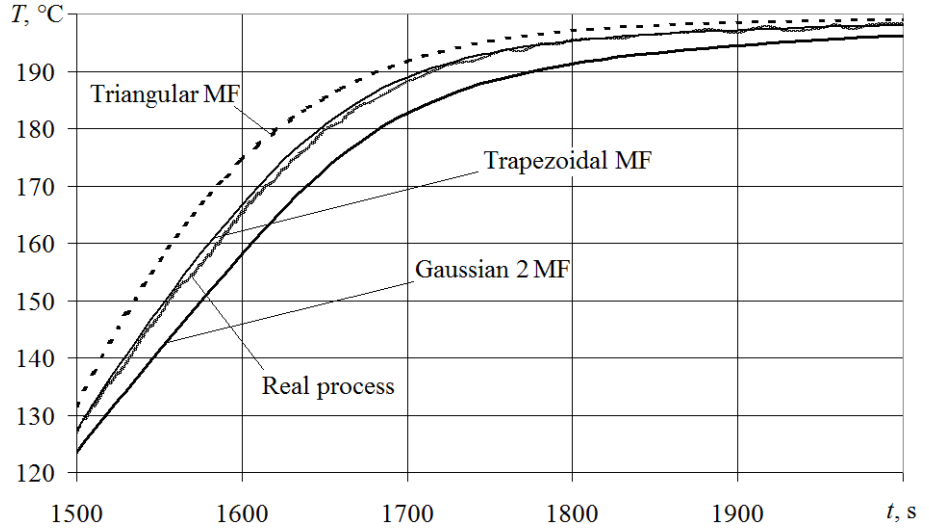


Fig. 11. Detailed graphics of the MSC output point heating transient processes

The highest adequacy to real processes has the neuro-fuzzy mathematical model with ANFISTC input variables linguistic terms of trapezoidal type (training error of its ANFISTC is 0,0934). Also high enough adequacy to real processes has the model with ANFISTC input variables linguistic terms of triangular type (training error of its ANFISTC is 0,879). And the lowest adequacy has the model with ANFISTC input variables linguistic terms of Gaussian 2 type (training error of its ANFISTC is 1,443), its output point heating transient process has a temperature drop on the time interval from 1000 to 1300 seconds, that does not correspond to the real processes of the MSC output point heating.

5 Conclusions

In this work the neuro-fuzzy mathematical model of the EPG complex MCS as a temperature control object is developed.

The obtained model gives the opportunity to study the behavior of the given temperature control object in the steady and transient modes, particularly to synthesize and adjust the temperature controller of the MCS ACS. The application of the mathematical apparatus of the fuzzy logic and artificial neural networks under the development of this model allows to take into account the specific features of the MSC multi-circuit structure and to display with quite high accuracy its basic features as the control object with essentially non-linear and undefined parameters.

The analysis of the received results of the computer simulation in the view of graphics of the static and dynamic MSC characteristics as the temperature control object shows that the highest adequacy of the developed model to the real processes is achieved with ANFISTC input variables trapezoidal membership functions.

References

1. Markina, L.M.: Development of New Energy-saving and Environmental Safety Technology at the Organic Waste Disposal by Ecopyrogenesis. J. Collected Works of NUS 4, 8 (2011) (in Ukrainian)
2. Chaikin, B.S., Mar'yanchik, G.E., Panov, E.M., Shaposhnikov, P.T., Vladimirov, V.A., Volovik, I.S., Makarevich, B.A.: State-of-the-art Plants for Drying and High-Temperature Heating of Ladles. International Journal of Refractories and Industrial Ce-ramics 47(5), 283–287 (2006)
3. Štemberk, P., Lanska, N.: Heating System for Curing Concrete Specimens under Prescribed Temperature, 13th Zittau Fuzzy Collqui-um. In: Proceedings of East-West Fuzzy Col-loquium 2006, Zittau, Hochschule Zittau/Goerlitz, Germany, pp. 82–88 (2006)
4. Han, Z.X., Yan, C.H., Zhang, Z.: Study on Robust Control System of Boiler Steam Temperature and Analysis on its Stability. Journal of Zhongguo Dianji Gongcheng Xuebao, Proceedings of the Chinese Society of Electrical Engineering 30(8), 101–109 (2010)
5. Fiss, D., Wagenknecht, M., Hampel, R.: Modeling a Boiling Process Under Uncertain-ties. In: 19th Zittau Fuzzy Colloquium, Proceedings of East-West Fuzzy Colloquium 2012, Zittau, Hochschule Zittau/Goerlitz, Germany, pp. 141–122 (2012)
6. Skrjanc, I.: Design of Fuzzy Model-Based Predictive Control for a Continuous Stirred-Tank Reactor. In: 12th Zittau Fuzzy Colloquium. In: Proceedings of East-West Fuzzy Colloquium 2005, Zittau, Hochschule Zittau/Goerlitz, Germany, pp. 126–139 (2005)
7. Xiao, Z., Guo, J., Zeng, H., Zhou, P., Wang, S.: Application of Fuzzy Neural Network Controller in Hydropower Generator Unit. J. Kybernetes 38(10), 1709–1717 (2009)
8. Kondratenko, Y.P., Kozlov, O.V.: Fuzzy Controllers in Reactors Control Systems of Multiloop Pyrolysis Plants. In: 19th Zittau Fuzzy Colloquium. In: Proceedings of East-West Fuzzy Colloquium 2012, Zittau, Hochschule Zittau/Goerlitz, Germany, pp. 15–22 (2012)
9. Zadeh, L.A.: Fuzzy sets, Information and Control, vol. 8, pp. 338–353 (1965)
10. Jamshidi, M., Vadiee, N., Ross, T.J. (eds.): Fuzzy Logic and Control: Software and Hardware Application. Prentice Hall Series on Environmental and Intelligent Manufacturing Systems (M. Jamshidi, ed.), vol. 2. Prentice Hall, Englewood Cliffs (1993)

11. Jamshidi, M.: On Software and Hardware Application of Fuzzy Logic. In: Yager, R.R., Zadeh, L.A. (eds.) Fuzzy Sets, Neural Networks and Soft Computing, vol. 20, Van Nostrand Reinhold, NY (1994)
12. Hampel, R., Wagenknecht, M., Chaker, N.: Fuzzy Control: Theory and Practice. Physika-Verlag. Heidelberg (2000)
13. Piegat, A.: Fuzzy Modeling and Control. Physica-Verlag, Heidelberg (2001)
14. Oh, S.K., Pedrycz, W.: The Design of Hybrid Fuzzy Controllers Based on Genetic Algorithms and Estimation Techniques. J. Kybernetes 31(6), 909–917 (2002)
15. Suna, Q., Li, R., Zhang, P.: Stable and Optimal Adaptive Fuzzy Control of Complex Systems Using Fuzzy Dynamic Model. J. Fuzzy Sets and Systems 133, 1–17 (2003)
16. Hayajneh, M.T., Radaideh, S.M., Smadi, I.A.: Fuzzy logic controller for overhead cranes. Engineering Computations 23(1), 84–98 (2006)
17. Wang, L., Kazmierski, T.J.: VHDL-AMS Based Genetic Optimisation of Fuzzy Logic Controllers. International Journal for Computation and Mathematics in Electrical and Electronic Engineering 26(2), 447–460 (2007)
18. Ho, G.T.S., Lau, H.C.W., Chung, S.H., Fung, R.Y.K., Chan, T.M., Lee, C.K.M.: Fuzzy rule sets for enhancing performance in a supply chain network. Industrial Management & Data Systems 108(7), 947–972 (2008)
19. Kondratenko, Y.P., Kozlov, O.V., Atamanyuk, I.P., Korobko, O.V.: Computerized Control System for the Pyrolysis Reactor Load Level Based on the Neural Network Controllers. Computing in Science and Technology (2012/2013); Kwater, T., Twarog, B. (eds.): Monographs in Applied Informatics, pp. 97–120. Wydawnictwo Uniwersytety Rzeszowskiego, Rzes-zow (2013)
20. Jang, J.-S.R.: ANFIS: Adaptive-Network-based Fuzzy Inference Systems. IEEE Transactions on Systems, Man, and Cybernetics 23(3), 665–685 (1993)
21. Jang, J.-S.R., Sun, C.-T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall (1996)
22. Dimirovski, G.M., Lokevenc, I.I., Tanevska, D.J.: Applied Adaptive Fuzzy-neural Inference Models: Complexity and Integrity Problems. In: Proceedings of 2nd International IEEE Conference Intelligent Systems, vol. 1, 22-24, pp. 45–52 (2004)
23. Rotach, V.Y.: Automatic Control Theory of Heat and Power Processes, M, Energoatomizdat, p. 296 (1985) (in Russian)
24. Kondratenko, Y.P., Kozlov, O.V.: Mathematic modeling of reactor's temperature mode of multiloop pyrolysis plant. In: Engemann, K.J., Gil-Lafuente, A.M., Merigó, J.M. (eds.) MS 2012. LNBIP, vol. 115, pp. 178–187. Springer, Heidelberg (2012)

Investigation of OWA Operator Weights for Estimating Fuzzy Validity of Geometric Shapes

Abdul Rahman and M.M. Sufyan Beg

Department of Computer Engineering
Jamia Millia Islamia (A Central University)
New Delhi-110025, India
rahman.jhansi@gmail.com, mmsbeg@cs.berkeley.edu

Abstract. The estimation of fuzzy validity (*f*-validity) of complex fuzzy objects (*f*-objects) by using fuzzy geometry (f-geometry) may be a useful tool in revealing unknown links or patterns e.g. finger prints, shoe print, face sketch of a criminal etc. at crime site. The Extended Fuzzy Logic (FLe) is a combination of Fuzzy Logic (FL) and Unprecisiated Fuzzy Logic (FLu). Whenever a precise solution of any problem is either impossible or bit costlier, then we opt for the concept of FLe. The f-geometry is an example of Unprecisiated Fuzzy Logic (FLu). The *f*-geometry has different *f*-objects like *f*-point, *f*-line, *f*-circle, *f*-triangle, etc. The aggregation models can be used for aggregating the component of *f*-objects. The Minimizing Distance from extreme Point (MDP), which is a nonlinear ordered weighted averaging (OWA) objective model, is used to estimate *f*-validity of fuzzy objects. The results generated by the MDP model are found closer to degree of OR-ness. The objective of this paper is to lay the foundation and encourage further discussion on the development of` methods for defining as well as estimating *f*-validity of some more complex *f*-objects for forensic investigation services.

1 Introduction

Sometime forensic experts have to make decision on the basis of clues collected by crime site team e.g. face sketch of criminal drawn by expert on the basis of onlooker's statement. Onlooker's statement is based on perception. Practically, exact interpretation of these statements into a face is either impossible or bit costlier. Because face sketch is drawn by free hand without use of ruler and compass. That may result in unclear face of criminal. This may help criminal to escape from law. Computational forensics is an emerging research area focusing on the solution of forensic problems using algorithmic and software methods. Its primary goal is the discovery and advancement of forensic knowledge involving modeling, computer simulation, computer based analysis and recognition [1]. Computer methods enable forensic professionals to reveal and identify previously unknown patterns in an objective and reproducible manners [2]. In forensic research field f-geometry may play a vital role. The f-geometry is a counterpart of Euclidian geometry in crisp theory. In *f*-geometry, objects are drawn by liberated hand without the use of geometric instruments. In FLu,

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_2, © Springer International Publishing Switzerland 2014

there is a concept of perception based f-valid reasoning. The *f*-geometry is an example of FLu. L.A.Zadeh has given the concept of Extended Fuzzy Logic (FL). FLe is a combination of Fuzzy Logic and Unprecisiated Fuzzy Logic. Whenever a perfect solution cannot be given or a process falls excessively costly then the role of the FLe comes into play [3,4,6,7]. The FLe provides the f-valid solution. The Ordered Weighted Averaging (OWA) provides a unified decision making platform under the uncertainty [5]. The f-geometry is a tool that is useful in sketching the different types of geometric shapes on the basis of perceptions i.e. the natural language statements. In this direction Sketching With Words is an emerging research area. In [6,7] authors have applied Sketching With Word technique for the estimation of perceptions in geometric shapes by using triangular membership function.

The f-objects may be the components of the different clues. The estimation of f-objects may be a useful tool in revealing unknown links or patterns e.g. finger prints, shoe print, face sketch of a criminal etc. at crime site. It is clearly shown in Fig.1 the skeleton of a face is made of different f-geometric objects e.g. f-circle, f-triangle, f-quadrilateral, and many more. In f-geometry different f-objects like f-point, f-line, f-circle, f-triangle, f-rectangle, f-square, and f-parallelogram have been defined in [4,6,7]. The f-rhombus is a complex f-object. The definition of f-rhombus as well as a method for estimation of f-validity is explained in this paper. The f-geometry may provide a framework for estimating the components of the face on the basis of the onlooker's statements. This may provide a scientific basis for human face recognition system by simulating the forensic sketch expert in the discipline of computational forensic. The Minimizing Distance from extreme Point (MDP) is most recent model and results generated by MDP aggregation model are closer to degree of OR-ness. So we have used MDP model for estimating the f-validity of f-rhombus. The estimation of f-rhombus may be a useful model for defining more complex f-objects. The objective of this paper is to lay the foundation and encourage further discussion on the development of` methods for defining as well as estimating f-validity of some more complex f-objects for forensic investigation services.

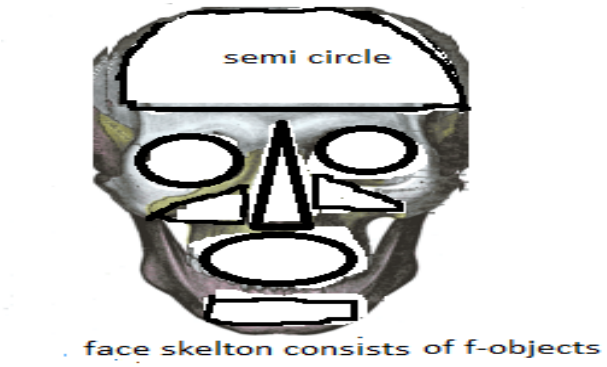


Fig. 1. Skeleton of Face

In this way f-geometry along with OWA operator weights may provide a vital link in catching the criminal in the field of computational forensics. In the literature some nonlinear Ordered Weighted Aggregation objective models like Maximum Entropy Model (MEM), Minimal Variability Model (MVM), Chi Square Model (CSM), Least Square Deviation Model (LSM) and Minimizing Distance from extreme Point Model (MDP) have been reported [9]. In [10] authors have analyzed the capabilities of memetic algorithms for the 3D image reconstruction of forensic objects.

The proposed work may be considered as a first step in aggregating the component of complex f-objects. Further we can aggregate the different parts of complete shape e.g. face sketch. In this paper the methodology of Computing With Words is not taken into account. That is out of scope for this paper. Only f-line and f-similar angle are used in this work. That is why we are going to explain only these two components. Please refer [4,6,7,8] in order to have the details of the rest of the components of f-geometry.

This paper is organized as follows. In Section 2, we have briefly looked into the f-geometry. The section 3 incorporates proposed f-theorem. In section 4, we have reviewed OWA techniques for estimating f-validity of f-objects. In section 5, we have estimated the f-validity by using f-theorem and OWA weights with experimental work. The final section comprises of conclusion and future directions.

2 Fuzzy Geometry

The Unprecisiated Fuzzy Logic introduces the concept of f-geometry. In Euclidean geometry crisp concept C corresponds to a fuzzy concept, in f-geometry. In this section we have discussed fuzzy line (f-line), fuzzy similar angle (f- similar angle), fuzzy theorem(f- theorem), fuzzy similarity(f-similarity), fuzzy validity, and fuzzy proof (f-proof). More detail can be seen in [4, 6, 7, 8].

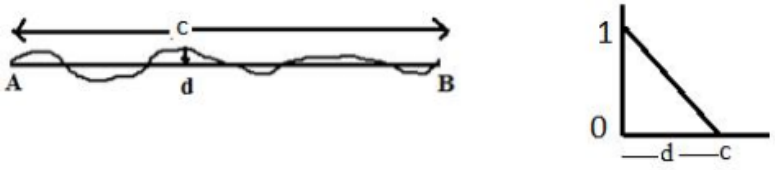


Fig. 2. (a) f-Line (b) Membership Function

2.1 *f*uzzy-Line

Let us consider an f-line which is like a curve that passes through a straight line *AB,* such that the distance between any point on the curve and the straight line *AB* is very small or negligible. With reference to Fig. 2(a), this implies that we are assigning a small value to the distance d [4].

$$\mu(\text{f- line}) = \begin{cases} \frac{c-d}{c} & \text{if} \quad 0 \leq d \leq c \\ 0 & \text{if} \quad c \leq d \end{cases} \tag{1}$$

In equation (1) d << D, where d is the largest difference between the f-line and straight line. D is the length of the crisp line AB. Here c is some real number.

From Fig. 2(b) we can conclude that the membership function increases with the decrease in the value of d. If the value of d is equal to 0 then we can conclude that given f-line is a straight line with validity index 1.

2.2 *f*uzzy Triangle

In f-geometry a closed shape is said to be f-triangle if its membership value is closer to the membership value of triangles. As shown in Fig.3.

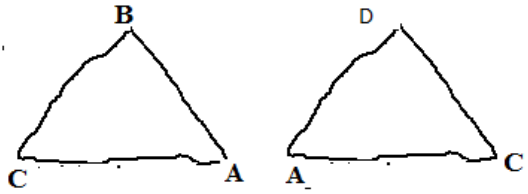


Fig. 3. *f*- Triangles

2.3 *f*uzzy-Similarity and Fuzzy-Validity

In f-geometry any two f-objects are said to be f-similar, if both of them have same shape. Very specifically, by uniform scaling one must be congruent to other. Conversely, f-similar polygons may be of same f-angles and scaling of f-sides may be proportionate. In section 2.4 we illustrate the concept of f-similarity. On the other hand f-validity is a measure of the degree of belongingness of any f-objects against the exact geometric object.

2.4 *f*uzzy-Theorem

The f-theorem in f-geometry is f-transform of a theorem in Euclidean geometry. In f-theorem, we try to formalize the f-concept in f-geometry, generally in the form of membership functions, e.g. by application of any rules. Assuming a formal illustration of the concept of the f-theorem, let us consider the Fig.3. The f- triangles ABC and ADC constituted with three fuzzy lines and three fuzzy angles.

2.4.1 Side Angle Side Postulates of Similar Triangle (SAS)

In f-geometry, two triangles are said to be *f*-similar if their membership function has high validity index for the property of similar triangles (SAS). The membership value of similar triangles decreases as difference in corresponding angle and difference in the proportion of two corresponding sides' increases. Membership function is given by (2).

$$\mu(\text{f-similar})=\mu_{S1}*\mu_{A2}*\mu_{S3} \tag{2}$$

Where μ_{S1} and μ_{S3} are membership functions of *f*- proportions of *f*-sides. The μ_{A2} is membership function of *f*-similar angle.

Equation (3) shows *f*- proportions of *f*-sides and (4) shows *f*-similar angle of *f*-triangle. In case of SAS, we assume that AB/DC' *=BC/AD' *= k (A constant) i.e. corresponding sides of the two triangles are in the same ratio as in geometry. Here AB/DC' and BC/AD' are in the same ratio as in geometry. Here AB/DC' and BC/AD' have taken the fuzzy proportion values $k_1$and k_2 respectively. The point to be noted here is AB/DC *= BC/AD means AB/DC' is approximately equals to BC/AD' [6,7]. The membership function of the difference in proportion is computed by (3). Where the value of j is given by $k-k_1$ and $k-k_2$ for AB/DC' and BC/AD' respectively.

$$\mu(f\text{-similar side})=\begin{cases} \dfrac{c-j}{c-b} & \text{if } b\le j\le c \\ 0 & \text{if } c\le j \end{cases} \tag{3}$$

In the following equation membership function of difference in θ_1 and θ_2 angle is given .

$$\mu(f\text{-similar angle})=\begin{cases} \dfrac{c-h}{c-b} & \text{if } b\le h\le c \\ 0 & \text{if } c< h \end{cases} \tag{4}$$

Here, $h=\theta_1-\theta_2$ is the difference between the angles as shown in Fig. 4.

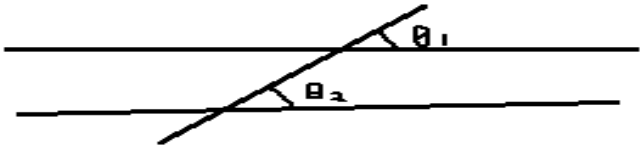


Fig. 4. *f*-Angle

2.5 *f*uzzy-Proof

In f-geometry, f-proof may be either pragmatic or logical. The pragmatic f-proof involves experiments while the logical f-proof is the f-transform of their counterpart of crisp geometry [1,4,5].

3 Proposed *f*uzzy-Theorem

This section introduces the definition of the f-rhombus. Further we have discussed f-theorem and their f-proof by using validation principle. Afterwards we have estimated f-validity of f-rhombus with MDP OWA weights technique. Similar concept can be used for defining more complex f-objects.

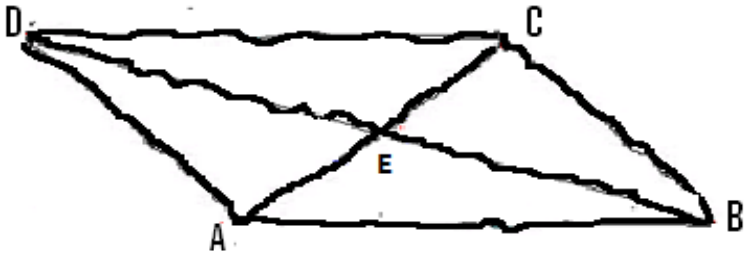


Fig. 5. *f*-Rhombus

3.1 *f*-Rhombus

In a four sided closed shape if all of its *f*-sides are *f*-similar such that the difference between all four *f*-sides are very small or negligible, then it is called f-rhombus as shown in Fig. 5. The membership function of f-rhombus is given below .

$$\mu(f\text{-rhombus}) = \mu_{s1} * \mu_{s2} * \mu_{s3} * \mu_{s4} \tag{5}$$

Where $\mu_{s1}, \mu_{s2}, \mu_{s3},$ and μ_{s4} are given by (3).

f-theorem1: If diagonals (f-lines) of f-parallelogram are perpendicular to each other then f-parallelogram will be a f-rhombus.

f-proof :" if the difference between interior angle made by the diagonals from right angle is either small or negligible " then f-parallelogram ABCD has a higher degree of validity index of f-rhombus. The validity index is given by (6).

$$\mu(\text{f-validity}) = \mu_{IA1} * \mu_{IA2} * \mu_{d1} * \mu_{d2} * \mu_{D} \tag{6}$$

The μ_{IA1},and μ_{IA2} are given by (4).

The μ_D is given by (7)

$$\mu_D = \mu_{d3} * \mu_{d4} * \mu_{d5} * \mu_{d6} \tag{7}$$

4 OWA Operator Weights Methods

4.1 Minimizing Distances from Extreme Points Method

This method is introduced by Byeong Seok Ahn. The MDP method has three steps, first step is identification of extreme points on the basis of degree of OR-ness (β) is given by the user, second is generation of weights , and final step is an aggregation of MDP weights with inputs [10].

4.1.1 Identification of Extreme Points

Any weighting vector w ϵK can be represented by a convex combination of the extreme points E= (e_1, e2…e_m) where e_j is a unit vector with 1 in the j^{th} position and 0 elsewhere. A weight set $K^{A\text{-}C}$ can be constructed by combining attitudinal character constraint (β) with set K.

$$K^{A-C} = \{ \mathbf{w}: Aw = b, w \geq 0 \} \tag{8}$$

Where b= (1,β)′ is a 2-D vector and A is a 2xm matrix.

$$A=(a_1,a_2,\ldots.a_m) = \begin{pmatrix} 1 & \ldots & 1 & \ldots & 1 \\ \frac{m-1}{m-1} & & \frac{m-i}{m-1} & & 0 \end{pmatrix}$$

$$A_{ji} = (a_j, a_i)\ j < i,\ j=1\ldots.m$$

$$A_{ij}^{-1} = \frac{m-1}{j-i} \begin{pmatrix} m-j/m-1 & -1 \\ i-m/m-1 & 1 \end{pmatrix} \tag{9}$$

In certain cases, however, a weighting vector **w** is negative due to wrong choice of index j and i. That is avoid deriving a legitimate index set such as

$$j \leq m(1-\beta)+\beta,$$

$$i \geq m(1-\beta)+\beta$$

On the basis of legitimate index, extreme points are given below.

$$ext_j^k = (\beta*(m-1)-(m-i))/(i-j) \tag{10.1}$$

$$ext_i^k = ((m-i)-\beta*(m-1))/(i-j) \tag{10.2}$$

Here β is a level of OR-ness, k denotes the number of extreme points, and m denotes the number of criteria.

4.1.2 Generation of Weights

The coordinates wise averaging of the extreme points results in the MDP OWA operator weights.

$$w_j=\frac{\sum_{j=1}^{k} ext_j^i}{k} \qquad j=1,\ldots.m \tag{11}$$

4.1.3 Aggregation of Weights and Inputs

OWA determines the f-validity in f-objects by using (12). Where $X=(x_1, x_2, x_{3,\ldots} x_m)$ are input parameters with the multi-criteria of size m. The y_i is the i^{th} largest input parameter.

$$OWA(x_1,x_2,x_3\ldots\ldots x_m) =\sum w_j y_i \tag{12}$$

5 Estimation of f-Validity Using OWA Operator

In this section the f-validity of f-rhombus is estimated by using the MDP OWA method.

5.1 Experiments and Results

The computations of the f-validity are performed by applying the concept of f-theorem on f-rhombus. The f-theorem1 is illustrated in Example1. The sample images used in experimental work are shown in Fig. 6.

Example1: The f-rhombus shown in Fig.5 has f-transform distance for the f-lines AB, BC, CD and AD are 2, 25, 3, and 6 respectively. This results in μ_d as {0.97, 0.74, 0.9681, 0.9309} by using (1). The f-transformation distance of f-lines AC and DB are 5 and 6 respectively. The membership values of f-lines AC and DB are estimated by (1) is 0.95 and 0.93 respectively. The values of internal angles DEC and AEB are 102.254 and 94.90173 respectively. The membership values of f-similarity of these angles from right angle calculated by (4) are 0.38, and 0.75.

To compute the value of μ_D we have applied MDP OWA model. Here we consider four parameters as inputs i.e. m=4.The weight vector {0.816,0.1,0.05, 0.033} which is generated by (11) with membership values {0.97, 0.74, 0.9681, 0.9309} produces results by (12) is

$$\begin{aligned}\mu_D &= \mu_{d1}*w_{1+}\mu_{d2}*w_2+\mu_{d3}*w_3+\mu_{d4}*w_4\\ &=0.97*0.816+0.96*0.1+0.93*0.05+0.74*0.033\\ &= 0.95.\end{aligned}$$

Membership of f-rhombus is calculated by using MDP OWA methods. The weight vector for m =5 by (11) is {0.79, 0.1, 0.05, 0.0325, 0.025}. For the degree of ORness (β) 0.9 produces f-validity by (12) is

$$\mu_{(\text{f-rhombus})} = \mu_D * w_1 + \mu_{d5} * w_2 + \mu_{d6} * w_3 + \mu_{IA2} * w_4 + \mu_{IA1} * w_5$$
$$= 0.95*0.79 + 0.95*0.1 + 0.93*0.005 + 0.75*0.0325 + 0.38*0.025$$
$$= 0.92$$

The f-validity is computed by the MDP OWA method for different values of degree of the OR-ness (β) from 0.9 to 0.6 are shown in Fig.7. It can be clearly seen in Fig.7 the results are closer to degree of OR-ness.

Fig. 6. Sample Images

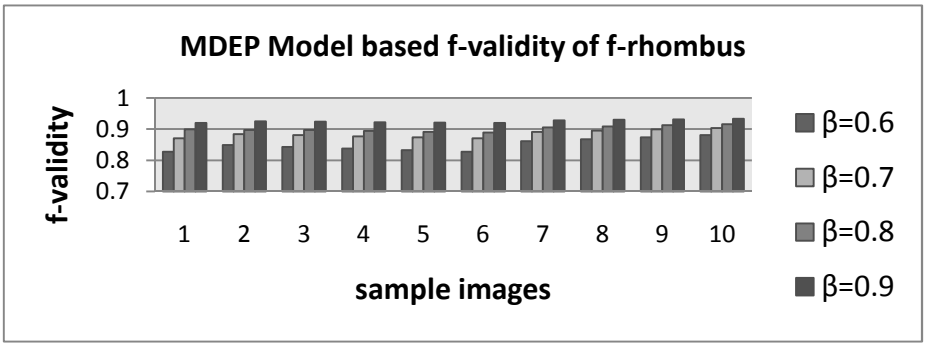


Fig. 7. *f*-validity generated MDP methods by *f*- theorem 1 for sample images of *f*-rhombus

6 Conclusion and Future Directions

In this paper we have proposed the definition of *f*-rhombus in *f*-geometry. Then we have estimated the *f*-validity of *f*-rhombus by transforming a crisp theorem to an *f*-theorem. We have reviewed and applied MDP aggregation model for estimating the degree of OR-ness of the *f*- rhombus. The results generated by MDP are closer to degree of the OR-ness. This characteristic of MDP weights may be very helpful in drawing shape of f-objects with desire degree of OR-ness. This work may open the door for formalizing more complex f-objects. The proposed work may be considered as a first step in aggregating the component of complex f-objects. Further we can aggregate the different parts of face to make a complete face. The concept of *f*-similarity can be used for estimating the similarity of different faces, which may be useful in matching the face of criminal with existing face sketches.

References

1. http://www.cedar.buffalo.edu/forensics/
2. Franke, K., Srihari, S.N.: Computational forensics: An overview. In: Srihari, S.N., Franke, K. (eds.) IWCF 2008. LNCS, vol. 5158, pp. 1–10. Springer, Heidelberg (2008)
3. Zadeh, L.A.: From Fuzzy Logic to Extended Fuzzy Logic—The Concept of f-Validity and the Impossibility Principle, Plenary session. In: FUZZ-IEEE 2007. Imperial College, London (July 24, 2007)
4. Zadeh, L.A.: Toward Extended Fuzzy Logic - A First Step, Fuzzy Sets and Systems. Information Sciences, 3175–3181 (2009)
5. Yager, R.R.: On OWA aggregation operators in multi-criteria decision making. IEEE Transactions on Systems, Man, and Cybernetics 18, 183–190 (1988)
6. Imran, B.M., Beg, M.M.S.: Elements of Sketching with Words. In: Proc of IEEE International Conference on Granular Computing (GrC2010), San Jose, California, USA, August 14-16, pp. 241–246 (2010)
7. Imran, B.M., Beg, M.M.S.: Towards Computational forensics with f-geometry. In: World Conference on Soft Computing' 2011. San Francisco State University, California (2011)
8. Imran, B.M., Beg, M.M.S.: Image Retrieval by Mechanization and f-Principle. In: Proc. Second International Conference on Data Management (ICDM2009), IMT Ghaziabad, India, February 10-11. McMillan Advanced Research Series, pp. 157–165 (2009) ISBN 023-063-762-0
9. Ahn, B.S.: Programming-Based OWA Operator Weights With Quadratic Objective Function. IEEE Transactions on Fuzzy Systems 20(5), 986–992 (2012)
10. Santamaría, J., Cordón, O., Damas, S., García-Torres, J.M., Quirin, A.: Performance evaluation of memetic approaches in 3D reconstruction of forensic objects. Soft Computing 13, 883–904 (2009), doi:10.1007/s00500-008-0351-7

Processing Quantities with Heavy-Tailed Distribution of Measurement Uncertainty: How to Estimate the Tails of the Results of Data Processing

Michal Holčapek[1] and Vladik Kreinovich[2]

[1] Centre of Excellence IT4Innovations, University of Ostrava, Institute for Research and Applications of Fuzzy Modeling, Ostrava, Czech Republic
michal.holcapek@osu.cz
[2] University of Texas at El Paso, El Paso, TX 79968, USA
vladik@utep.edu

Abstract. Measurements are never absolutely accurate; so, it is important to estimate how the measurement uncertainty affects the result of data processing. Traditionally, this problem is solved under the assumption that the probability distributions of measurement errors are normal – or at least are concentrated, with high certainty, on a reasonably small interval. In practice, the distribution of measurement errors is sometimes heavy-tailed, when very large values have a reasonable probability. In this paper, we analyze the corresponding problem of estimating the tail of the result of data processing in such situations.

1 Formulation of the Problem

Need for Data Processing. In many practical situations, we are interested in the values of a quantity y which is not easy (or even impossible) to measure directly: for example, we may be interested in tomorrow's weather, in the distance to a faraway planet, in the amount of oil in an oil well, etc. In such situations in which we cannot measure y *directly*, we can often measure y *indirectly*, i.e.:

- measure the values of auxiliary quantities $x_1, \ldots, x_n$ which are related to the desired quantity y by a known relation $y = f(x_1, \ldots, x_n)$, and then
- use the results $\widetilde{x}_1, \ldots, \widetilde{x}_n$ of measuring the quantities x_i and the known dependence to compute the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ for y.

The process of computing $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is known as *data processing.*

Need to Estimating Uncertainty of the Result of Data Processing. Measurements are never 100% accurate; so, in general, the measurement results $\widetilde{x}_i$ are somewhat different from the actual values x_i of the corresponding quantities. Because of these measurement errors, the estimate $\widetilde{y} = f(\widetilde{x}_1, \ldots, \widetilde{x}_n)$ is, in general, different from the desired value $y = f(x_1, \ldots, x_n)$ (often, there is an additional difference

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_3,

cause by the fact that the dependence between y and x_i is only approximately known). It is therefore important not just to generate an estimate $\widetilde{y}$, but also to gauge how much the actual value y can differ from this estimate, i.e., what is the uncertainty of the result of data processing; see, e.g., [7].

Estimating Uncertainty of the Result of Data Processing: Traditional Statistical Approach. Usually, there are many different (and independent) factors which contribute to the measurement error. In many such situations, it is possible to apply the Central Limit Theorem (see, e.g., [9]), according to which, under reasonable conditions, the distribution of the joint effect of numerous independent factors is close to normal. In such situations, it is therefore reasonable to assume that all the measurement errors $\Delta x_i \stackrel{\text{def}}{=} \widetilde{x}_i - x_i$ are independent and normally distributed.

To describe a normal distribution, it is sufficient to know the mean μ and the standard deviation σ. Thus, under the normality assumption, to gauge the distribution of each measurement error Δx_i, we must know the mean μ_i and the standard deviation σ_i of this measurement error. If the known mean is different from 0, this means that this measuring instrument has a bias; we can always compensate for this bias by subtracting the value μ_i from all the measured values. After this subtraction, the mean error will become 0. Thus, without losing generality, we can assume that each measurement error is normally distributed with mean 0 and known standard deviation σ_i.

The traditional way of estimating the resulting uncertainty $\Delta y \stackrel{\text{def}}{=} \widetilde{y} - y$ in y is based on this assumption. Specifically, since the measurement errors Δx_i are usually relatively small, we can expand the expression

$$\Delta y = \widetilde{y} - y = f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(x_1, \ldots, x_n) =$$

$$f(\widetilde{x}_1, \ldots, \widetilde{x}_n) - f(\widetilde{x}_1 - \Delta x_1, \ldots, \widetilde{x}_n - \Delta x_n)$$

in Taylor series in Δx_i, ignore quadratic and higher order terms, and keep only terms in Δx_i in this dependence. As a result, we get an expression

$$\Delta y = \sum_{i=1}^{n} c_i \cdot \Delta x_i,$$

where $c_i \stackrel{\text{def}}{=} \dfrac{\partial f}{\partial x_i}$. Based on this expression, we conclude that the linear combination Δy of n independent normally distributed random variables is also normally distributed, its mean value of is 0, and its variance σ^2 is equal to: $\sigma^2 = \sum\limits_{i=1}^{n} c_i^2 \cdot \sigma_i^2$ (see, e.g., [7]).

Heavy-Tailed Distributions. There are many practical situations in which the probability distribution for the measurement error is drastically different from normal. In many such situations, the variance is infinite; such distributions are called *heavy-tailed*. Since then, similar heavy-tailed distributions have been empirically found in many other application areas; see, e.g., [1,8].

Historical Comment: Heavy-Tailed Distributions and Fractals. Heavy-tailed distributions surfaced in the 1960s, when Benoit Mandelbrot, the author of fractal theory, empirically studied the fluctuations of prices and showed [4] that large-scale fluctuations follow the Pareto power-law distribution, where for some x_0, for all $x \geq x_0$, the probability density function has the form $\rho(x) = A \cdot x^{-\alpha}$, for some empirical constants $A > 0$ and $\alpha \approx 2.7$. For this empirical value α, variance is infinite.

Mandelbrot studied not only the local price fluctuations, but also the global geometry of the curves describing the dependence of price on time. It turned out that this analysis is closely related to the notion of dimension. Indeed, for sets S which are smooth curves and surfaces and for volumes surrounded by smooth surfaces, dimension can be described as follows. For each $\varepsilon > 0$, we can *ε-approximate* the set S by a finite set $S' = \{s_1, \ldots, s_n\}$, ε-approximate in the sense that:

- every point s from the set S is ε-close to some point $s_i \in S'$, and
- vice versa, every point $s_i \in S'$ is ε-close to some point $s \in S$.

For each set S, we can have ε-approximating sets S' with different number of elements. For each ε, we can gauge the size of the given set S by finding the number of elements $N_\varepsilon(S)$ in the *smallest* ε-approximating finite set.

For a 1-D smooth curve S, the smallest number $N_\varepsilon(S)$ is attained if we take the points $s_1, \ldots, s_n \in S$ located at equal distance $\approx 2\varepsilon$ from each other. The number of such points is asymptotically equal to $N_\varepsilon(S) \sim \text{const} \cdot \dfrac{L}{\varepsilon}$, where L is the length of the curve S.

For a 2-D smooth surface S, the smallest number $N_\varepsilon(S)$ is attained if we take the points on a rectangular 2-D grid with linear step $\approx \varepsilon$. The number of such points is asymptotically equal to $N_\varepsilon(S) \sim \text{const} \cdot \dfrac{A}{\varepsilon^2}$, where A is the area of the surface S.

For a 3-D body S, the smallest number $N_\varepsilon(S)$ is attained if we take the points on a rectangular 3-D grid with linear step $\approx \varepsilon$. The number of such points is asymptotically equal to $N_\varepsilon(S) \sim \text{const} \cdot \dfrac{V}{\varepsilon^3}$, where V is the volume of the 3-D body S.

It turns out that for the price trajectory S, we have $N_\varepsilon(S) \sim \dfrac{C}{\varepsilon^a}$ for some constant C and a fraction (non-integer) a. By analogy with the smooth sets, the value a is called a *dimension* of the trajectory S. Thus, the trajectory S is a set of a fractal dimension; Mandelbrot called such sets *fractals*.

The above empirical result, together with similar empirical discovery of heavy-tailed laws in other application areas, has led to the formulation of *fractal theory*; see, e.g., [5,6].

Comments

- Please note that Mandelbrot's empirical observations only describe the probability density $\rho(x)$ for values $x \geq x_0$; the values $\rho(x)$ for $x < x_0$ can be different.

- In general, the condition that $\int \rho(x)\,dx = 1$ implies that $\alpha > 1$.
- One can easily check that the variance $\int x^2 \cdot \rho(x)\,dx$ is infinite when $\alpha \le 3$.

Problem. If the measurement errors Δx_i of the inputs x_i are distributed according to the heavy-tailed distributions, then what can we conclude about Δy?

What We Do in This Paper. In this paper, we provide an answer to the above question for the simplest cases when data processing consists of applying a single arithmetic operation: addition, subtraction, multiplication, or division.

2 Main Results

Case of Addition $y = f(x_1, x_2) = x_1 + x_2$. For addition, $\Delta y = \Delta x_1 + \Delta x_2$. When the measurement error Δx_1 of the first input has a tail with asymptotics $\rho_1(\Delta x_1) \sim A_1 \cdot |\Delta x_1|^{-\alpha_1}$ and the measurement error Δx_1 of the first input has a tail with asymptotics $\rho_2(\Delta x_2) \sim A_2 \cdot |\Delta x_2|^{-\alpha_2}$, then the tail for Δy has the asymptotics $\rho(\Delta y) \sim A \cdot |\Delta y|^{-\alpha}$ with $\alpha = \min(\alpha_1, \alpha_2)$.

Proof for the Case of Addition $y = f(x_1, x_2) = x_1 + x_2$. We know that $\rho(\Delta y) = \int \rho_1(\Delta x_1) \cdot \rho_2(\Delta y - \Delta x_1)\,d(\Delta x_1)$. Asymptotics mean that for any given accuracy, for sufficiently large values Δx_1 and Δx_2, we have $\rho_1(\Delta x_1) \approx A_1 \cdot |\Delta x_1|^{-\alpha_1}$ and $\rho_2(\Delta x_2) \approx A_2 \cdot |\Delta x_2|^{-\alpha_2}$. What is the asymptotic expression for the probability density $\rho(\Delta y)$ for large values Δy?

A large value of $\Delta y = \Delta x_1 + \Delta x_2$ can come from three different situations:

1) when Δx_1 is large (i.e., the asymptotic expression for $\rho_1(\Delta x_1)$ holds) and Δx_2 is not large in this sense;
2) when Δx_2 is large (i.e., the asymptotic expression for $\rho_2(\Delta x_2)$ holds) and Δx_1 is not large in this sense; and
3) when both Δx_1 and Δx_2 are large in this sense.

The first situation leads to terms proportional to $|\Delta x_1|^{-\alpha_1} = |\Delta y - \Delta x_2|^{-\alpha_1}$. Since in this case, Δx_2 is limited by the threshold after which the values become large, we have $\Delta x_2/\Delta y \to 0$ as $\Delta y \to \infty$ and thus, $|\Delta y - \Delta x_2|^{-\alpha_1} \sim |\Delta y|^{-\alpha_1}$. The second situation similarly leads to terms asymptotically equal to $|\Delta y|^{-\alpha_2}$.

In the third case, for some $K > 1$, the integral which describes $\rho(\Delta y)$ (over the whole real line) can be represented as a sum of the integral $I_{\text{in}}(\Delta y)$ over $[-K \cdot |\Delta y|, K \cdot |\Delta y|]$ and the integral $I_{\text{out}}(\Delta y)$ over the outside of this interval.

The inner integral $I_{\text{in}}(\Delta)$ is bounded by $M \cdot (2K \cdot |\Delta y|)$, where M is the maximum of the the product

$$\rho_1(\Delta x_1) \cdot \rho_2(\Delta y - \Delta x_1) = A_1 \cdot (\Delta x_1)^{-\alpha_1} \cdot A_2 \cdot (\Delta y - \Delta x_1)^{-\alpha_2}.$$

Differentiating this expression w.r.t. Δx_1 and equating derivative to 0, we conclude that $\Delta x_1 = \dfrac{\alpha_1}{\alpha_1 + \alpha_2} \cdot \Delta y$, hence the corresponding maximum is equal to $\text{const} \cdot |\Delta y|^{-(\alpha_1 + \alpha_2)}$. Thus,

$$I_{\text{in}}(\Delta y) \le M \cdot (2K \cdot |\Delta y|) = \text{const} \cdot |\Delta y|^{-(\alpha_1 + \alpha_2 - 1)}$$

for some positive constant.

Outside the interval, $|\Delta y| \le \frac{1}{K} \cdot |\Delta x_1|$, thus, $|\Delta y - \Delta x_1| \le \left(1 + \frac{1}{K}\right) \cdot |\Delta x_1|$ and so,

$$\rho_1(\Delta x_1) \cdot \rho_2(\Delta y - \Delta x_1) = A_1 \cdot |\Delta x_1|^{-\alpha_1} \cdot A_2 \cdot |\Delta y - \Delta x_1|^{-\alpha_2} \le$$

$$A_1 \cdot A_2 \cdot \left(1 + \frac{1}{K}\right)^{-\alpha_2} \cdot |\Delta x_1|^{-\alpha_1} \cdot |\Delta x_1|^{-\alpha_2} = \text{const} \cdot |\Delta x_1|^{-(\alpha_1+\alpha_2)}$$

for some positive constant. Integrating both sides of the resulting inequality, we conclude that

$$I_{\text{out}} \le \int_{-\infty}^{-K\cdot|\Delta y|} \text{const} \cdot |\Delta x_1|^{-(\alpha_1+\alpha_2)}\, d(\Delta x_1) +$$

$$\int_{K\cdot|\Delta y|}^{\infty} \text{const} \cdot |\Delta x_1|^{-(\alpha_1+\alpha_2)}\, d(\Delta x_1) = \text{const} \cdot |\Delta y|^{-(\alpha_1+\alpha_2-1)}$$

for some positive constant.

Both $I_{\text{in}}(\Delta y)$ and $I_{\text{out}}(\Delta y)$ are bounded by $\text{const} \cdot |\Delta y|^{-(\alpha_1+\alpha_2-1)}$, so their sum $\rho(\Delta y)$ is also bounded by a similar expression.

Summarizing: the asymptotic expression for $\rho(\Delta y)$ is the sum of three positive terms of the type $|\Delta y|^{-\alpha}$: a term corresponding to $\alpha = \alpha_1$, a term corresponding to $\alpha = \alpha_2$, and a term bounded by $\alpha = \alpha_1 + \alpha_2 - 1$. Since $\alpha_i > 1$, we have $\alpha_1 + \alpha_2 - 1 > \alpha_i$.

In general, when $\alpha < \alpha'$, then for large z, the ratio $\frac{z^{-\alpha'}}{z^{-\alpha}}$ tends to 0. This means in the sum of power-law asymptotic expressions, the term with the smallest value of α dominates, in the sense that the asymptotics of the sum follows the power law with the smallest possible exponent α. In our case, since $\alpha_1 + \alpha_2 - 1 > \alpha_i$, this smallest exponent is $\min(\alpha_1, \alpha_2)$.

Case of a General Linear Combination. One can check that a similar formula holds for the difference $y = x_1 - x_2$ and, more generally, for an arbitrary linear combination $y = a_0 + \sum_{i=1}^{m} a_i \cdot x_i$. Namely, when the measurement error Δx_i of the the i-th input has a tail with asymptotics $\rho_i(\Delta x_i) \sim A_i \cdot |\Delta x_i|^{-\alpha_i}$, then the tail for Δy has the asymptotics $\rho(\Delta y) \sim A \cdot |\Delta y|^{-\alpha}$ with $\alpha = \min(\alpha_1, \ldots, \alpha_m)$.

Case of Product $y = f(x_1, x_2) = x_1 \cdot x_2$: Analysis of the Problem. For the product, from $y = x_1 \cdot x_2$ and $y + \Delta y = \widetilde{y} = \widetilde{x}_1 \cdot \widetilde{x}_2 = (x_1 + \Delta x_1) \cdot (x_2 + \Delta x_2)$, we conclude that $\Delta y = \Delta x_1 \cdot x_2 + x_1 \cdot \Delta x_2 + \Delta x_1 \cdot \Delta x_2$.

We know the asymptotics of the probability distribution for Δx_1 and Δx_2, so $\Delta x_1 \cdot x_2$ and $x_1 \cdot \Delta x_2$ should have asymptotics with the same exponents α_1 and α_2. Let us find the asymptotics for the product $r \stackrel{\text{def}}{=} \Delta x_1 \cdot \Delta x_2$. Similarly to the case of addition, the corresponding terms come from three cases:

- when Δx_1 is large and Δx_2 is not large; this leads to terms $\sim |r|^{-\alpha_1}$;
- when Δx_2 is large and Δx_1 is not large; this leads to term $\sim |r|^{-\alpha_2}$;
- when both Δx_1 and Δx_2 are large; this leads to the term $\sim |r|^{-(\alpha_1+\alpha_2-1)}$, which (similarly to the case of addition) can be asymptotically ignored in comparison with terms $\sim |r|^{-\alpha_i}$.

Thus, similarly to the case of addition, we have terms with exponent α_1, we have terms with exponent α_2, and we have other terms which can be asymptotically ignored. Hence, we arrive at the following conclusion.

Case of Product $y = f(x_1, x_2) = x_1 \cdot x_2$: Result. When the measurement error Δx_1 of the first input has a tail with asymptotics $\rho_1(\Delta x_1) \sim A_1 \cdot |\Delta x_1|^{-\alpha_1}$ and the measurement error Δx_2 of the second input has a tail with asymptotics $\rho_2(\Delta x_2) \sim A_2 \cdot |\Delta x_2|^{-\alpha_2}$, then $\rho(\Delta y) \sim A \cdot |\Delta y|^{-\alpha}$ with $\alpha = \min(\alpha_1, \alpha_2)$.

Case of Product or Ratio of Several Terms. One can check that a similar formula holds for the ratio $y = x_1/x_2$ and, more generally, for an arbitrary combination $y = a_0 \cdot \prod_{i=1}^{m} x_i^{a_i}$. Namely, when the measurement error Δx_i of the the i-th input has a tail with asymptotics $\rho_i(\Delta x_i) \sim A_i \cdot |\Delta x_i|^{-\alpha_i}$, then the tail for Δy has the asymptotics $\rho(\Delta y) \sim A \cdot |\Delta y|^{-\alpha}$ with $\alpha = \min(\alpha_1, \ldots, \alpha_m)$.

Comment. The main objective of this paper is to deal with measurement (epistemic) uncertainty. However, the same formula can be used if we have *aleatory* uncertainty. For example, we can use these formulas to analyze what happens if:

- we have a population of two-job individuals with first-salary distribution $\rho_1(x_1)$ and second-salary distribution $\rho_2(x_2)$,
- we know that these distributions are independent, and
- we want to find the distribution of the total salary $y = x_1 + x_2$.

3 Future Work: From Asymptotics to a Complete Description of the Corresponding Probability Distributions

Asymptotics for a General Case Remains a Challenge. In the classical statistical approach, it is natural to start with the case of linear functions. Once it is clear how to deal with this case, we can extend our formulas to the case of a general (smooth) function $f(x_1, \ldots, x_n)$: namely, as we have shown in Section 1, we can expand the function $f(x_1, \ldots, x_n)$ into Taylor series and use the fact that in a small vicinity of each point, quadratic (and higher order) terms in this expansion can be safely ignored, and we can approximate the original function by the linear terms in its Taylor expansion. In the classical statistical approach, restriction to a small neighborhood makes perfect sense: for example, for a normal distribution, the probability of the deviation Δx exceeding six standard deviations (6σ) is so small ($\approx 10^{-6}\%$) that such deviations can be safely ignored.

In contrast, for a heavy-tailed distribution, the probability density function $\rho(\Delta x)$ decreases slowly with Δx, as $\rho(\Delta x) \approx A \cdot |\Delta x|^{-\alpha}$. For example, for $\alpha = 2$, the probability of Δx exceeding 6σ is $\approx 6^{-2} \approx 3\%$, which is quite probable. Even deviations of size 100σ are possible: they occur once every 10,000 trials. For such large deviations, we can no longer ignore quadratic or higher order terms; so, we can no longer reduce any smooth function to its linear approximation: each linear function has to be treated separately.

Need to Go from Asymptotics to a Complete Description. In the above text, we only find the exponent α corresponding to the asymptotics of the probability distribution for the approximation error $\Delta y = \widetilde{y} - y$. It is desirable to find the whole distribution for Δy. For that, in addition to the exponent α, we also need to find the following:

- the coefficient A at the asymptotic expression $\rho(\Delta y) \sim A \cdot |\Delta y|^{-\alpha}$;
- the threshold Δ_0 after which this asymptotic expression provides an accurate description of the probability density, and
- the probability density $\rho(\Delta y)$ on the interval $[-\Delta_0, \Delta_0]$ on which the asymptotic expression is not applicable.

Once we know a similar information for the input measurement errors Δx_1 and Δx_2, we can use the formula (3) (or similar formulas corresponding to other data processing algorithms) to estimate the corresponding characteristics for Δy.

What If We Only Have Partial Information about the Distribution of Errors of Direct Measurements. In practice, we only have partial information about the probability distributions $\rho_i(\Delta x_i)$ of the errors Δx_i of direct measurements.

Usually, we consider situations in which we know an interval on which the random variable is located with certainty. For example, for normal distribution with mean μ and standard deviation σ, we can safely conclude that all possible values are located within the six-sigma interval $[\mu - 6\sigma, \mu + 6\sigma]$, since the probability to be outside this interval is $\leq 10^{-8}$. For such distributions, uncertainty means, e.g., that instead of the exact values of the corresponding cumulative distribution functions $F(x) \stackrel{\text{def}}{=} \text{Prob}(X \leq x)$, we only know an interval $[\underline{F}(x), \overline{F}(x)]$ of possible values of $F(x)$. The corresponding interval-valued function $[\underline{F}(x), \overline{F}(x)]$ is known as a *probability box*, or *p-box*, for short; see, e.g., [2,3].

Several algorithms are known for propagating p-boxes via data processing, i.e., for transforming the p-boxes corresponding to the input uncertainty Δx_i to the p-box for the output uncertainty Δy. It is desirable to extend these algorithms so hat they will be able to also cover a similar interval uncertainty about the values A, α, and Δ_0 describing the heavy-tailed distributions.

Acknowledgments. This work was supported in part by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), by the National Science Foundation grants HRD-0734825, HRD-1242122, DUE-0926721, by Grants 1 T36 GM078000-01 and

1R43TR000173-01 from the National Institutes of Health, and by a grant N62909-12-1-7039 from the Office of Naval Research.

The authors are thankful to the anonymous referees for valuable suggestions.

References

1. Chakrabarti, B.K., Chakraborti, A., Chatterjee, A.: Econophysics and Sociophysics: Trends and Perspectives. Wiley-VCH, Berlin (2006)
2. Ferson, S.: RAMAS Risk Calc 4. CRC Press, Boca Raton (2002)
3. Ferson, S., Kreinovich, V., Hajagos, J., Oberkampf, W., Ginzburg, L.: Experimental Uncertainty Estimation and Statistics for Data Having Interval Uncertainty, Sandia National Laboratories, pp. 2007–2939 (2007), Publ. 2007-0939
4. Mandelbrot, B.: The variation of certain speculative prices. J. Business 36, 394–419 (1963)
5. Mandelbrot, B.: The Fractal Geometry of Nature, Freeman. San Francisco, California (1983)
6. Mandelbrot, B., Hudson, R.L.: The (Mis)behavior of Markets: A Fractal View of Financial Turbulence. Basic Books, New York (2006)
7. Rabinovich, S.: Measurement Errors and Uncertainties: Theory and Practice. Springer, New York (2005)
8. Resnick, S.I.: Heavy-Tail Phenomena: Probabilistic and Statistical Modeling. Springer, New York (2007)
9. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall/CRC, Boca Raton (2007)

A Logic for Qualified Syllogisms

Daniel G. Schwartz

Department of Computer Science, Florida State University,
Tallahassee, Florida 32306-4530, USA
schwartz@cs.fsu.edu
http://www.cs.fsu.edu/~schwartz

Abstract. In various works, L.A. Zadeh has introduced fuzzy quantifiers, fuzzy usuality modifiers, and fuzzy likelihood modifiers. This paper provides these notions with a unified semantics and uses this to define a formal logic capable of expressing and validating arguments such as '*Most* birds can fly; Tweety is a bird; therefore, it is *likely* that Tweety can fly'. In effect, these are classical Aristotelean syllogisms that have been 'qualified' through the use of fuzzy quantifiers. It is briefly outlined how these, together with some likelihood combination rules, can be used to address some well-known problems in the theory of nonmonotonic reasoning.

Keywords: Fuzzy quantification, fuzzy usuality, fuzzy likelihood, qualified syllogisms.

1 Introduction

The notion of *fuzzy quantifier* as a generalization of the classical 'for all" and 'there exists' was introduced by L.A. Zadeh in 1975 [12]. This provided a semantics for fuzzy modifiers such as *most, many, few, almost all*, etc. and introduced the idea of reasoning with syllogistic arguments along the lines of '*Most* men are vain; Socrates is a man; therefore, it is *likely* that Socrates is vain', where vanity is given as a fuzzy predicate. This and numerous succeeding publications [13–18] developed well-defined semantics also for *fuzzy probabilities* (e.g., *likely, very likely, uncertain, unlikely*, etc.) and fuzzy *usuality modifiers* (e.g., *usually, often, seldom*, etc.). In addition, Zadeh has argued at numerous conferences over the years that these modifiers offer an appropriate and intuitively correct approach to nonmonotonic reasoning.

The matter of exactly how these various modifiers are interrelated, however, and therefore of a concise semantics for such syllogisms, was not fully explored. Thus while a new methodology for nonmonotonic reasoning was suggested, it was never developed. The present work grew initially out of an effort to realize this goal. What follows here is a thumbnail sketch of a comprehensive reasoning system that has previously been published as [11]. This work is being presented here to give it visibility within the fuzzy community.

Since the publication of [11], two prominent threads related to nonmonotonic reasoning have emerged. One is the subject of belief revision known as the AGM

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_4,

theory, being named after its originators [1]. A recent summary of the first twenty five years of research in this area is [5]. The other is the study of Answer Set Programming, for which the most well-known reference is [2]. Interest in this subject has spawned eleven conferences, the most recent of which is [3]. Neither of these threads consider anything even remotely resembling the approach being proposed here.

2 Intuitive Motivation

We will define a system **Q** for reasoning with *qualified syllogisms*. In effect, these are classical Aristotelean syllogisms that have been 'qualified' through the use of fuzzy quantification, usuality, and likelihood. (The term 'fuzzy likelihood' is here preferred over 'fuzzy probability', taking the latter to mean a probability that is evaluated as a fuzzy number.) In contrast with the syllogisms originally considered by Zadeh, we here deal only with the case of fuzzy modifiers in application to crisp (nonfuzzy) predicates. Some examples are

Most birds can fly.
Tweety is a bird.
It is *likely* that Tweety can fly.

Usually, if something is a bird, it can fly.
Tweety is a bird.
It is *likely* that Tweety can fly.

Very few cats have no tail.
Felix is a cat.
It is *very unlikely* that Felix has no tail.

From a common-sense perspective, such arguments are certainly intuitively correct. A more detailed analysis is as follows.

First, note that there is a natural connection between fuzzy quantification and fuzzy likelihood. To illustrate, the statement

Most birds can fly.

may be regarded as equivalent with

If x is a bird, then it is *likely* that x can fly.

The implicit connection is provided by the notion of a statistical sampling. In each case one is asserting

Given a bird randomly selected from the population of all birds, there is a *high probability* that it will be able to fly.

Suppose we express this equivalence as

$$(\textit{Most}\ x)(\mathrm{Bird}(x) \rightarrow \mathrm{CanFly}(x)) \leftrightarrow (\mathrm{Bird}(x) \rightarrow \textit{Likely}\mathrm{CanFly}(x))$$

Table 1. Interrelations across the three kinds of modifiers

Quantification	*Usuality*	*Likelihood*
all	always	certainly
almost all	almost always	almost certainly
most	usually	likely
many/about half	frequently/often	uncertain/about 50-50
few/some	occasionally/seldom	unlikely
almost no	almost never/rarely	almost certainly not
no	never	certainly not

Then the first of the two syllogisms involving Tweety can be reduced to an application of this formula, together with the syllogism

$$\frac{\mathrm{Bird}(x) \rightarrow \mathit{Likely}\mathrm{CanFly}(x) \quad \mathrm{Bird}(\mathrm{Tweety})}{\mathit{Likely}\mathrm{CanFly}(\mathrm{Tweety})}$$

This follows because the left side of the equivalence is the first premise of the original syllogism, and the right side of the equivalence is the first premise of the above syllogism. A key observation to be made here is that the latter syllogism follows by instantiating x with Tweety and applying ordinary (classical) Modus Ponens. This suggests that the desired formulation of fuzzy quantification and fuzzy likelihood may be obtained by adjoining classical logic with an appropriate set of modifiers. It also suggests that the modifiers of interest may be introduced in the manner of either quantifiers or modal operators, and that the semantics for such a system could be based on some version of probability theory.

A second observation is that there is a similar connection between the foregoing two concepts and the concept of usuality. Based on the same idea of a statistical sampling, one has that

Usually, if something is a bird, then it can fly.

is equivalent with the former two assertions. Thus one should be able to include usuality modifiers along with quantifiers and likelihood modifiers in a similar extension of classical logic.

The system **Q** is an outgrowth of these various insights and reflections. In addition to the syllogisms given above, it allows for expression of all similar syllogisms as represented by the lines of Table 1 (where the two 'Tweety' examples are given by the third line, and the 'Felix' example is given by first and last entry of the sixth line).

3 Formal Syntax

We shall begin by defining the kind of languages to be employed. Let the modifiers in Table 1, in top-down then left-right order, be represented by $\mathcal{Q}_3, \ldots, \mathcal{Q}_{-3}$, $\mathcal{U}_3, \ldots, \mathcal{U}_{-3}$, $\mathcal{L}_3, \ldots, \mathcal{L}_{-3}$. As *symbols* select: an (*individual*) *variable*, denoted

by x; countably infinitely many (*individual*) *constants*, denoted generically by $a, b, \ldots$; countably infinitely many unary *predicate symbols*, denoted generically by $\alpha, \beta, \ldots$; seven *logical connectives*, denoted by $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\dot{\rightarrow}$, $\ddot{\neg}$, and $\ddot{\vee}$; the abovementioned modifiers $\mathcal{Q}_i$, $\mathcal{U}_i$, and $\mathcal{L}_i$; and *parentheses* and *comma*, denoted as usual. As will be seen, the dotted connectives are used to formalize part of the metalanguage. Let the *formulas* be the members of the sets

$$\mathrm{F}_1 = \{\alpha(x) | \alpha \text{ is a predicate symbol}\}$$

$$\mathrm{F}_2 = \mathrm{F}_1 \cup \{\neg P, (P \vee Q), (P \wedge Q) | P, Q \in \mathrm{F}_1 \cup \mathrm{F}_2\}^1$$

$$\mathrm{F}_3 = \{(P \rightarrow Q) | P, Q \in \mathrm{F}_2\}$$

$$\begin{aligned}\mathrm{F}_4 = \{&\mathcal{L}_3(P \dot{\rightarrow} \mathcal{L}_i Q), \mathcal{L}_3(P \dot{\rightarrow} \mathcal{Q}_i Q), \mathcal{L}_3(P \dot{\rightarrow} \mathcal{U}_i Q),\\ &\mathcal{Q}_3(P \dot{\rightarrow} \mathcal{L}_i Q), \mathcal{Q}_3(P \dot{\rightarrow} \mathcal{Q}_i Q), \mathcal{Q}_3(P \dot{\rightarrow} \mathcal{U}_i Q),\\ &\mathcal{U}_3(P \dot{\rightarrow} \mathcal{L}_i Q), \mathcal{U}_3(P \dot{\rightarrow} \mathcal{Q}_i Q), \mathcal{U}_3(P \dot{\rightarrow} \mathcal{U}_i Q) |\\ &P, Q \in \mathrm{F}_2 \cup \mathrm{F}_3, i = -3, \ldots, 3\}\end{aligned}$$

$$\mathrm{F}_5 = \{\mathcal{L}_i P, \mathcal{Q}_i P, \mathcal{U}_i P, | P, Q \in \mathrm{F}_2 \cup \mathrm{F}_3, i = -3, \ldots, 3\}$$

$$\mathrm{F}_6 = \mathrm{F}_4 \cup \mathrm{F}_5 \cup \{\ddot{\neg} P, (P \ddot{\vee} Q) | P, Q \in \mathrm{F}_4 \cup \mathrm{F}_5 \cup \mathrm{F}_6\}$$

$$\mathrm{F}'_1 = \{P(a/x) | P \in \mathrm{F}_1 \text{ and } a \text{ is an individual constant}\}$$

$$\mathrm{F}'_2 = \{P(a/x) | P \in \mathrm{F}_2 \text{ and } a \text{ is an individual constant}\}$$

$$\mathrm{F}'_3 = \{P(a/x) | P \in \mathrm{F}_3 \text{ and } a \text{ is an individual constant}\}$$

$$\begin{aligned}\mathrm{F}'_4 = \{&\mathcal{L}_3(P \dot{\rightarrow} \mathcal{L}_i Q)(a/x) | \mathcal{L}_3(P \dot{\rightarrow} \mathcal{L}_i Q) \in \mathrm{F}_4, a \text{ is an individual constant,}\\ &\text{and } i = -3, \ldots, 3\}\end{aligned}$$

$$\mathrm{F}'_5 = \{\mathcal{L}_i P(a/x) | P \in \mathrm{F}_5, a \text{ is an individual constant, and } i = -3, \ldots, 3\}$$

$$\mathrm{F}'_6 = \mathrm{F}'_4 \cup \mathrm{F}'_5 \cup \{\ddot{\neg} P, (P \ddot{\vee} Q) | P, Q \in \mathrm{F}'_5 \cup \mathrm{F}'_6\}$$

where $P(a/x)$ denotes the formula obtained from P by replacing every occurrence of the variable x with an occurrence of the constant a. As abbreviations take

$$(P \ddot{\wedge} Q) \quad \text{for} \quad \ddot{\neg}(\ddot{\neg} P \ddot{\vee} \ddot{\neg} Q)$$

$$(P \ddot{\rightarrow} Q) \quad \text{for} \quad (\ddot{\neg} P \ddot{\vee} Q)$$

$$(P \ddot{\leftrightarrow} Q) \quad \text{for} \quad ((P \ddot{\rightarrow} Q) \ddot{\wedge} (Q \ddot{\rightarrow} P))$$

Formulas without modifiers are *first-* or *lower-level* formulas, and those with modifiers are *second-* or *upper-level*. The members of the set $\mathrm{F}_1 \cup \mathrm{F}'_1$ are *elementary first-* or *lower-level* formulas, and the members of $\mathrm{F}_4 \cup \mathrm{F}'_4 \cup \mathrm{F}_5 \cup \mathrm{F}'_5$ are *elementary second-* or *upper-level* formulas. A lower-level formula is *open* if it contains the variable x, and *closed* if not.

[1] This notation abbreviates the usual inductive definition, in this case the smallest class of formulas containing F_1 together with all formulas that can be built up from formulas in F_1 in the three prescribed ways.

By a *language* L is meant any collection of symbols and formulas as described above. Languages differ from one another essentially only in their choice of individual constants and predicate symbols. As an example, the first of the foregoing syllogisms can be written in a language employing the individual constant a for Tweety and the predicate symbols α and β for Bird and CanFly—and, for clarity, writing these names instead of the symbols—as

$$\begin{array}{l} \mathcal{Q}_1(\text{Bird}(x) \rightarrow \text{CanFly}(x)) \\ \mathcal{L}_3\text{Bird}(\text{Tweety}) \\ \hline \mathcal{L}_1\text{CanFly}(\text{Tweety}) \end{array}$$

In words: For *most* x, if x is a Bird then x CanFly; it is *certain* that Tweety is a Bird; therefore it is *likely* that Tweety CanFly.

4 The Bayesian Semantics

This section and the next define two alternative semantics for **Q**, one Bayesian and one non-Bayesian. The first will be the more general, but the second will be more useful for certain kinds of applications. In both semantics, an *interpretation* I for a language L will consist of a *likelihood mapping* l_I which associates each lower-level formula with a number in $[0, 1]$, and a *truth valuation* v_I which associates each upper-level formula with a *truth value*, T or F. The subscript I will be dropped when the intended meaning is clear.

Here the definition of l is based on the Bayesian subjectivist theory of probability as described in [8], pp. 29–34. A key feature of Bayesian theory is that it takes the notion of conditional probability as primitive. A *likelihood mapping* l_I *for* an interpretation I *of* a language L, will be any function defined on the lower-level formulas P of L, and the ordered pairs $(Q|P)$ of lower-level formulas of L, satisfying: for elementary P,

$$l(P) \in [0, 1]$$

for ordered pairs $(Q|P)$ of formulas (elementary or not),

$$l(Q|P) \in [0, 1]$$

and, for any P and Q (elementary of not),

$$l(\neg P) = 1 - l(P)$$

$$l(P \wedge Q) = l(Q|P)l(P)$$

$$l(P \vee Q) = l(P) + l(Q) - l(P \wedge Q)$$

$$l(P \rightarrow Q) = l(Q|P)$$

if $l(P) = r$, then for any a, $l(P(a/x)) = r$

$$l(Q|P)l(P) = l(P|Q)l(Q)$$

The value $l(P)$ is here taken to be the Bayesian *degree of belief* (in the truth) of P. The value $l(Q|P)$ is taken to be the Bayesian *conditional probability*, which by

definition is the degree of belief (in the truth) of P under the assumption that Q is known (to be true) with absolute certainty. Under this interpretation common sense would dictate that, if $l(P) = 0$, then $l(Q|P)$ should be undefined. The last of the above equations is a reconstrual of the familiar "inversion formula" (see [8], p. 32) and ensures that $\wedge$ and $\vee$ are commutative. The second from the last line asserts that, if a formula P involving the variable x is held with a certain degree of belief, then in the absence of any special information about an individual a, the formula $P(a/x)$ will be held to the same degree. The only thing left to make any such l a Bayesian probability function is to agree that 'absolute certainty' will be represented by the value 1.

To define the valuation mapping v, one must first select, for each $i = -3, \ldots, 3$, a *likelihood interval* $\iota_i \subseteq [0, 1]$ in the manner of

$$\begin{aligned}
\iota_3 &= [1, 1] \quad \text{(singleton 1)} \\
\iota_2 &= [\tfrac{4}{5}, 1) \\
\iota_1 &= [\tfrac{3}{5}, \tfrac{4}{5}) \\
\iota_0 &= (\tfrac{2}{5}, \tfrac{3}{5}) \\
\iota_{-1} &= (\tfrac{1}{5}, \tfrac{2}{5}] \\
\iota_{-2} &= (0, \tfrac{1}{5}] \\
\iota_{-3} &= [0, 0] \quad \text{(singleton 0)}
\end{aligned}$$

These intervals then become associated with the corresponding modifiers. Their choice is largely arbitrary, but should in principle be guided either by intuition or experimental results based on psychological studies (see [11] for a discussion and references). The only formal requirement is that they be nonoverlapping and cover the interval $[0, 1]$. Given such a set of intervals, the mapping v is defined by, for all $i = -3, \ldots, 3$: for open lower-level P, Q, and with $\mathcal{M}$ being any of $\mathcal{L}$, $\mathcal{Q}$, or $\mathcal{U}$,

$$v(\mathcal{M}_3(P \dot{\rightarrow} \mathcal{M}_i Q)) = T \text{ iff } l(P \rightarrow Q) \in \iota_i$$

for closed lower-level P and Q,

$$v(\mathcal{L}_3(P \dot{\rightarrow} \mathcal{L}_i Q)) = T \text{ iff } l(P \rightarrow Q) \in \iota_i$$

for open lower-level P and $\mathcal{M}$ being any of $\mathcal{L}$, $\mathcal{Q}$, or $\mathcal{U}$,

$$v(\mathcal{M}_i P) = T \text{ iff } l(P) \in \iota_i$$

for closed lower-level P,

$$v(\mathcal{L}_i P) = T \text{ iff } l(P) \in \iota_i$$

and for open or closed upper-level P and Q,

$$v(\ddot{\neg} P) = T \text{ iff } v(P) = F$$

$$v(P \ddot{\vee} Q) = T \text{ iff either } v(P) = T \text{ or } v(Q) = T$$

It is straightforward to verify that this provides a well-defined semantics for the languages in concern. Note that a second-level formula is either T or F, so that

this part of the system is classical. This justifies introducing $\ddot{\wedge}$, $\ddot{\rightarrow}$, and $\ddot{\leftrightarrow}$ in the manner that is customary for classical logic, i.e., via the abbreviations given in Section 3. By contrast, at the lower level there is no similarly convenient syntactical way to express the definition of $l(P \vee Q)$ in terms of $l(P \wedge Q)$, so the two connectives must be defined separately.

To illustrate this semantics, let us verify in detail that the foregoing syllogism regarding Tweety is *valid* in any such interpretation I, i.e. that if the premises of the syllogism are both T in I, then so also will be the conclusion. It will be seen that validity in this example is a direct result of associating $\mathcal{Q}_1$ (*most*) and $\mathcal{L}_1$ (*likely*) with the same likelihood interval. Suppose I is such that

$$v(\mathcal{Q}_1(\text{Bird}(x) \rightarrow \text{CanFly}(x)) = T$$

$$v(\mathcal{L}_3\text{Bird}(\text{Tweety})) = T$$

From the latter we obtain by definition of v that

$$l(\text{Bird}(\text{Tweety})) = 1$$

which means that Bird(Tweety) is absolutely certain. From the former we obtain by definition of v that

$$l(\text{Bird}(x) \rightarrow \text{CanFly}(x)) \in \iota_1$$

By definition of l, this gives

$$l(\text{Bird}(\text{Tweety}) \rightarrow \text{CanFly}(\text{Tweety})) \in \iota_1$$

whence

$$l(\text{CanFly}(\text{Tweety})|\text{Bird}(\text{Tweety})) \in \iota_1$$

In accordance with Bayesian theory, the latter means that the degree of belief in CanFly(Tweety), given that Bird(Tweety) is absolutely certain, is in ι_1. This, together with the above certainty about Tweety being a bird, yields that the degree of belief in CanFly(Tweety) is in ι_1. Then, by definition of l,

$$l(\text{CanFly}(\text{Tweety})) \in \iota_1$$

giving, by definition of v, that

$$v(\mathcal{L}_1\text{CanFly}(\text{Tweety})) = T$$

This is what we were required to show.

In general, it can be shown that the upper-level validates all the axioms and inference rules of classical propositional calculus, in particular, *Modus Ponens*: From P and $P\ddot{\rightarrow}Q$ infer Q. In addition, it validates the *Substitution Rule*: From P infer $P(a/x)$, for any individual constant a, as well as equivalences of the form discussed in Section 2, formally expressed here as

$$\mathcal{Q}_i(\alpha(x) \rightarrow \beta(x))\ddot{\leftrightarrow}(\mathcal{L}_3\alpha(x)\ddot{\rightarrow}\mathcal{L}_i\beta(x)) \qquad (*)$$

for all $i = -3, \ldots, 3$. Verification of these items, together with additional formulas of interest validated by this semantics, can be found in [11].

5 The Counting Semantics

Whenever one uses a quantifier in everyday conversation, there is an implicit reference to an underlying domain of discourse. This observation evidently served as the basis for Zadeh's original formulation of fuzzy quantification. For example, '*Most* birds can fly' refers to a domain of individuals which is presumed to include a collection of birds, and an assertion to the effect that there is a 'high probability' that a randomly chosen bird will be able to fly (Section 2) is represented mathematically by the condition that a 'large proportion' of birds are able to fly.

Unfortunately, the semantics developed in the preceding section does not reflect this type of meaning, since it makes no direct reference to the underlying objects (birds). As such, Bayesian theory does not say anything about how one's degrees of belief are to be determined; it says only that they must be chosen in such a way that they conform to certain laws.

The present section develops an alternative semantics which explicitly portrays the role of the underlying domain. This *counting semantics* arises by restricting Zadeh's notion of 'σ-count' to crisp predicates (see the aforementioned references).

An *interpretation* I for a language L will now consist of a *universe* U_I of *individuals* (here assume U_I is finite), assignment of a unique individual $a_I \in U_I$ to each individual constant a of L, assignment of a unique unary predicate α_I on U_I to each predicate symbol α of L, a *likelihood mapping* l_I which associates each lower-level formula with a number in $[0, 1]$, and a *truth valuation* v_I which associates each upper-level formula with a *truth value*, T or F. As before, the subscript I will be dropped when the intended meaning is clear.

Given assignments for the individual constants and predicate symbols, the mappings l and v are defined in the following way. Observe that the assignments α_I induce the assignment of a unique subset P_I of U_I to each (open) formula P in F_2 according to

$$(\neg P)_I = (P_I)^c$$

$$(P \vee Q)_I = P_I \cup Q_I$$

$$(P \wedge Q)_I = P_I \cap Q_I$$

For subsets $X \subseteq U$, define a *proportional size* σ by

$$\sigma(X) = |X|/|U|$$

where $|\cdot|$ denotes cardinality. Then l is defined by: for $P \in \mathrm{F}_2$,

$$l(P) = \sigma(P_I)$$

for $(P \to Q) \in \mathrm{F}_3$,

$$l(P \to Q) = \sigma(P_I \cap Q_I)/\sigma(P_I)$$

with l undefined if $\sigma(P_I) = 0$; and for $P \in \mathrm{F}_2 \cup \mathrm{F}_3$,

$$\text{if } l(P) = r, \text{ then } l(P(a/x)) = r$$

It is easy to see that σ is a probability function. These definitions merely replicate the standard way of defining probability where events are represented as subsets of a universe of alternative possibilities. The value $\sigma(P_I)$ is defined to be the probability that a randomly selected a_I in U_I will be in P_I. This means that, for each a and each open $P \in \mathrm{F}_2$, and given no additional information about a, $l(P(a/x))$ is the probability that $a_I \in P_I$. The definition of $l(P \to Q)$ is the traditional (non-Bayesian) way of defining conditional probability in terms of joint events (see [8], p. 31). Thus the value of this ratio is, by definition, the probability that an individual a_I will be in Q_I, given that a_I is known to be in P_I.

Assuming this version of l, the corresponding v is defined exactly as in Section 4. It is a routine matter to verify that this semantics validates all the same syllogisms and formulas as were considered for the Bayesian semantics in [11]. (This is not to say, however, that the two semantics are necessarily equivalent with respect to the given class of languages L, an issue which as yet remains unresolved.) To illustrate, the 'Tweety' syllogism can be established as follows. As before, assume that both premises have value T. Letting Pr denote probability, we have

$$
\begin{aligned}
&v(\mathcal{Q}_1(\mathrm{Bird}(x) \to \mathrm{CanFly}(x)) = T \\
&\quad \text{iff } l(\mathrm{Bird}(x) \to \mathrm{CanFly}(x)) \in \iota_1 && \text{(def. } v\text{)} \\
&\quad \text{iff } \sigma(\mathrm{Bird}_I \cap \mathrm{CanFly}_I) \in \iota_1 && \text{(def. } l\text{)} \\
&\quad \text{iff } \forall a_I, \mathrm{Pr}(a_I \in \mathrm{Bird_I}) = 1 \text{ implies} \\
&\qquad \mathrm{Pr}(a_I \in \mathrm{CanFly}) \in \iota_i && \text{(non-Bayes cond.)} \\
&\quad \text{iff } \forall a, l(\mathrm{Bird}(a)) = 1 \text{ implies} \\
&\qquad l(\mathrm{CanFly}(a)) \in \iota_i && \text{(discussion above)} \\
&\quad \text{iff } \forall a, v(\mathcal{L}_3\mathrm{Bird}(a)) = T \text{ implies} \\
&\qquad v(\mathcal{L}_1\mathrm{CanFly}(a)) = T && \text{(def. } v\text{)}
\end{aligned}
$$

Then taking the last line with Tweety as an instance of a and combining this with the second premise of the syllogism gives the desired result. As with the Bayesian semantics, the counting semantics also validates classical propositional calculus at the upper level, as well as all of the same additional formulas discussed in [11].

It would be easy to implement such a mapping σ in any database; one need only scan records and perform counts wherever appropriate. In other types of applications, however (e.g., many expert systems), the underlying universe will be such that it is not possible to count the numbers of objects that satisfy certain relations. For example, it is not known exactly how many birds there are in the world, nor how many of them can fly. Hence instead of basing the likelihood valuation l on actual counts, it would be more reasonable to define it in terms of estimates of sizes of populations. Such estimates might be arrived at by means of statistical samplings; alternatively, they might be subjective estimates of relative sizes, essentially educated guesses, not necessarily based

on any deeper methodology. In the latter case one is nearing a return to the type of reasoning portrayed by the Bayesian semantics. The counting semantics would nonetheless be useful in this context, inasmuch as the principles of set theory can be used to help ensure that these estimates are selected in intuitively plausible ways. For example, if A's are known to always be B's, then in any valid interpretation the set of A's should be a subset of the set of B's. Such an approach might be characterized as subjective, but non-Bayesian.

The restriction to finite universes was made in order to define the counting semantics in terms of relative cardinalities. It seems reasonable, however, that one could extend to infinite domains via an abstract measure-theoretic formulation of probability as in Kolmogorov [7].

6 Application to Nonmonotonic Reasoning

In order to apply the logic Q to the tasks of nonmonotonic reasoning, several additional components are required. First is needed a logic for likelihood combination. For example, if by one line of reasoning one derives *LikelyP*, and by another derives *UnlikelyP*, then one would like to combine these to obtain *UncertainP*. In effect, one needs a set of inference rules covering all possible likelihood combinations. Such a set of rules is described in Table 2, where the numbers are subscripts for the likelihood modifiers. To illustrate, the foregoing example is represented by the 0 in the cell at row 2, column -2. (The $*$ in the upper right and lower left corners represent contradictory conclusions, which can be handled by means of a special 'reason maintenance' process. This is a form of nonmonotonic reasoning first identified by Doyle [4, 6]. A version of this can be formulated via the notion of 'path logic' discussed below. Please see [11] for details.)

Table 2. Rules for likelihood combination

	3	2	1	0	-1	-2	-3
3	3	3	3	3	3	3	*
2	3	2	2	2	2	0	-3
1	3	2	1	1	0	-2	-3
0	3	2	1	0	-1	-2	-3
-1	3	2	0	-1	-1	-2	-3
-2	3	0	-2	-2	-2	-2	-3
-3	*	-3	-3	-3	-3	-3	-3

Second is needed a means for providing such inference rules with a well-defined semantics. A problem arises in that simultaneously asserting *LikelyP* and *UnlikelyP* requires that P have two distinct likelihood values. This cannot be accommodated in a conventional formalism. To remedy this is introduced the notion of a *path* logic, which explicitly portrays reasoning as an activity that takes place in *time*. In effect, one distinguishes between different occurrences of

P in the derivation path (i.e., the sequence of derivation steps normally regarded as a *proof*) by labeling each of them with a *time stamp* indicating its position in the path. In this manner the likelihood mapping can be defined on labeled formulas, in which case each differently labeled occurrence of P can have its own well-defined likelihood value.

A third needed component is a means of distinguishing between predicates that represent *kinds* of things and those that represent *properties* of things. To illustrate, in the 'Tweety' syllogism, 'Bird' represents a kind, whereas 'CanFly' represents a property. For this purpose [11] introduces the notion of a *typed predicate*, indicated formally by superscripts as in $\text{Bird}^{(k)}$ and $\text{CanFly}^{(p)}$.

Last is needed a way of expressing a *specificity relation* between kinds of things, together with an associated *specificity rule*. For example, if '$All(\text{Penguin}^{(k)}(x) \rightarrow \text{Bird}^{(k)}(x)$' is asserted in the derivation path, asserting in effect that the set of penguins is a subset of the set of birds, then one needs to make an extralogical record that $\text{Penguin}^{(k)}$ is more specific than $\text{Bird}^{(k)}$. Given this, one can apply the principle that more specific information always takes priority over less specific.

Collectively, these various components comprise a system for a style of non-monotonic reasoning known as as default reasoning with exceptions. The problems associated with formulating this kind of reasoning have been illustrated by a variety of conundrums, the most well-known being the situation of Opus as illustrated in Figure 1, taken from Touretzky [9]. As the figure indicates, the situation with Tweety is clear, namely, Tweety can fly; but the situation with Opus is contradictory. By one line of reasoning, Opus is a penguin, penguins are birds, and birds can fly, so Opus can fly, whereas by another line, Opus is a penguin, and penguins cannot fly, so Opus cannot fly.

A way to resolve this conundrum can be shown in terms of Figure 2. The diagram portrays the results of adding the following formulas into the derivation path:

1) $\mathcal{L}_3\text{Bird}(\text{Tweety})$
2) $\mathcal{Q}_1(\text{Bird}(x) \rightarrow \text{CanFly}_1(x))$
3) $\mathcal{L}_3\text{Opus}(\text{Penguin})$
4) $\mathcal{Q}_1(\text{Penguin}(x) \rightarrow \text{Bird}(x))$
5) $\mathcal{Q}_1(\text{Penguin}(x) \rightarrow \neg\text{CanFly}_2(x))$

where the subscripts on CanFly indicate the first and second occurrences of $\text{CanFly}(x)$ in the path. Note first that one can obtain the Tweety syllogism described earlier from (1) and (2) as follows. By (*) in Section 4, one can add

6) $\mathcal{Q}_1(\text{Bird}(x) \rightarrow \text{CanFly}(x)) \dot{\leftrightarrow}$
$(\mathcal{L}_3\text{Bird}(x) \ddot{\rightarrow} \mathcal{L}_1\text{CanFly}(x))$

Then, by classical propositional calculus, (2) and (7) yield

7) $(\mathcal{L}_3\text{Bird}(x) \ddot{\rightarrow} \mathcal{L}_1\text{CanFly}(x))$

From this, instantiation of Tweety for x gives

8) $(\mathcal{L}_3\text{Bird}(\text{Tweety}) \ddot{\rightarrow} \mathcal{L}_1\text{CanFly}(\text{Tweety}))$

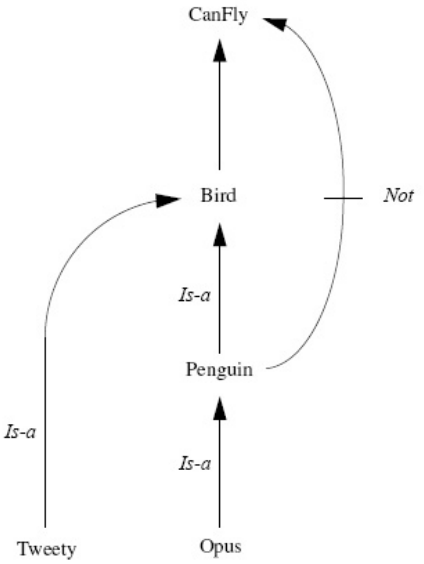


Fig. 1. Tweety can fly, but can Opus?

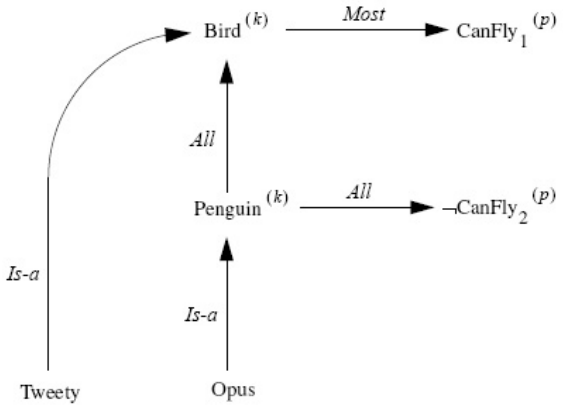


Fig. 2. Tweety *likely* can fly, and Opus *certainly* cannot

Then, by Modus Ponens, (1) and (8) give

9) $\mathcal{L}_1$CanFly(Tweety))

For the case of Opus, one can similarly apply classical propositional calculus to (3), (4), and (2) to derive

10) $\mathcal{L}_1$CanFly(Opus))

and to (3) and (5) to derive

11) $\mathcal{L}_3 \ddot{\neg}\text{CanFly(Opus)})$

Then, by the specificity rule, since Penguin is more specific than Bird, its properties take priority, and one concludes

12) $\mathcal{L}_3 \ddot{\neg}\text{CanFly(Opus)})$

This makes use of the derivation path for allowing different occurrences of CanFly to have different likelihood values, but it does not require likelihood combination. A conundrum that employs likelihood combination rules and which also is handled effectively by this reasoning system is the well-known Nixon Diamond [10]. Again the reader is referred to [11] for details.

References

1. Alchourrón, C.E., Gärdenfors, P., Makinson, D.: On the logic of theory change: partial meet contraction and revision functions. Journal of Symbolic Logic 50(2), 510–530 (1985)
2. Baral, C.: Knowledge Representation, Reasoning, and Declarative Problem Solving. Cambridge University Press, Cambridge (2003)
3. Delgrande, J.P., Faber, W. (eds.): LPNMR 2011. LNCS, vol. 6645. Springer, Heidelberg (2011)
4. Doyle, J.: A truth maintenance system. Artificial Intelligence 12, 231–272 (1979)
5. Fermé, E., Hansson, S.O.: AGM 25 years: twenty-five years of research in belief change. J. Philos Logic 40, 295–331 (2011)
6. Smith, B., Kelleher, G. (eds.): Reason Maintenance Systems and their Applications, Ellis Horwood, Chichester, England (1988)
7. Kolmogorov, A.N.: Foundations of the Theory of Probability, 2nd English edn. Chelsea, New York (1956)
8. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo (1988)
9. Touretzky, D.: Implicit ordering of defaults in inheritance systems. In: Proceedings of the Fifth National Conference on Artificial Intelligence, AAAI 1984, pp. 322–325. Morgan Kaufmann, Los Altos (1984)
10. Touretzky, D., Horty, J.E., Thomason, R.H.: A clash of intuitions: the current state of nonmonotonic multiple inheritance systems. In: Proceedings of IJCAI-1987, Milan, Italy, pp. 476–482.
11. Schwartz, D.G.: Dynamic reasoning with qualified syllogisms. Artificial Intelligence 93, 103–167 (1997)
12. Zadeh, L.A.: Fuzzy logic and approximate reasoning (in memory of Grigore Moisil). Synthese 30, 407–428 (1975)
13. Zadeh, L.A.: PRUF—a meaning representation language for natural languages. International Journal of Man-Machine Studies 10, 395–460 (1978)
14. Zadeh, L.A.: A computational approach to fuzzy quantifiers in natural languages. Computers and Mathematics 9, 149–184 (1983)

15. Zadeh, L.A.: Fuzzy probabilities. Information Processing and Management 20, 363–372 (1984)
16. Zadeh, L.A.: Syllogistic reasoning in fuzzy logic and its application to usuality and reasoning with dispositions. IEEE Transactions on Systems, Man, and Cybernetics 15, 754–763 (1985)
17. Zadeh, L.A.: Outline of a theory of usuality based on fuzzy logic. In: Jones, A., Kaufmann, A., Zimmerman, H.-J. (eds.) Fuzzy Sets Theory and Applications (Reidel, Dordrecht), pp. 79–97 (1986)
18. Zadeh, L.A., Bellman, R.E.: Local and fuzzy logics. In: Dunn, J.M., Epstein, G. (eds.) Modern Uses of Multiple-Valued Logic (Reidel, Dordrecht), pp. 103–165 (1977)

Flexible Querying Using Criterion Trees: A Bipolar Approach

Guy De Tré[1], Jozo Dujmović[2], Joachim Nielandt[1], and Antoon Bronselaer[1]

[1] Dept. of Telecommunications and Information Processing, Ghent University,
Sint-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
{Guy.DeTre,Joachim.Nielandt,Antoon.Bronselaer}@UGent.be
[2] Dept. of Computer Science, San Francisco State University,
1600 Holloway Ave, San Francisco, CA 94132, U.S.A.
jozo@sfsu.edu

Abstract. Full exploration of databases requires advanced querying facilities. This is especially the case if user preferences related to expected results are complex. Traditional query languages like SQL and OQL only have limited facilities for expressing query criteria that are composed of simple criteria. So, while searching for information, users often have to translate their complex requirements (which are typical for human reasoning) into simpler queries, which in many cases can only partly reflect what the user is actually looking for. As a potential solution, we recently proposed a query language extension that is based on soft computing techniques and supports the use of so-called criterion trees. In this paper, we further extend criterion trees so that they can contain both mandatory and optional query conditions. More specifically, we study optional query conditions from a bipolar point of view and propose and illustrate a framework for handling them in query processing.

Keywords: Fuzzy querying, criterion trees, GCD, partial absorption.

1 Introduction

When users want to search for information in a database system, their needs and preferences have to be specified in so-called WHERE-clauses of queries or similar constructs. Traditional query languages like SQL [9] and OQL [2] only support WHERE-clauses in which the query criteria are specified by a Boolean expression that consists of simple expressions connected by logical connectives ($\wedge$, $\vee$ and $\neg$). Parentheses can be used to alter the sequence of evaluation.

Such Boolean expressions do not offer the facilities and flexibility that are required to fulfill complex information needs as often encountered in real-life situations. Indeed, humans often tend to express criteria in a soft way. They structure and group criteria, assign a different importance to different criteria or subgroups of criteria, make a distinction between mandatory and optional criteria, etc. For example, if somebody is searching for a house in a real estate database, it is quite natural to require affordability (acceptable price and maintenance costs) and suitability (e.g., good comfort and a good location). Most

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_5,

homebuyers require simultaneous satisfaction of affordability and suitability criteria and would NOT accept homes where either affordability or suitability is not satisfied, requiring a hard (partial) conjunction operator. If this is not the case, then the aggregator must be a soft partial conjunction. In addition, for some homebuyers affordability is more important than suitability; an opposite criterion is also possible. If the query language criterion cannot express these fundamental requirements, it is not going to be acceptable for most homebuyers. Furthermore, good comfort might be further specified by living comfort and basic facilities. Living comfort refers to at least two bathrooms, three bedrooms, garage, etc., whereas basic facilities refer to gas, electricity, sewage, etc. Good location might be subdivided by accessibility, healthy environment, nearby facilities, etc. Some of these criteria (e.g., garage) might be mandatory, while others (e.g., dentist at close distance) might be optional. Traditional query languages have no specific facilities to deal with such complex search conditions.

Soft computing techniques help to overcome these shortcomings. Soft criteria specifications and criteria preferences can be dealt with by using 'fuzzy' querying techniques of which an overview is, among others, given in [12]. In order to efficiently cope with queries where special aggregators are required or users need to generalize or specialize their criteria for obtaining better insight in what they are looking for, we recently proposed a hierarchically structured criteria specification that is called a *criterion tree* [4]. As originally proposed, criterion trees cannot cope with optional query criteria. Nevertheless, such a facility is required if one wants to adequately support human consistent searching and decision making [1,6]. It is currently subject to a more general research topic that is commonly known as bipolarity in flexible querying (see, e.g., [5]).

In this paper, we propose to extend criterion trees with facilities for handling optional query criteria. The paper is further structured as follows. In Section 2, some preliminaries of criterion trees and bipolar querying are given. Next, we study the aggregation of mandatory and optional query criteria in Section 3. Two aggregation operators *'and optionally'* and *'or optionally'* are proposed. The main advantage of these operators is that they assign a bonus, resp. penalty (or malus), to the query satisfaction depending on the satisfaction, resp. dissatisfaction, of the optional criterion. Next, in Section 4, the extension and evaluation of criterion trees is presented, considering the novel operators. An illustrative example is given in Section 5. Section 6 concludes the paper.

2 Preliminaries

2.1 Criterion Trees

A *criterion tree* is a tree structure of which each node can be seen as a container for information. Each leaf node contains an elementary query criterion c_A, which is defined on a single database attribute A and expresses the user's preferences with respect to the acceptable values for that attribute while computing the answer set of the query. In general, a fuzzy set with membership function μ_A over the domain dom_A of A can be used to define the criterion. The membership

grade $\mu_A(v) \in [0,1]$ of a domain value $v \in dom_A$ then expresses the extent to which v is preferred by the user.

All non-leaf nodes of a criterion tree contain a symbol representing an aggregation operator. Moreover, each child node n_i of a non-leaf node n has an associated weight w_i, reflecting its relative importance within the subset of all child nodes of the non-leaf node. Hereby, for a non-leaf node with k child nodes it must hold that $\sum_{i=1}^{k} w_i = 1$. With this choice, we follow the semantics of weights as used in the LSP methodology [7]. The supported basic aggregators are conjunction (C), hard partial conjunction (HPC), soft partial conjunction (SPC), neutrality (A), soft partial disjunction (SPD), hard partial disjunction (HPD) and disjunction (D). This set is in fact a selection of seven special cases from the infinite range of generalized conjunction/disjunction (GCD) functions and can be easily extended when required [7].

Once specified, criterion trees can be used in the specification of the WHERE-clause of a query. This is illustrated in Section 4. Their evaluation for a relevant database record r results in a criterion satisfaction specification, which can then be used in the further evaluation and processing of the query. Criterion trees are evaluated in a bottom-up way. This means that, when considering a relevant database tuple r, the elementary criteria c_i of the leaf nodes are first evaluated. When specified by a membership function μ_A, the evaluation $\gamma_{c_i}(r)$ of c_i boils down to determining the membership value of the actual value $r[A]$ of A for r, i.e., $\gamma_{c_i}(r) = \mu_A(r[A])$. Next, non-leaf nodes (if any) are evaluated in a bottom-up fashion. A non-leaf node n can be evaluated as soon as all its child nodes n_i, $i = 1, \ldots, k$ have been evaluated. For evaluation purposes, the following implementation of GCD is used [8]:

$$M(x_1, \ldots, x_n; q) = \begin{cases} (\sum_{i=1}^{n} w_i x_i^q)^{1/q} & \text{, if } 0 < |q| < +\infty \\ \prod_{i=1}^{n} x_i^{w_i} & \text{, if } q = 0 \\ \min(x_1, \ldots, x_n) & \text{, if } q = -\infty \\ \max(x_1, \ldots, x_n) & \text{, if } q = +\infty \end{cases} \tag{1}$$

where $x_i \in [0,1]$, $1 \leq i \leq n$ are the input values which represent query satisfaction degrees (hereby, 0 and 1 respectively denote 'not satisfied at all' and 'fully satisfied'). The normalized weights $0 < w_i \leq 1$, $1 \leq i \leq n$, $\sum_{i=1}^{n} w_i = 1$ specify the desired relative importance of the inputs. Furthermore, the computed exponent $q \in [-\infty, +\infty]$ determines the logic properties of the aggregator. Special cases of exponent values are: $+\infty$ corresponding to full disjunction D, $-\infty$ corresponding to full conjunction C and 1 corresponding to weighted average A. The other exponent values q allow to model other aggregators, ranging continuously from full conjunction to full disjunction, and can be computed from a desired value of orness (ω). For aggregation in criterion trees, we used the following numeric approximation for q [7]:

$$q = \frac{0.25 + 1.89425x + 1.7044x^2 + 1.47532x^3 - 1.42532x^4}{\omega(1-\omega)} \tag{2}$$

where

$$x = \omega - 1/2.$$

Suitable orness-values are the following: $\omega = 1/6$ for HPC, $\omega = 5/12$ for SPC, $\omega = 7/12$ for SPD and $\omega = 5/6$ for HPD, assuming that partial disjunction is modeled as the De Morgan dual of partial conjunction.

Considering a database record r, the query satisfaction degree $\gamma_n(r)$ corresponding to n is computed using Eq. (1) and the following arguments: $\gamma_{n_i}(r)$, $i = 1, \ldots, k$, w_i being the weight that has been associated with n_i, $i = 1, \ldots, k$, and q being the value that models the aggregator that is associated with n. The overall satisfaction degree for a record r using a criterion tree is obtained when the root node n_{root} of the tree is evaluated, i.e., by computing $\gamma_{n_{root}}(r)$.

2.2 Bipolar Querying

An important issue in bipolar querying concerns the handling of constraints and wishes (see, e.g., [10,5,1]). Bipolarity hereby refers to the fact that users might distinguish between mandatory and desired criteria while specifying their query preferences. For handling desired criteria, two aggregators 'and if possible' and 'or else' have been proposed in [1] and defined as follows (with $k \in [0, 1]$):

$$\gamma_{c_1 \text{ and if possible } c_2}(r) = \min(\gamma_{c_1}(r), k\gamma_{c_1}(r) + (1 - k)\gamma_{c_2}(r)) \tag{3}$$

and

$$\gamma_{c_1 \text{ or else } c_2}(r) = \max(\gamma_{c_1}(r), k\gamma_{c_1}(r) + (1 - k)\gamma_{c_2}(r)). \tag{4}$$

In what follows, we propose a generalization of these operators which uses slightly different semantics and is based on the conjunctive and disjunctive partial absorption operators as originally proposed and studied in [6].

3 Aggregation of Optional Criteria

The behavior and aggregation of optional criteria has been studied and analyzed in [6] and resulted in the following 'and optionally' (conjunctive partial absorption) and 'or optionally' (disjunctive partial absorption) operators [3].

$$\gamma_{(c_1 \text{ and optionally } c_2)}(r) = w_2\gamma_{c_1}(r) \mathit{\Delta} (1 - w_2)[w_1\gamma_{c_1}(r) \nabla (1 - w_1)\gamma_{c_2}(r)] \tag{5}$$

where $\mathit{\Delta} \in \{C, HPC\}$ and $\nabla \in \{D, SPD, HPD, A\}$, and

$$\gamma_{(c_1 \text{ or optionally } c_2)}(r) = w_2\gamma_{c_1}(r) \nabla (1 - w_2)[w_1\gamma_{c_1}(r) \mathit{\Delta} (1 - w_1)\gamma_{c_2}(r)] \tag{6}$$

where $\nabla \in \{D, HPD\}$ and $\mathit{\Delta} \in \{C, SPC, HPC, A\}$. Both operators are asymmetric.

Using weighted power means as in Eq. (1), both operators can be implemented by

$$M(x_1, x_2; q_1, q_2) = [(1 - w_2)[w_1 x_1^{q_2} + (1 - w_1)x_2^{q_2}]^{q_1/q_2} + w_2 x_1^{q_1}]^{1/q_1} \tag{7}$$

where $x_1 \in [0,1]$ is the mandatory input, $x_2 \in [0,1]$ is the desired input and the exponents q_1 and q_2 are those reflecting the selected aggregators for Δ and ∇ (being computed as described in Section 2).

The weights w_1 and w_2 are computed so as to reflect as adequately as possible the impact of the mean penalty P and mean reward R percentages provided by the user. Hereby the underlying semantics of P and R are defined by the following border conditions [6] (and their dual counterparts for 'or optionally'):

$$\forall\, 0 < x \leq 1 : (x \text{ and optionally } 0) = x(1-p), 0 \leq p < 1 \tag{8}$$

(hence if the optional condition is not satisfied at all, then criterion satisfaction is decreased with a penalty of p)

$$\forall\, 0 < x < 1 : (x \text{ and optionally } 1) = x(1+r), 0 \leq r < 1/x - 1 \tag{9}$$

(hence if the optional condition is fully satisfied, then criterion satisfaction is increased with a reward of r). Note that p and r can be zero. The values P and R are (approximately) the mean values of p and r and usually expressed as percentages. Decision-makers select desired values of P and R and use them to compute the corresponding weights w_1 and w_2. More details on this computation can be found in [6].

By taking $\Delta = C$ and $\nabla = A$, Eq. (3) is obtained as a special case of Eq. (5). Likewise, with $\nabla = D$ and $\Delta = A$, Eq. (6) yields Eq. (4). The main advantage of the 'and optionally' and 'or optionally' operators is that they enable the use of both a reward and a penalty, whereas Eq. (3) and (4) by definition only assign a reward in case of (partial) satisfaction of the optional condition. Such a penalty facility is however required if we want to adequately reflect human reasoning. Consider for example two house descriptions in a database where a mandatory condition 'proximity of bus stop' is perfectly satisfied for both. If an optional condition 'proximity of dentist' is only satisfied for the first house and there is no penalty facility available, then it would not be possible to distinguish between the overall satisfaction of both houses. However, humans would naturally assign a penalty to the second house and prefer the first one.

4 Extended Criterion Trees

Criterion trees can be extended with the 'and optionally' ($ANDOP$) and 'or optionally' ($OROP$) operators. This can be described using Extended BNF (EBNF) [11] by:

```
nabla_conjunction = "D" | "SPD" | "HPD" | "A"
delta_conjunction = "C" | "HPC"
nabla_disjunction = "D" | "HPD"
delta_disjunction = "C" | "SPC" | "HPC" | "A"
aggregator = "C" | "HPC" | "SPC" | "A" | "SPD" | "HPD" | "D"
criterion tree = elementary criterion | composed criterion
composed criterion = aggregator "(" criterion tree":"weight","
```

```
   criterion tree":"weight {"," criterion tree":"weight}")" |
    "ANDOP("nabla_conjunction"," delta_conjunction"," P"," R")
      (Mandatory:" criterion tree", Optional:" criterion tree")" |
    "OROP("nabla_disjunction"," delta_disjunction"," P"," R")
      (Sufficient:" criterion tree", Optional:" criterion tree")"
elementary criterion = attribute "IS {("min value"," suitability")"
   {",(" value"," suitability")" } ",("max value"," suitability")}"
```

where { } means 'repeat 0 or more times'.

The values in the specification of an elementary criterion must form a strictly increasing sequence. Together they specify the piecewise linear membership function μ_A of a fuzzy set that reflects the user's preferred values (suitability) for the attribute A under consideration.

For the asymmetric $ANDOP$ and $OROP$ operators, the first criterion tree reflects the mandatory/sufficient criterion whereas the second criterion tree defines the optional criterion.

Once specified, extended criterion trees can be used in the specification of the WHERE-clause of a query. Extended criterion trees are evaluated bottom-up, by first evaluating the elementary criteria. For a database record r this is done by determining the membership value $\mu_A(r[A])$ of the actual value $r[A]$ of A in r. The composed criteria are evaluated as soon as all their components have been evaluated. Eq. (7) is used for the evaluation of the $ANDOP$ and $OROP$ operators, whereas Eq. (1) is used for the evaluation of the other aggregators.

5 An Illustrative Example

Specifying search criteria for a house in a real estate database is often a complex task. It boils down to specifying weighted criteria and subcriteria, followed by carefully considering penalties and rewards that might result from the satisfaction and dissatisfaction of optional criteria. Assume that the user is looking for an affordable house with good comfort, condition and location. Such a search can be specified using the following SQL statement for regular relational databases.

```
SELECT id, address, price, TREE(c_suitability) AS satisfaction
FROM real_estates r, location l
WHERE (r.location_id=l.id) AND satisfaction>0.5
ORDER By satisfaction
```

The query uses a predefined function $TREE$ which takes an extended criterion tree as argument and computes the overall satisfaction degree ($satisfaction$) of the database records being processed by the query. The criterion tree $c\_suitability$ is specified by

```
c_suitability=HPC(c_comfort:0.4, c_condition:0.4, c_location:0.2)
```

with subtrees $c\_comfort$, $c\_condition$ and $c\_location$. It specifies that a house is considered to be suitable if it is comfortable, the overall condition of the house

is good and its location is adequate. For the aggregation, hard partial conjunction (HPC) is used and comfort and condition are considered to be more important than comfort and location. Typically, users will then specify in more detail what they expect with respect to comfort, condition and location. This is done by specifying each of the three subtrees in more details. For example, *c_location* =

```
ANDOP(A,C,15,10)(Mandatory:
      A(SPC(c_railway_station:0.3, c_road:0.5, c_highway:0.2):0.5,
        HPC(c_sport:0.4, c_doctor:0.2, c_restaurant/bar:0.4):0.5),
            Optional:green_area)
```

This specification reflects that according to the user, a good location is determined by two mandatory criteria and one optional criterion. The mandatory criteria respectively reflect good accessibility and proximity of facilities which are of equal importance to the user. Good accessibility is in this case expressed by proximity of a railway station, proximity of a regional road and proximity of a highway. These three subcriteria are aggregated with a soft partial conjunction operator (SPC) and proximity of a regional road is considered to be more important than the other two criteria. Proximity of facilities is further specified as proximity of sport facilities, proximity of medical practitioners and proximity of bars and restaurants. Hard partial conjunction (HPC) is used for the aggregation. Proximity of green area is considered to be optional. Hence, the 'and optionally' operator ($ANDOP$) is used to combine the mandatory and optional criteria. A penalty of 15% is considered for houses with a complete lack of green area in their environment. Houses that fully satisfy the green area criterion will earn a reward of 10%.

The remaining criteria in the specification of *c_location* are all examples of elementary criteria. Elementary criteria are specified by a membership function. For example, *c_doctor* can be specified by

```
r.distance_to_doctor IS {(5,1), (20,0)}
```

which denotes that travel distances of more than 20 minutes to reach a doctor are unacceptable.

Evaluation of *c_suitability* for a given record r is done with the function $TREE$. This function first evaluates the elementary criteria and then evaluates the internal nodes of the criterion tree in a bottom-up approach. Hereby, an internal node can be evaluated as soon as all of its child nodes have been evaluated. The satisfaction degree resulting from the evaluation of the root node (*c_suitability*) is returned by the function $TREE$. In the query, only records with a satisfaction degree larger than 0.5 will be returned. Of course, in practice, criterion trees can be much more complex than the one given in this example.

6 Conclusions

Criterion trees offer flexible facilities for specifying complex query conditions. In this paper we extended criterion trees with 'and optionally' and 'or optionally'

operators which allow to properly deal with optional query criteria. These operators consistently reflect human reasoning and enable the use of both a reward and a penalty for cases where the optional criteria are satisfied, resp. dissatisfied.

The proposed work is currently being implemented within the framework of the open source PostgreSQL object-relational database system. In further research, we plan to focus on performance and optimization issues.

References

1. Bosc, P., Pivert, O.: On Four Noncommutative Fuzzy Connectives and their Axiomatization. Fuzzy Sets and Systems 202, 42–60 (2012)
2. Cattell, R.G.G., Barry, D.K.: The Object Data Standard: ODMG 3.0. Morgan Kaufmann, San Francisco (2000)
3. De Tré, G., Dujmović, J.J., Bronselaer, A., Matthé, T.: On the Applicability of Multi-criteria Decision Making Techniques in Fuzzy Querying. Communications in Computer and Information Sciences 297, 130–139 (2012)
4. De Tré, G., Dujmović, J., Nielandt, J., Bronselaer, A.: Enhancing flexible querying using criterion trees. In: Larsen, H.L., Martin-Bautista, M.J., Vila, M.A., Andreasen, T., Christiansen, H. (eds.) FQAS 2013. LNCS (LNAI), vol. 8132, pp. 364–375. Springer, Heidelberg (2013)
5. Dubois, D., Prade, H.: Handling bipolar queries in fuzzy information processing. In: Galindo, J. (ed.) Handbook of Research on Fuzzy Information Processing in Databases, pp. 97–114. IGI Global, Hershey (2008)
6. Dujmović, J.J.: Partial Absorption Function. Journal of the University of Belgrade, EE Dept. Series Mathematics and Physics 659, 156–163 (1979)
7. Dujmović, J.J.: Preference Logic for System Evaluation. IEEE Transactions on Fuzzy Systems, vol 15(6), 1082–1099 (2007)
8. Dujmović, J.J.: Characteristic Forms of Generalized Conjunction/Disjunction. In: IEEE World Congress on Computational Intelligence, Hong Kong (2008)
9. ISO/IEC 9075-1:2011: Information technology – Database languages – SQL – Part 1: Framework, SQL/Framework (2011)
10. Lacroix, M., Lavency, P.: Preferences: Putting more knowledge into queries. In: VLDB 1987 Conference, Brighton, UK, pp. 217–225 (1987)
11. Wirth, N.: What Can We Do About the Unnecessary Diversity of Notation for Syntactic Definitions. Communications of the ACM 20(11), 822–823 (1977)
12. Zadrozny, S., De Tré, G., De Caluwe, R., Kacprzyk, J.: An Overview of Fuzzy Approaches to Flexible Database Querying. In: Galindo, J. (ed.) Handbook of Research on Fuzzy Information Processing in Databases, pp. 34–54. IGI Global, USA (2008)

Non-stationary Time Series Clustering with Application to Climate Systems

Mohammad Gorji Sefidmazgi[1], Mohammad Sayemuzzaman[2], and Abdollah Homaifar[1]

[1] Department of Electrical Engineering,
Autonomous Control and Information Technology Center
[2] Department of Environmental Engineering
North Carolina A&T State University
{mgorjise,msayemuz}@aggies.ncat.edu, homaifar@ncat.edu

Abstract. In climate science, knowledge about the system mostly relies on measured time series. A common problem of highest interest is the analysis of high-dimensional time series having different phases. Clustering in a multi-dimensional non-stationary time series is challenging since the problem is ill-posed. In this paper, the Finite Element Method of non-stationary clustering is applied to find regimes and the long-term trends in a temperature time series. One of the important attributes of this method is that it does not depend on any statistical assumption and therefore local stationarity of time series is not necessary. Results represent low-frequency variability of temperature and spatiotemporal pattern of climate change in an area despite higher frequency harmonics in time series.

Keywords: Non-stationary time series, Time series clustering, spatiotemporal pattern.

1 Introduction

Complexity of climate change limits the knowledge about it and therefore decreases its predictability even over a few days. It is complex because many nonlinear variables within the Earth's atmosphere such as temperature, barometric pressure, wind velocity, humidity, clouds and precipitation are interacting. Analyzing climate system in longer timescales and larger areas and also other parameters which influence climate (Earth's surface, Sun, etc.) is complex too. All of climatic variables are observed in limited number of measurement stations and few times per day and limited accuracy, thus our knowledge is restricted to limited time series. Therefore, methods of time series analysis are important in climate [1]. An important characteristic of climatic time series is the non-stationarity. It means that their statistical properties are changing during time and a unique model cannot represent the time series. A common problem in this field is the analysis of high-dimensional time series containing different phases [2]. We assume that the time series has a dynamical model with some time dependent parameters. Then, define a phase (cluster, regime or segment)

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_6,

as a period of time such that during each of these phases, the model parameters are constant. In fact, temporal changes of model parameters take place at a much slower speed than the changes in the system variables themselves. The problem of non-stationary time series clustering is defined to find these regimes numerically [3].

There are many approaches such as Gaussian Mixture Model (GMM) and the Hidden Markov Model (HMM) in literature to detect the phases numerically [4], [5], [6]. In these approaches, one statistical model is assumed for each regime and then the best change points are found by minimizing a cost function in the form of Maximum Likelihood. The cost function is solved by the Expectation Maximization approach.

In this paper, we use a newly developed method based on Finite Elements to find regime changes [7]. The advantage of the FEM method is that it doesn't require any statistical assumption on time series (such as Markovian or Gaussian). It also has intrinsic flexibility to change the persistence of detected regimes. It means it can have longer or shorter regimes by changing some parameters in its procedure. An important characteristic of climatic time series is existence of a linear trend that shows whether variables (for example temperature) are rising or falling. In this work, those regimes are detected in the time series that have different linear trends. When data has a linear trend in each cluster, time series is not locally stationarity and other clustering methods can't solve this problem.

2 Finite Element Method for Clustering

FEM clustering is an approach developed to detect regimes in a non-stationary time series. The basic idea is to assume a model for the time series in each regime, and then find the best switching times and model parameters by solving an optimization problem. This is common in other clustering approaches too. The difference is that the model in each regime can be a non-statistical model. Including additional assumption to the cost function makes it possible to solve this problem using the finite element method. Finally, the minimization problem converted to a linear quadratic programming (LQP) which is solved iteratively to determine the parameters of interest that include the slope and intercept in each regime.

Let x_t be an observed *n*-dimensional time series defined over period of time $[0, T]$. Assuming that we want to fit a first degree polynomial in each regime in the form of $\theta_{0i} + \theta_{1i}.t$ (where *i* is the regime index), we can define *model distance functional*:

$$g(x_t, \theta_i) = \|x_t - (\theta_{0i} + \theta_{1i}.t\,)\|^2 \tag{1}$$

Since time series has *K* regimes with unknown switching times, the overall cost function can be defined as:

$$\sum_{i=1}^{K} \int_0^T \gamma_i(t).\, g(x_t, \theta_i)\, dt \xrightarrow[\Gamma(t),\Theta]{} min \tag{2}$$

where Θ is the time-independent set of unknown parameters and $\gamma_i(t)$ is the cluster affiliation function which are convex and positive. In (2), we have:

$$\Theta = [\theta_1, \dots, \theta_K] \qquad \sum_{i=1}^{K} \gamma_i(t) = 1 \tag{3}$$

$$\Gamma(t) = [\gamma_1(t), \dots, \gamma_K(t)] \qquad \gamma_i(t) \geq 0$$

If all of $\gamma_i(t)$ are 0 or 1 in different times, clusters are *deterministic*. On the other hand if it can have other values between 0 and 1, clusters are *fuzzy*. This can be defined based on application. As stated in [3], numerical solution for this optimization problem is difficult since the number of unknown can be much more than number of known parameters and also no information is available about function $\Gamma(t)$. Therefore, the problem is ill-posed in the sense of *Hadamard* and thus requires adding additional assumptions to solve the problem. This process is known as regularization [8]. For example in *Tikhonov regularization*, additional assumption (called regularization term) is included in the minimization problem. Here, it is assumed that the cluster affiliations functions $\gamma_i(t)$are smooth and their derivative are bounded.

$$\sum_{i=1}^{K} \int_0^T \left[\gamma_i(t).\, g(x_t, \theta_i) + \epsilon^2 \left(\frac{\partial \gamma_i(t)}{\partial t} \right)^2 \right] dt \xrightarrow[\Gamma(t),\Theta]{} min \tag{4}$$

In the above equation, ϵ is called regularization term. To solve the above problem numerically, it must be converted from continuous time domain to discrete-time domain. For this reason, Galerkin discretization is utilized here. Galerkin methods can convert a continuous operator problem (such as a differential equation) to a discrete problem. It is widely used in FEM literature for solving differential equations [9]. In our problem, the FEM basis function defined in the form of *N* triangular functions which are called *hat* functions. A set of continuous functions is defined with the local support on $[0, T]$ as in Figure 1. Applying discretization procedure to $\Gamma(t)$ yields:

$$\gamma_i(t) = \tilde{\gamma}_i(t) + error = \sum_{k=1}^{N} \tilde{\gamma}_i{}^{(k)}.\, v_k(t) + error \tag{5}$$

$$\tilde{\gamma}_i{}^{(k)} = \int_0^T \gamma_i(t).\, v_k(t) dt \tag{6}$$

where $\tilde{\gamma}_i{}^{(k)}$ are scalars called Galerkin coefficient. After some mathematical simplification and using the locality of finite elements basis function support, one can find an optimization in the form of linear quadratic programming.

$$\sum_{i=1}^{K} \left[a(\theta_i)^T \bar{\gamma}_i + \epsilon^2 \bar{\gamma}_i{}^T H \bar{\gamma}_i \right] \xrightarrow[\bar{\gamma},\Theta]{} min \tag{7}$$

$$\bar{\gamma}_i = \left[\tilde{\gamma}_i{}^{(1)}, \dots, \tilde{\gamma}_i{}^{(k)}, \dots., \tilde{\gamma}_i{}^{(N)} \right] \tag{8}$$

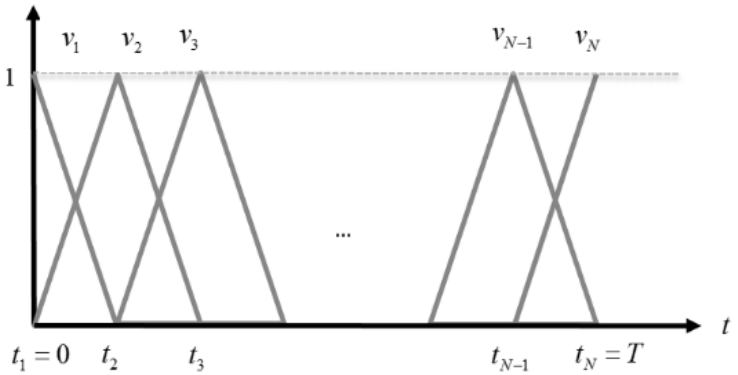


Fig. 1. Finite element basis functions in the form of hat function

$$a(\theta_i) = \left[\int_{t_1}^{t_2} v_1(t)g(x_t,\theta_i)dt, \dots, \int_{t_{k-1}}^{t_{k+1}} v_k(t)g(x_t,\theta_i)dt, \dots, \int_{t_{N-1}}^{t_{N+1}} v_N(t)g(x_t,\theta_i)dt\right] \tag{9}$$

$$H = \begin{pmatrix} \frac{1}{\Delta} & \frac{-1}{\Delta} & 0 & \cdots & 0 & 0 \\ \frac{-1}{\Delta} & \frac{2}{\Delta} & \ddots & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \frac{-1}{\Delta} & 0 & \vdots \\ 0 & 0 & \frac{-1}{\Delta} & \frac{2}{\Delta} & \ddots & 0 \\ \vdots & 0 & 0 & \ddots & \ddots & \frac{-1}{\Delta} \\ 0 & 0 & \cdots & 0 & \frac{-1}{\Delta} & \frac{1}{\Delta} \end{pmatrix} \tag{10}$$

H is a tri-diagonal matrix called *stiffness matrix*. Convexity conditions on model distance functional are converted to constrain on Galerkin coefficients:

$$\sum_{i=1}^{K} \tilde{\gamma}_i^{(k)} = 1 \qquad k = 1, \dots, N \tag{11}$$

$$\tilde{\gamma}_i^{(k)} \geq 0 \qquad i = 1, \dots, K$$

The optimization problem above should be solved with respect to $\bar{\gamma}$ and θ iteratively. After finding Galerkin coefficient, we can build $\gamma_i(t)$ using FEM basis

function and Eq. (5). Initially, some random initial $\bar{\gamma}_i^{initial}$ is assumed such that it fulfills the convexity conditions and then $\theta_i^{initial}$ is found. After that, the iterative procedure is ran for enough iteration numbers. First, the problem is solved with respect to $\bar{\gamma}_i$ for a fixed θ_i and second it is solved with respect to θ_i for a fixed $\bar{\gamma}_i$ and so on. The solution with respect to θ_i is found analytically based on the defined distance functional as:

$$\theta_{i0} = \frac{\begin{vmatrix} \sum_{t=0}^{T} x_t.\gamma_i(t) & \sum_{t=0}^{T} t.x_t.\gamma_i(t) \\ \sum_{t=0}^{T} t.x_t.\gamma_i(t) & \sum_{t=0}^{T} {x_t}^2.\gamma_i(t) \end{vmatrix}}{\begin{vmatrix} \sum_{t=0}^{T} \gamma_i(t) & \sum_{t=0}^{T} t.\gamma_i(t) \\ \sum_{t=0}^{T} t.\gamma_i(t) & \sum_{t=0}^{T} t^2.\gamma_i(t) \end{vmatrix}} \tag{12}$$

$$\theta_{i1} = \frac{\begin{vmatrix} \sum_{t=0}^{T} \gamma_i(t) & \sum_{t=0}^{T} x_t.\gamma_i(t) \\ \sum_{t=0}^{T} t.\gamma_i(t) & \sum_{t=0}^{T} t.x_t.\gamma_i(t) \end{vmatrix}}{\begin{vmatrix} \sum_{t=0}^{T} \gamma_i(t) & \sum_{t=0}^{T} t.\gamma_i(t) \\ \sum_{t=0}^{T} t.\gamma_i(t) & \sum_{t=0}^{T} t^2.\gamma_i(t) \end{vmatrix}} \tag{13}$$

For solving the optimization with respect to $\bar{\gamma}_i$, all the $\bar{\gamma}_i$ are augmented in a vector λ and the problem is converted to one linear quadratic programming.

$$\frac{1}{2}\lambda^T G\lambda + A^T\lambda \xrightarrow[\lambda]{} min \tag{14}$$

$$A = [a(\theta_1), \dots, a(\theta_i), ., a(\theta_K)] \tag{15}$$

$$G = \epsilon^2 \begin{bmatrix} H & 0 & \cdots & 0 \\ 0 & H & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & H \end{bmatrix} \tag{16}$$

and the new constraints become

$$\lambda_s \geq 0 \qquad \forall s = 1, \dots, N \times K \tag{17}$$

$$F.\lambda = Q \tag{18}$$

$$F = \underbrace{[I_{N\times N} \quad I_{N\times N} \quad \dots \quad I_{N\times N}]}_{K\ times} \tag{19}$$

$$Q = [1 \quad 1 \quad \dots \quad 1]^T{}_{N\times 1} \tag{20}$$

There are three parameters that should be set at the start of the procedure: number of clusters K, regularization parameter ϵ and the number of hat functions N (or width of hat functions Δ). Decreasing N reduces the order of LQP and consequently complexity of calculations. On the other hand, it decreases accuracy of clustering and it

may result in losing some very short regimes. A challenging problem arises when choosing K. A trivial solution exists when every data point is a cluster; and as a result, it is not possible to find the optimal number of clusters. By increasing K, the value of the cost function always decreases and when $K = number\ of\ data\ point$, this value approaches to zero. Since the number of clusters is unknown in advance, trial and error along with human judgment is used to select K subjectively. A criterion for choosing K is based on the value of the cluster affiliation functions when it becomes about 1 or 0. If we assume clusters are deterministic, when the cluster affiliation at time t for cluster i is 1, it means the datum at t completely belongs to that cluster (cluster affiliation equal to 0 means that data does not belong to cluster i at all). Increasing ϵ leads to an increase in the length of regimes. To find the optimal parameters, ϵ and K should be changed simultaneously. In the beginning, we set K equals to a sufficiently large number and then decrease K and run the algorithm for different ϵ to find acceptable results, this means the value of $\gamma_i(t)$ is about 0 or 1 in all of the time period [3]. After finding the trends, their statistical significance should be tested using Mann-Kendall approach [10].

3 Application in Climate Data Analysis

In this paper, temperature time series in North Carolina is studied as a case study. A data set of the average temperatures in 249 stations across NC are analyzed from the beginning of 1950 until the end of October 2009. The data is converted from daily to monthly in order to decrease the complexity of calculations. The dimension of resulting time series is 249 ×718. Temperature time series has a dominant harmonics with the period of one year which is called *seasonality*. This annual cycle has been removed by subtracting the multi-year monthly means. This is done, by subtracting the mean that is built over all values corresponding to the same month.

$$x_i^{new} = x_i - \bar{x}_i \tag{21}$$

where x_i^{new} is the deseasonalized value for month i (say for the month of January), x_i is original value for the same month (January) and $\bar{x}_i$ is the average monthly value in month i for the entire period of data (i.e. average of all January's data). Next, the FEM clustering applied to time series. Initially the value of K was assumed to be 10 and the algorithm was executed for different values of the regularization parameter (ε). For each ε, the algorithm ran several times to find best answer for constrained optimization problem. The regularization parameter is a real value between 0 and approximately 30. In this application, we are looking for deterministic clusters. When we reach a K where all the $\gamma_i(t)$ are about 0 or 1, an optimal solution is found. For the time series in this study, we found six regimes with different length and trends.

Figure 2 shows deseasonalized monthly time series in one of the dimensions and the linear trends detected by the FEM. In this figure, narrow lines show deseasonalized temperature time series in one of the stations and bold lines are linear trends

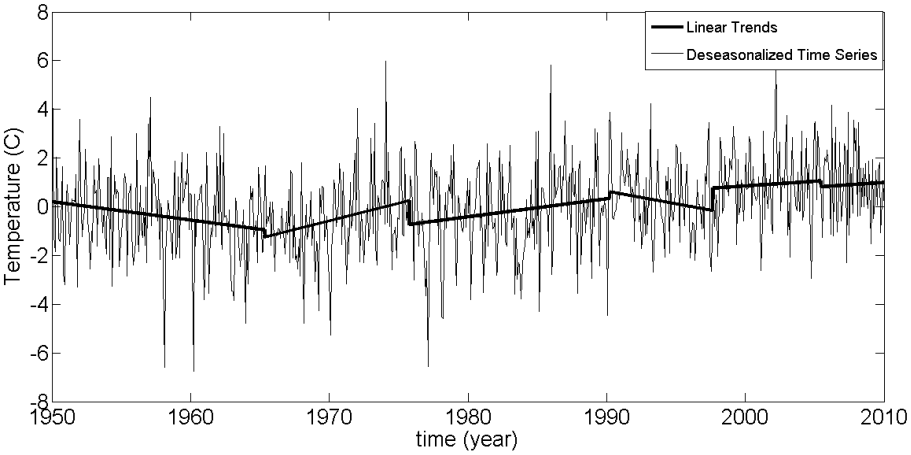


Fig. 2. Time series in one dimension and its regimes/trends

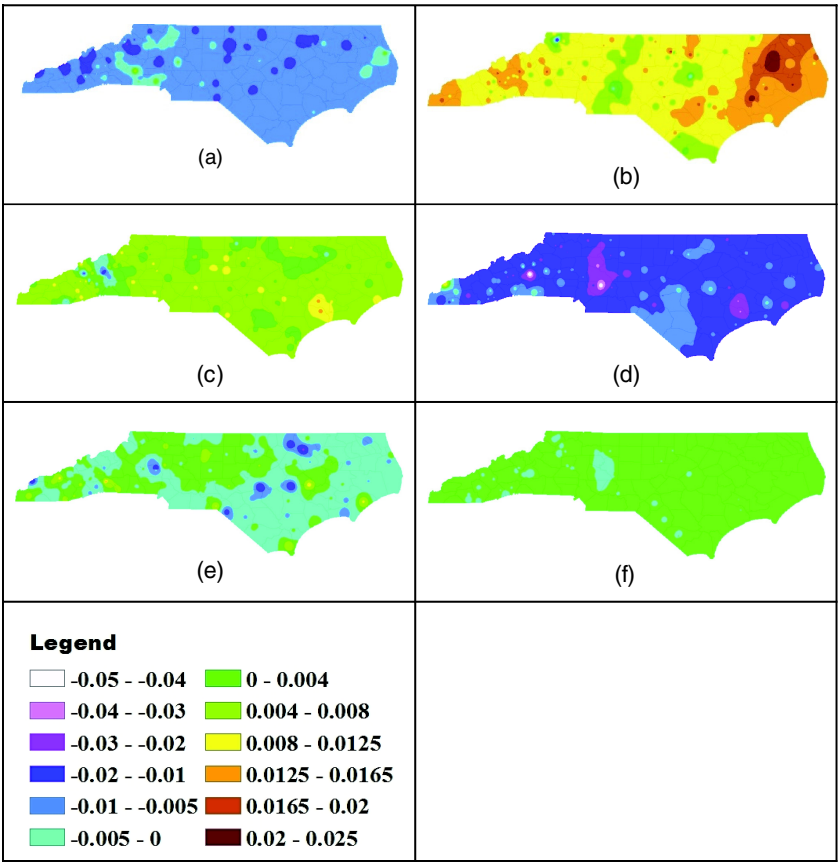


Fig. 3a-3f. Linear trend in NC for six regimes

found by the FEM algorithm. Figure 3a-f shows the value of trend for six regimes in the North Carolina's map. Therefore, FEM clustering can reveal spatial in addition to temporal pattern of climate change. In Figure 3a-f, it is clear that there are two notable decreasing trends between 1965-1964 and 1990-1998. There is a remarkable increase in 1964-1976. Also, the 2nd and the 4th regimes show the warmest and coolest trend, respectively. Warming and cooling in eastern parts of the state in regime 2 and 5 are interesting. Different climatic phenomena may cause these patterns of change in NC, such as *El-Nino*, *Atlantic Multidecadal Oscillation* (AMO) and etc. [1]. We can compare these climatic indices with the results. For example comparison of these trends shows a correlation with AMO. Therefore we may infer that NC temperature is mostly affected by AMO.

4 Conclusion

In this paper, finite element method for clustering a multi-dimensional time series is used to find regimes in a climatic time series where each regime has a different linear trend. An appropriate cost function was defined and using Tikhonov regularization and Galerkin discretization, the cost function is converted to a familiar linear quadratic problem. There is a trade-of between number of Finite Elements Basis Function, volume of computation and consequently accuracy. Also, the regularization parameter can change the length of detected regimes. By trial and error, an optimal number of regimes can be estimated. A climatic time series of North Carolina is analyzed by this method. The results represent spatiotemporal pattern of climate change corresponding to areas of studies.

Acknowledgement. This work partially supported by the Expeditions in Computing by the National Science Foundation under Award Number: CCF-1029731. For additional information, contact corresponding author, Dr. Homaifar (homaifar@ncat.edu). With special thanks to Dr. Manoj Jha and Dr. Stefan Liess.

References

1. Mudelsee, M.: Climate Time Series Analysis. Classical Statistical and Bootstrap Methods. Springer (2010)
2. Kremer, H., Günnemann, S., Seidl, T.: Detecting Climate Change in Multivariate Time Series Data by Novel Clustering and Cluster Tracing Techniques. In: IEEE International Conference on Data Mining Workshops, San Jose (2010)
3. Horenko, I.: Finite element approach to clustering of multidimensional time series. SIAM Journal on Scientific Computing 32(1), 62–83 (2010)
4. Qi, Y., Paisely, J.W., Carin, L.: Music Analysis Using Hidden Markov Mixture Models. IEEE Transactions on Signal Processing 55(11), 5209–5224 (2007)
5. Meyer, F.G., Xilin, S.: Classification of fMRI Time Series in a Low-Dimensional Subspace with a Spatial Prior. IEEE Transactions on Medical Imaging 27(1), 87–98 (2007)

6. Kelly, J., Hedengren, J.: A Steady-State Detection (SSD) Algorithm to Detect NonStationary Drifts in Processes. Journal of Process Control 23(3), 326–331 (2013)
7. Metzner, P., Horenko, I.: Analysis of Persistent Non-stationary Time series and Applications. Communications in Applied Mathematics and Computational Science 7(2), 175–229 (2012)
8. Aster, R.: Parameter Estimation and Inverse Problems. Academic Press (2013)
9. Zienkiewicz, O.: The Finite Element Method Its Basis and Fundamentals. Elsevier (2005)
10. Modarres, R., Sarhadi, A.: Rainfall Trends Analysis of Iran in the Last Half of the Twentieth Century. Journal of Geophysical Research 114(D3) (2009)

A Generalized Fuzzy T-norm Formulation of Fuzzy Modularity for Community Detection in Social Networks

Jianhai Su[1] and Timothy C. Havens[1,2]

[1] Department of Computer Science
[2] Department of Electrical and Computer Engineering
Michigan Technological University
Houghton, MI, USA
{jianhais,thavens}@mtu.edu

Abstract. Fuzzy community detection in social networks has caught researchers' attention because, in most real world networks, the vertices (i.e., people) do not belong to only one community. Our recent work on generalized modularity motivated us to introduce a generalized fuzzy t-norm formulation of fuzzy modularity. We investigated four fuzzy t-norm operators, Product, Drastic, Lukasiewicz and Minimum, and the generalized Yager operator, with five well-known social network data sets. The experiments show that the Yager operator with a proper parameter value performs better than the product operator in revealing community structure: (1) the Yager operator can provide a more certain visualization of the number of communities for simple networks; (2) it can find a relatively small-sized community in a flat network; (3) it can detect communities in networks with hierarchical structures; and (4) it can uncover several reasonable covers in a complicated network. These findings lead us to believe that the Yager operator can play a big role in fuzzy community detection. Our future work is to build a theoretical relation between the Yager operator and different types of networks.

1 Introduction

Community detection in graphs has a long history. It can be widely applied to many modern complex networks, such as biological networks, social networks, information networks, to better understand how vertices function in those networks and the possible patterns in the networks. Modularity, introduced by Newman and Girvan [1], triggered a vast array of work on modularity-based community detection. Our recent work [2] on generalizing Newman and Girvan's modularity inspires our proposal to replace the product operator in modularity with a generalized t-norm operator. Following this direction, we give the conjecture that different operators might be able to detect different types of communities and demonstrate this by doing experiments on five commonly used test data sets. Our results show that the product operator is not the best choice for all networks, and that the Yager operator [3] with a proper parameter choice gives a better result.

The rest of paper is organized as follows. Section 2 talks about fuzzy communities and recent related research; section 3 starts from Girvan and Newman's modularity, and then derives the generalized modularity in detail. Section 4 shows how to extend the

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_7,

fuzzy modularity with a fuzzy t-norm to interpret the multiplication of two selection variables, and also introduces basic fuzzy t-norm operators, such as the Yager operator. Experiments are done in section 5 and section 6 summarizes.

2 Fuzzy Communities

Community structure or cliques are a property of many graph-based networks. They can provide invaluable information on important trends in networks; thus, community structure can provide efficient solutions for hard problems. For example, in a network of on-line shopping markets, a proper grouping of customers with similar interests can improve on-line recommendation systems and bring more business opportunities to retailers. Technically, cliques are groups of vertices in a network that have denser connections with each other than they do with the rest of the network [1]. Traditionally, community structure detection is called a partitioning [4]. That is, each vertex either belongs to a community or not. We call these types of partition, *crisp* partitions.

Crisp partitions of a network cannot represent community structure well in some real networks, especially those that are (similar to) social networks. Generally speaking, one person plays different roles in a society and thus could link to many kinds of communities, such as school, friend circle, and groups based on similar interest. So crisp partitions would not uncover important people who play the role of intermediation between different communities. From this consideration, the notion of fuzzy community arises. The detection of fuzzy communities in a network results into a cover [4]. In a cover, the belongingness of a vertex to a community is valued as a degree of truth, of the extent that a vertex belongs to a community.

Based on the idea that vertices in one community are more likely to form a clique, Palla et al. [5] came up with the Clique Percolation Method, one of the most popular methods used in detecting overlapping communities. They define the term, k-clique, as a complete graph with k vertices. The k-clique communities that are found can share vertices, but the method fails to reach some vertices, like vertices with degree one. Zhang et al. [6] proposed another method for detecting overlapping communities, which can be summarized as three steps: spectral mapping, fuzzy clustering, and optimization of a quality function for each cover. Liu [7] presented another quality function (which is proven to be equivalent to Zhang's in [2]) to find fuzzy communities under a simulated annealing strategy. Havens et al. [2] proposed a generalized modularity-based quality function for fuzzy community detection that shows better performance than both Zhang and Liu's functions on the well-known Karate Club data set [8] and American College Football data set [1]. Havens also provides an analytical argument that his generalized modularity is theoretically valid, while Zhang and Lui's modularities are not.

Usually, in community detection, a network is represented by a graph $G = (V, E, W)$, where V is the set of vertices in the network, E is the set of existing edges in the network, and W is an $n \times n$ matrix, where n is the number of vertices in G. Every entry w_{ij} in W indicates the edge-weight between vertex i and vertex j. The graph may be directed or not, weighted or not. Here, we focus on undirected and weighted graphs, meaning that W is symmetric.

3 Modularity

In [2, 6, 7], different quality functions, also named modularity, are used to optimally select a cover for a network in a cover space that is established by detecting covers along a given range of community numbers using the *fuzzy c-means* (FCM) clustering method [9]. Every cover generated by FCM is actually represented by a $c \times n$ matrix U, where c is the number of communities; every entry u_{ij} denotes the membership of vertex j in community i. Thus, any cover of a network with a given c is an element of a set called a fuzzy partition set, denoted by M_{fcn} [2],

$$M_{fcn} = \{U \in \mathcal{R}^{c\times n}; 0 \leq u_{ij} \leq 1, \forall i, j; 0 < \sum_{j=1}^{n} u_{ij} < n, \forall i; \sum_{i=1}^{c} u_{ij} = 1, \forall j\}. \quad (1)$$

Modularity was introduced by Newman and Girvan and is used to evaluate the goodness of the results of community detection. The higher the value of the modularity is, the better quality the partition/cover is. Modularity is built on the notion of a null model that assumes that no community structure exits in a random graph [1]. So the degree of an existence of structure in a community can be measured by comparing actual edge density of a detected community to the expected edges density of the same community in the null model. By summing over all communities, modularity judges the validity (or quality) of the partition or cover. For a crisp partition of an undirected graph, the Newman and Girvan modularity is defined as:

$$Q = \sum_{i=1}^{c} \left(e_{ii} - a_i^2\right), \quad (2)$$

where e_{ii} is the fraction of all edges in the graph that are in community i, and a_i is defined as the fraction of edges that begin or end in community i. An extended modularity for a weighted and undirected graph is given in [6], as:

$$Q = \frac{1}{\|W\|} \sum_{k=1}^{c} \sum_{i,j\in V_k} \left(\omega_{ij} - \frac{m_i m_j}{\|W\|}\right), \quad (3)$$

where V_i is the set of the vertices that are in the ith community, m_i is the degree of vertex i (or $m_i = \sum_j w_{ij}$), and $\|W\|$ is the sum of all the elements in W. By introducing entries of the $c \times n$ partition matrix U as selection variables, we can go one step further and produce the generalized modularity, which can work for any type of graph partition (crisp, fuzzy, probabilistic, or possibilistic). The generalized modularity of U w.r.t $G = (V, E, W)$ is [2]

$$Q_g = \frac{tr(UBU^T)}{\|W\|}, U \in M_{pcn}, \quad (4)$$

where M_{pcn} is the set of all possibilistic partitions. The generalized modularity at (4) more explicitly shows the role of U in the calculation of modularity.

4 Generalized Fuzzy T-norm Modularity

By extending Q_g further, we can see how selection variables play a part in modularity.

$$Q = \frac{1}{\|W\|} \sum_{k=1}^{c} \sum_{i,j \in V_k} \left(\omega_{ij} - \frac{m_i m_j}{\|W\|} \right) u_{ki} u_{kj}, \tag{5}$$

where u_{ki} and u_{kj} are the membership of vertex i and vertex j in the kth community respectively. So, in equation (5), the multiplication of u_{ki} and u_{kj} indicates the degree of truth to which both vertex i and vertex j are in the kth community. The multiplication here can be interpreted as a fuzzy t-norm, namely the product.

Let $C = \{1, \ldots, c\}$ be the set of community indices, then u_{ki} and u_{kj} are the memberships of vertices i and j in the kth community, respectively. We can extend this notation and say that $u_i(k)$ and $u_j(k)$ are the associated membership functions (actually the fuzzy set of communities for each vertex), where

$$u_i = \{u_i(k) | 0 \leq u_i(k) \leq 1, \quad k \in C\}, \tag{6}$$

$$u_j = \{u_j(k) | 0 \leq u_j(k) \leq 1, \quad k \in C\}. \tag{7}$$

Then, the intersection of fuzzy sets u_i and u_j, denoted by B, can be written as [10]

$$B = (u_i \cap u_j)(k), \quad k \in C, \tag{8}$$

where $(u_i \cap u_j)(.)$ is the membership of the communities in the intersection of u_i and u_j or

$$(u_i \cap u_j)(k) = i(u_i(k), u_j(k)), \quad k \in C. \tag{9}$$

In generalized modularity, the binary operator $i(,)$ is a product (or can generalized thereof as a product). However, because of our interpretation that u is a fuzzy set, then this product can be generalized to any fuzzy t-norm. Other than product, there are three other common fuzzy t-norm operators: drastic, Lukasiewicz, and minimum. The flexibility of fuzzy t-norm operators imply that product might not be good for all problems and a better operator might exist for a given type of community. Thus, we extend the generalized modularity at (5) to

$$Q_i = \frac{1}{||W||} \sum_{k=1}^{c} \sum_{i,j \in V_k} \left(w_{ij} - \frac{m_i m_j}{||W||} \right) i(u_{ki}, u_{kj}). \tag{10}$$

We examine this possibility in the context of Q_g by a comparison of generalized modularity (viz., modularity using the product t-norm) and the fuzzy t-norm modularity Q_i using the Yager t-norm operator [3]

$$i_Y(x, y) = \max(0, 1 - ((1-x)^p + (1-y)^p)^{\frac{1}{p}}), \quad p \geq 0. \tag{11}$$

For $p = 0$, Yager reduces to drastic; for $p = 1$, Yager becomes Lukasiewicz; and, for $p = +\infty$, Yager represents minimum. We now turn to experiments that show how the Yager t-norm can produce superior community detection results for several social network data sets.

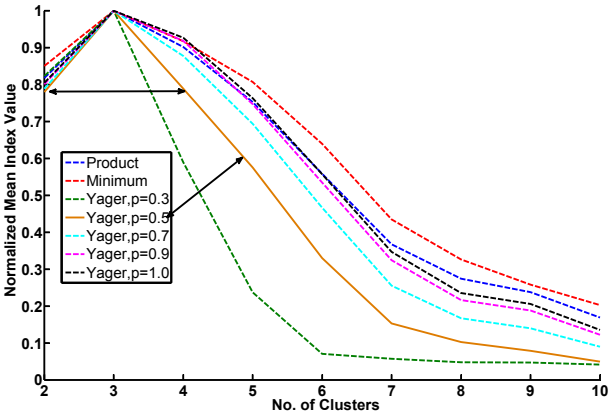
5 Experiments

To demonstrate our generalized fuzzy t-norm modularity, we show how modularity can be used to detect the number of communities in real social networks. We show the difference in performance by using Q_i by adjusting the parameter p in the Yager operator for five commonly used real networks: Karate Club [8], Sawmill [11], Dolphin Network [12], Political Books [13] and American College Football [1]. For each data set, five different values of p were used. We compared Yager-t-norm Q_i with the varying p parameter to the generalized modularity at (5) and Q_i using the min fuzzy t-norm. The basic experiment environment setting is the same as that in [2]: MULTICUT spectral clustering with FCM is used to partition the graph; the FCM fuzzifier is 2; 100 trials are carried out on each data set for each value c in a range of different community numbers (nominally $[2, 3, \ldots, 10]$). Two visualizations are produced to show the quality of detection result. One visualization plots the normalized average modularity value (over the 100 trials) versus community number c. This visualization shows the number of communities by the peak of the graph. The other visualization shows an image of the number of trials (out of 100) in which each operator maximized for a given number of communities c. This shows the number of communities by white blocks in the image.

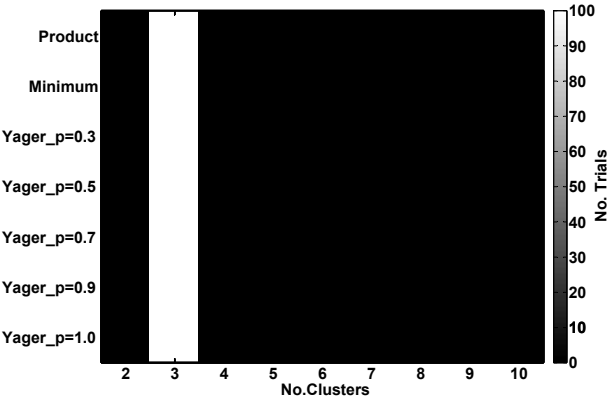
Karate Club Network, collected by Zachary, is a well-known benchmark used to evaluate community detection algorithms. It has 34 vertices and the weight of an edge indicates the strength of association of two club members who interact with each other outside of the club environment. A grouping, suggested by Zachary, is to partition the club around the president and the instructor, separating into two clusters. But Newman and Girvan argue that three communities exist in the club. The third community contains five members who only have connection with the instructor in the club, and not the president. In our experiment, this argument is confirmed. Figure 1(a) demonstrates that the Yager t-norm Q_i with $p = 0.5$ shows a stronger peak at $c = 3$ than the other modularity formulations, which suggests that the fuzzy t-norm modularity is effective at improving the confidence of the validity of the $c = 3$ community partition. Figure 1(b) shows that each modularity index strongly prefers the notion that there are $c = 3$ communities in the Karate Club network, which is expected.

Sawmill is a communication network within a small enterprise. All employees were asked to rate his or her contact with every other person in a week as a score on a five-point scale. Two employees were linked in the network if they rated their contact no less than 3 each. The number of employees or vertices in Sawmill is 36. The suggested number of communities is 4, since there are two major sections, the mill and the planner sections, and two languages are used, English and Spanish. As shown in Fig. 2(a), Yager operator with the parameter of $p = 1.0$ gives the sharpest peak at $c = 4$. This is especially noteworthy because the other curves are not as explicit in their choice of $c = 4$; this is most noticeable in the Minimum and Yager $p = 10$ indices, which also seem to suggest that $c = 3$ is a possibility.

Dolphin Network is a social network of 62 bottle-nose dolphins living in Doubtful Sound, New Zealand. These data were collected by Lusseau from 7 years of observation and each edge of the network represents the presence or absence of a statistically strong association between each two pair of dolphins. The network can be divided into two large communities in term of the age of dolphins. Within the young community,



(a) Normalized average modularity

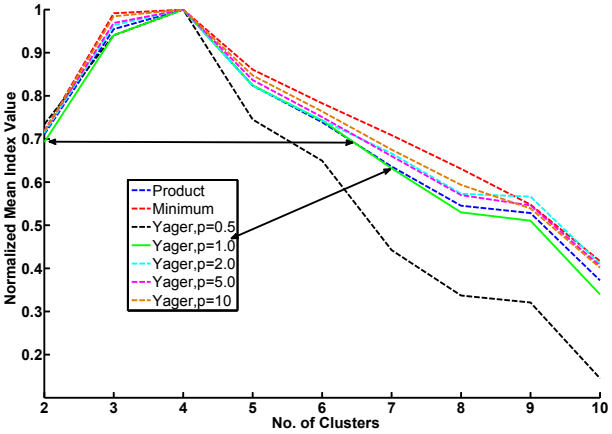


(b) Choice of community number

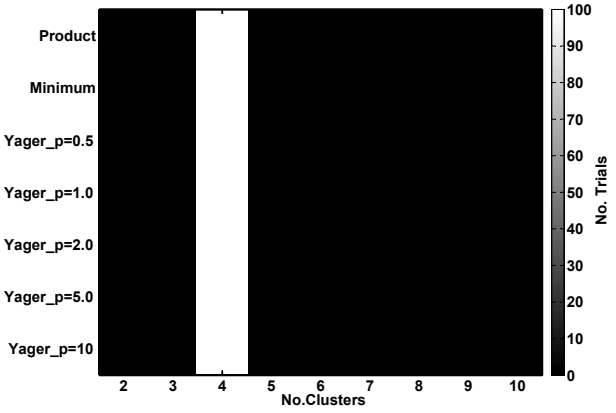
Fig. 1. Karate Club Data

dolphins may also be clustered into three sub-communities. One of them contains all female dolphins. So, for a coarse detection, the suggested number of communities is 2; for a finer-grained detection, the suggested number of communities is 4. As shown in Fig. 3, the product operator (or standard generalized modularity) and the Yager t-norm Q_i with $p = 20$ prefers the coarse partition of $c = 2$, but the Yager t-norm Q_i with $p \geq 50$ finds $c = 4$ communities. Interestingly, the min t-norm Q_i also finds $c = 4$. This experiment demonstrates that the flexibility of the Yager t-norm allows us to find different community configurations in a hierarchical network.

Political Books is a network of 105 books on American politics, complied by Valdis Krebs and recreated and published by Mark Newman. Nodes represent political books sold by the on-line bookseller, Amazon.com. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also



(a) Normalized average modularity

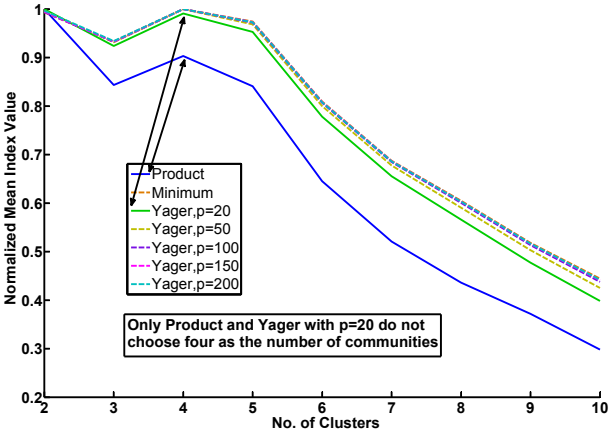


(b) Choice of community number

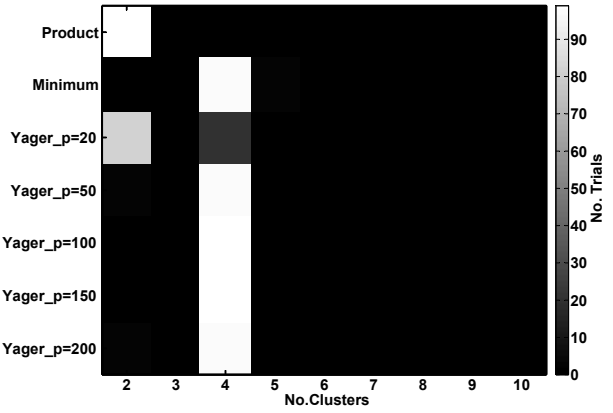
Fig. 2. Sawmill Data

bought these other books" feature on Amazon. So, finding communities in this network is to find groups of people who have similar taste in political books. After reading the descriptions and reviews of these books posted on Amazon, Newman marked each book as liberal, neutral or conservative. The number of books in the neutral community is much smaller than that in each of the other two communities. The suggested number of communities is three. As seen in Fig. 4, the product t-norm (or generalized modularity) suggest there are only $c = 2$ communities, liberal and conservative communities, while the Yager t-norm with $p = 300$ and $p = 500$ prefers the suggested $c = 3$ communities.

Americal College Football is a network of competing relations of 115 Division I college football teams during the 2000 regular season. If two teams played against each other, the weight on the edge connecting the corresponding two vertices is 1; otherwise,



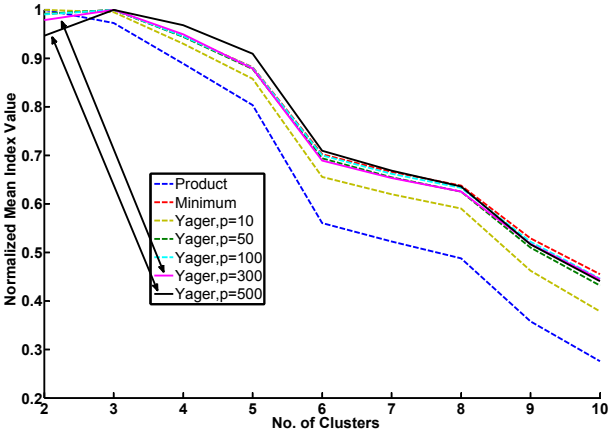
(a) Normalized average modularity



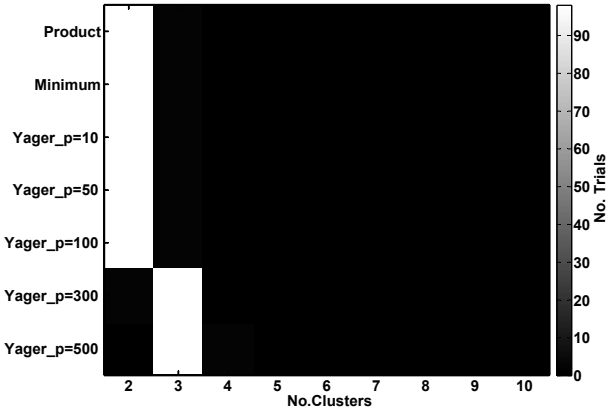
(b) Choice of community number

Fig. 3. Dolphin Network Data

it is 0. The 115 teams belong to 11 conferences; teams in one conference usually play more games against teams within the same conference. Hence, a naturally suggested number of communities is 11. But it has also been suggested that $c = 10$ is an appropriate choice as there is one conference that plays teams from several other conferences (it is a small conference). As shown in Fig. 5, all the indices prefer 10 communities, but the Yager t-norm Q_i with $p = 2.0$ gives the highest value of modularity among all operators at $c = 11$. So, it is more likely to infer that 11 community is also a reasonable result.

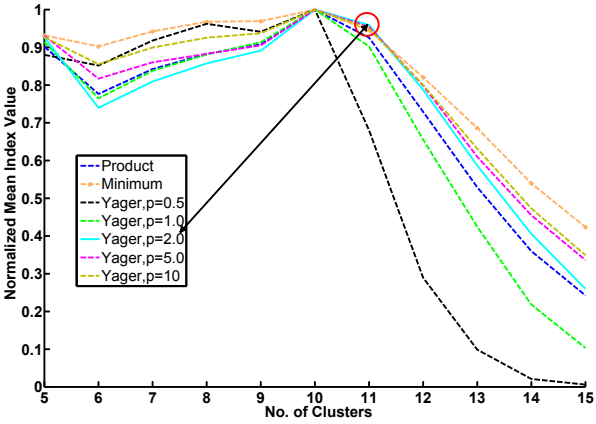


(a) Normalized average modularity

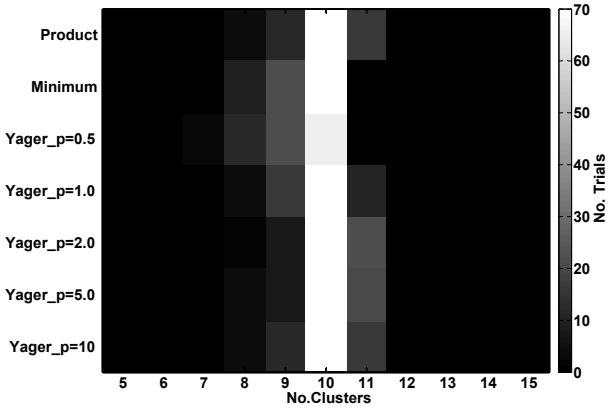


(b) Choice of community number

Fig. 4. Political Books Data



(a) Normalized average modularity



(b) Choice of community number

Fig. 5. American College Football Data

6 Conclusion

Inspired by generalized modularity, we adapted the product of two selection variables (which become fuzzy for the case of fuzzy community detection) as a t-norm to produce a generalized fuzzy t-norm modularity. Our conjecture was that the product t-norm may not be the best choice for community detection in all networks, and that a flexible fuzzy t-norm operator could allow better community detection results. To verify our conjecture, we adopted the Yager t-norm, one of several 1-parametric t-norms. The results of our experiment lead to four preliminary conclusions and confirm our conjecture.

1. For simple networks, like Karate Club and Sawmill, the Yager t-norm modularity with a proper parameter value gives a sharper visualization of the number of communities when compared with the product-based modularity;
2. For networks composed of both relatively small- and large-sized communities, e.g., the Political Books network, the product-based modularity can fail to detect the small communities, but the Yager t-norm modularity is able to detect them;
3. For networks with hierarchical structures, like the Dolphin Network, the product-based modularity is only able to uncover a coarse community structure, while the Yager-based modularity with varied parameter settings detects the multiple levels of community structure;
4. For complicated networks, like the American College Football network, the Yager t-norm modularity can provide more information to infer the complicated community structure.

Although these conclusions are based on only a small sample of data sets, we can conclude that the product t-norm is *not* an all-encompassing best choice for all networks. And we can see that different types of communities could be revealed more easily by using our generalized fuzzy t-norm modularity. Furthermore, our empirical results suggest two open questions:1) how to optimally determine the Yager parameter value in community detection? And 2) what kind of parametric t-norm is the best for community detection of a given network type or configuration? In the future, addressing the first question, we will examine methods by which Yager's t-norm parameter can be chosen automatically to search for different types of communities in a network, e.g., by some computational intelligence method. For the second question, a possible way to start an investigation is to analyze how the characteristics of other parametric t-norms can affect the community detection of a given network.

Acknowledgements. We thank Superior High Performance Computing at Michigan Technological University for their support.

References

1. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proceedings of the National Academy of Sciences of the United States of America, 7821–7826 (2002)

2. Havens, T.C., Bezdek, J.C., Leckie, C., Ramamohanarao, K., Palaniswami, M.: A soft modularity function for detecting fuzzy communities in social network (2012)
3. Yager, R.R.: On the general class of fuzzy connectives. Fuzzy Sets and Systems 4, 235–242 (1980)
4. Fortunato, S.: Community detection in graphs. Physics Reports, vol. 486, pp. 75–174 (2010)
5. Palla, G., Dernyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435, 814–818 (2005)
6. Zhang, S., Wang, R., Zhang, X.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physics A: Statistical Mechanics and its Applications 374, 483–490 (2007)
7. Liu, J.: Fuzzy modularity and fuzzy community structure in networks. The European Physical Journal B 77(4), 547–557 (2010)
8. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)
9. Bezdek, J.C., Keller, J., Krisnapuram, R., Pal, N.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publishers, Norwell (1999)
10. Zadeh, L.A.: Fuzzy sets. Information and Control 8(3), 338–353 (1965)
11. Michael, J.H., Massey, J.G.: Modeling the communication network in a sawmill. Forest Products 47, 25–30 (1997)
12. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. Behavioral Ecology and Sociobiology 54, 396–405 (2003)
13. Krebs, V.: Books about U.S.A. politics, `http://www.orgnet.com/`

Soft Computing Models in Online Real Estate

Jozo Dujmović[1], Guy De Tré[2], Navchetan Singh[1], Daniel Tomasevich[1], and Ryoichi Yokoohji[1]

[1] Department of Computer Science, San Francisco State University, San Francisco, CA, USA
jozo@sfsu.edu
[2] Dept. of Telecommunications and Information Processing, Ghent University, Ghent, Belgium
Guy.DeTre@telin.ugent.be

Abstract. In this paper we present a decision support system that uses soft computing models for evaluation, selection and pricing of homes. The system (called LSPhome) is based on the Logic Scoring of Preference (LSP) evaluation method and implemented in the context of online real estate. The goal of this system is to use weighted compensative logic models that can precisely express user needs, and help both buyers and sellers of homes. The design of such a system creates specific logic and computational challenges. Soft computing logic problems include the use of verbalized importance scales for derivation of andness, penalty-controlled missingness-tolerant logic aggregation, detailed and verbalized presentation of evaluation results, and development of optimum pricing models. Computational problems include fast and parallel collection of heterogeneous information from the Internet, and development of user interface for fast and simple creation of customized soft computing decision criteria by nonprofessional decision makers.

Keywords: Evaluation, selection, real estate, missing data, verbalization.

1 Introduction

Real estate is an area that includes a spectrum of soft computing decision problems. In this paper we present a survey of the most important soft computing models that are used in online real estate (ORE). The first such a problem is the development of criteria for evaluation and selection of homes. The home evaluation criteria are based on weighted compensative logic functions that can model adjustable degrees of simultaneity and replaceability, mandatory, sufficient, and optional requirements, as well as adjustable degrees of importance of various home attributes. The aggregation of home quality and home affordability is also a soft computing logic problem. Similarly, the problem of optimum home pricing can also be solved using soft computing models. In ORE we frequently encounter problems of decision making with incomplete (missing) inputs, and the need to expand aggregation models with missingness-tolerant aggregators. Finally, the users of ORE decision models are not decision experts, but nonprofessionals who need simple verbalized approach to specifying soft computing decision models. These seemingly heterogeneous problems are closely related in the

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_8,

context of ORE. Thus, the goal of this paper is to show all fundamental components of the soft computing decision infrastructure in ORE.

In the USA the real estate market data and procedures are governed by the National Association of Realtors [11]. Full information about homes on sale and other marketed properties can be found in the Multiple Listing Service (MLS) [14]. ORE web sites (e.g. [13],[16]) use MLS data and provide application programming interfaces (API) that can be used to access data about available homes and their characteristics. These data can be used as inputs for evaluation and selection process based on soft computing criteria.

The paper is organized in three main sections. Section 2 describes soft computing models for home evaluation in the context of buying and selling a home. Section 3 surveys the penalty-controlled missingness-tolerant aggregation, and the verbalization problems. Section 4 presents experimental results generated by the LSPhome system.

2 LSP criterion Function for Home Evaluation

The LSP method [5] provides soft computing evaluation criteria built in three basic steps. The first step develops a list of attributes $a_1,...,a_n$, $a_i \in R$, $i=1,...,n$ that characterize relevant properties of evaluated homes. The second step is to provide requirements for each attribute in the form of elementary criteria functions $g_i : R \to I$, $I=[0,1]$; they assign degrees of satisfaction to attribute values $x_i = g_i(a_i)$, $i=1,...,n$. The third step generates an overall degree of satisfaction (overall suitability) as an aggregate of attributes' degrees of satisfaction: $S = L(x_1,...,x_n)$. The mapping $L: I^n \to I$ is based on weighted compensative logic functions [1],[10],[3],[12] that are implemented as specific forms of means [9], [2].

2.1 Attribute Tree

The home evaluation attribute tree based on data that can be retrieved from the Internet is shown in Fig. 1. The attributes are grouped in two main groups: the quality of home location, and the quality of the home. The quality of home location is based on an analysis of points of interest that are available from Google. The attributes that affect the home quality come from ORE web sites.

2.2 Elementary Criteria

The number of home evaluation attributes in the attribute tree in Fig 1. is 36. For each of these attributes we provide an attribute criterion function that reflects the requirements of a specific user. Some of attribute criteria are specific for each user and others can be shared by all users. Two such examples are shown in Fig. 2. The criterion #112 uses data obtained from the LSPhome user interface shown in Fig.3.

```
1 HOME SUITABILITY

   11 QUALITY OF LOCATION
      111 Suitability of neighborhood
         1111 Walkability
         1112 Shopping and dining
         1113 Health support
         1114 Suitability for children
         1115 Suitability for seniors
      112 Walking distance from the ideal location

   12 QUALITY OF HOME
      121 Available space
         1211 Area belonging to home
            12111 Total internal living area of home
            12112 Outer usable area belonging to home
         1212 Rooms and other designated areas
            12121 Primary rooms
               121211 Number of bedrooms
               121212 Number of bathrooms
               121213 Kitchen
               121214 Dining room/area
               121215 Living/family room
            12122 Additional space and storage
               121221 Additional space
                  1212211 Breakfast room/area
                  1212212 Home office
                  1212213 Laundry
               121222 Storage and auxiliary areas
                  1212221 Walk-in closets
                  1212222 Pantry
                  1212223 Auxiliary utility areas
         1213 Parking space
            12131 Reserved parking
               121311 Garage
                  1213111 Private garage
                  1213112 Shared garage
               121312 Reserved uncovered parking space
            12132 Public parking (first-come, first-served)
               121321 Free public parking
                  1213211 Street parking next to home
                  1213212 Street parking close to home
               121322 Paid public parking
      122 Home features
         1221 Home organization/layout
            12211 Type of home
            12212 Number of floors
         1222 Home construction features
            12221 External wall material
            12222 Type of floor
            12223 Type of roof
         1223 Home energy supply
         1224 Home temperature regulation
            12241 Source of energy for heating
            12242 Type of heating system
            12243 Type of cooling system
         1225 Home age and maintenance
            12251 Home age
            12252 Last modification/improvement
```

Fig. 1. The home evaluation attribute tree

The presented interface provides a limited capability for homebuyers to specify their requirements. This is necessary to avoid too much detail that would discourage the majority of general population users. In all cases the users are expected to specify the ideal location of their desired home and the maximum allowable distance $D_{\max}$ from the ideal location. The evaluation of homes using the attribute criterion #112 (Fig. 2) is based on the relative distance $100D/D_{\max}$. The presented attribute criterion shows a relatively high tolerance for all distances except those close to $D_{\max}$. By selecting $D_{\max}$ (see Fig. 3) the users can customize the attribute criterion function.

112		Walking distance from the ideal location
Value	%	The ideal location is a user-specified location selected as a point that completely satisfies all user requirements. The distance can be expressed as (1) walking, (2) car, (3) public transport, or (4) bicycle distance. We use the normalized relative walking distance x = 100D/Dmax, where D = walking distance between an evaluated home and the ideal location (miles or km) Dmax = The maximum acceptable walking distance from the ideal location (miles or km). *Dmax must be selected by each user*
0	100	
40	86	
70	70	
90	50.6	
100	0	
12222		**Type of floor**
Value	%	The type or material of the walking surface of the primary living areas of the home. The main options are: ST = stone HW = hardwood SW = softwood L = laminate floor V = vinyl/linoleum P = parquet SL = slate T = tile (ceramic) C = carpet Evaluation method: 1 = ST/SL/T, 2 = V, 3 = SW/C, 4 = L, 5 = HW, 6 = P
1	35	
2	50	
3	70	
4	75	
5	85	
6	100	

Fig. 2. Sample elementary criteria

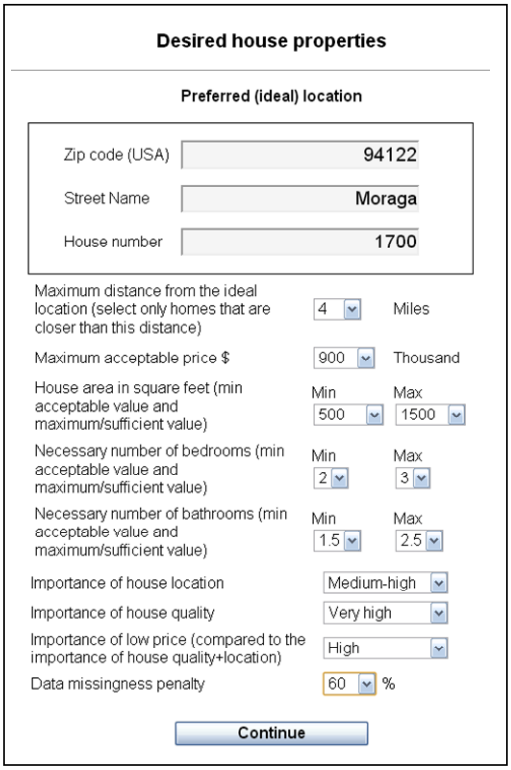


Fig. 3. LSPhome interface for specifying user requirements

Other user supplied elementary criteria are the available area, the number of bedrooms and the number of bathrooms. All of them are specified in the range from the minimum acceptable value $a_{\min}$ to the maximum (sufficient) value $a_{\max}$. The simplest form of such elementary criteria is the following: $x = g(a) = \min[1, (a / a_{\max})]$.

An alternative more flexible version can be obtained by assigning the minimum default suitability $x_{\min}$ to the minimum acceptable value $a_{\min}$ as follows:

$$x = g(a) = \begin{cases} 0, & a < a_{\min} \\ \min\left(1, \left(a - a_{\min} + x_{\min}(a_{\max} - a)\right)/\left(a_{\max} - a_{\min}\right)\right), & a \geq a_{\min} \end{cases}$$

To simplify the use of LSPhome, only the essential user requirements are customizable. All user-shareable and less specific elementary criteria are not customizable and one such example is the criterion #12222 shown in Fig.2. That criterion is a fixed scoring system that reflects an average standpoint acceptable for the majority of users. E.g., if the ORE web site provides a home with hardwood floor, then, for all homebuyers, the corresponding floor satisfaction degree is 85%. The use of fixed elementary criteria significantly reduces the number of necessary user inputs and simplifies the communication with users.

2.3 Logic Aggregation Structure

Aggregation of all attribute suitability degrees yields the overall suitability of the evaluated home. The aggregation is based on the superposition of several basic aggregators that are implemented using the generalized conjunction/disjunction function (GCD) [4]. The soft computing *suitability aggregation structure* (SAS), in the form of a "shade diagram," [5] is shown in Figs. 4 and 5. The suitability aggregation structure uses a spectrum of weighted compensative logic functions. In the case of GCD we use the system of 17 distinct degrees of orness $\omega = 0, 1/16, ..., 1$, (or andness $\alpha = 1 - \omega$) symbolically denoted C, C++, C+, C+-, CA, C-+, C-, C--, A, D--, D-, D-+, DA, D+-, D+, D++, D, described in [3]. The aggregators starting with letter C denote various forms of conjunction (pure and hard or soft partial) and aggregators starting with letter D denote various forms of disjunction (pure and hard or soft partial) [4]. The hard partial conjunction function is a model of mandatory requirements ($f_C(x_1, ..., x_k) = 0$, $x_i = 0$, $i \in \{1, ..., k\}$, $k > 1$) and the hard partial disjunction is a model of sufficient requirements ($f_D(x_1, ..., x_k) = 1$, $x_i = 1$, $i \in \{1, ..., k\}$, $k > 1$). Soft versions provide a positive output if a single input is positive. The aggregator A denotes the neutrality (the arithmetic mean). E.g., to evaluate the suitability of neighborhood (#111) we first identify the locations of all relevant points of interest for evaluation of walkability, shopping and dining, health support, and suitability for children and seniors, and then we aggregate these suitability degrees using a weighted soft partial conjunction C- (andness $\alpha = 5/8$), with the highest relative importance (weight) assigned to walkability and to suitability for children. Such weights reflect the fact that most homebuyers are young families. In Figs. 4 and 5 "+" in the first column denotes mandatory attributes and "-" denotes optional attributes.

The SAS shown in Fig. 4 includes a user-supplied final aggregator F of location suitability and home quality. The user is requested to specify in verbal form the overall importance of location suitability and house quality (Fig. 3) and the parameters of the resulting GCD aggregator are then computed as shown in Section 3 and [7]. The resulting overall suitability scores for N competitive homes are S_i, $i = 1, ..., N$.

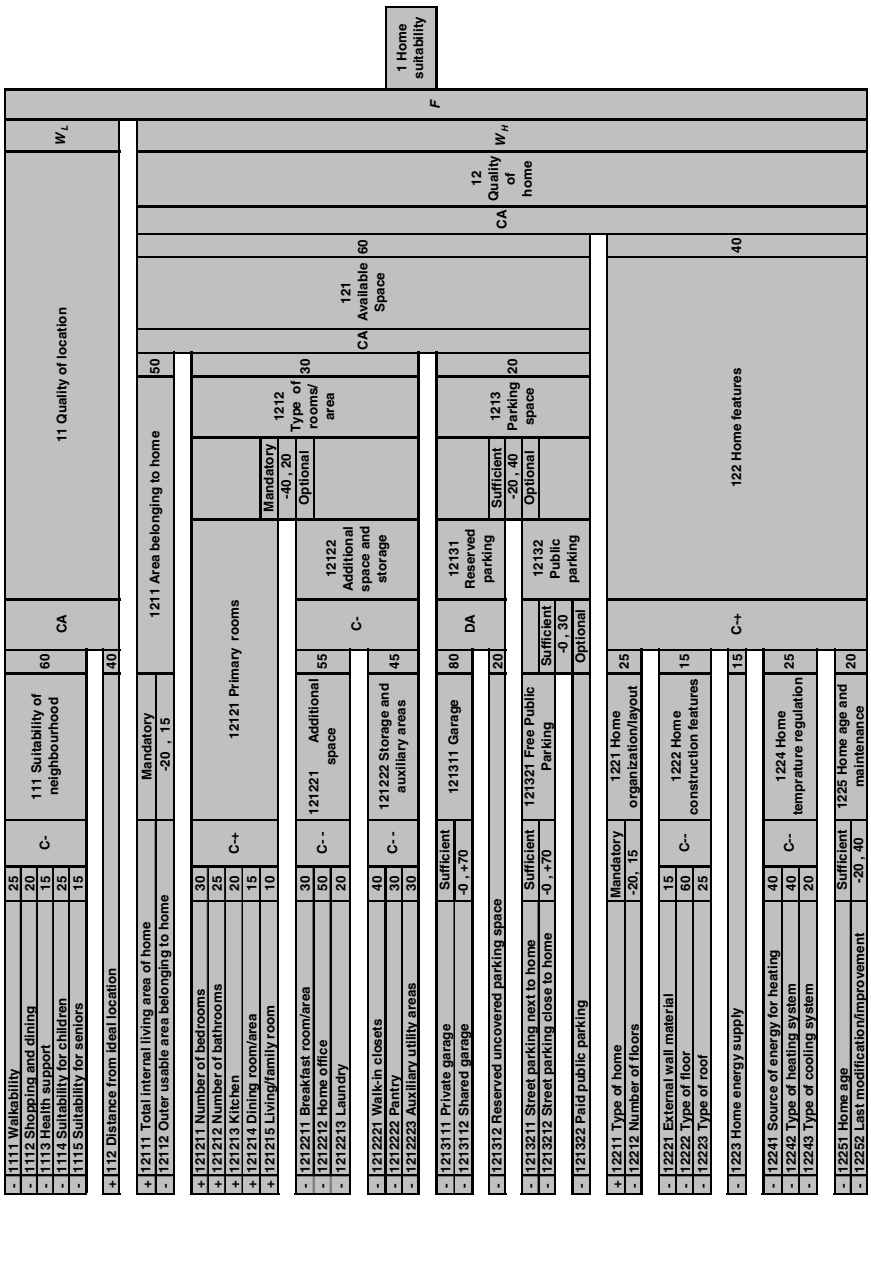


Fig. 4. The home criterion suitability aggregation structure (SAS)

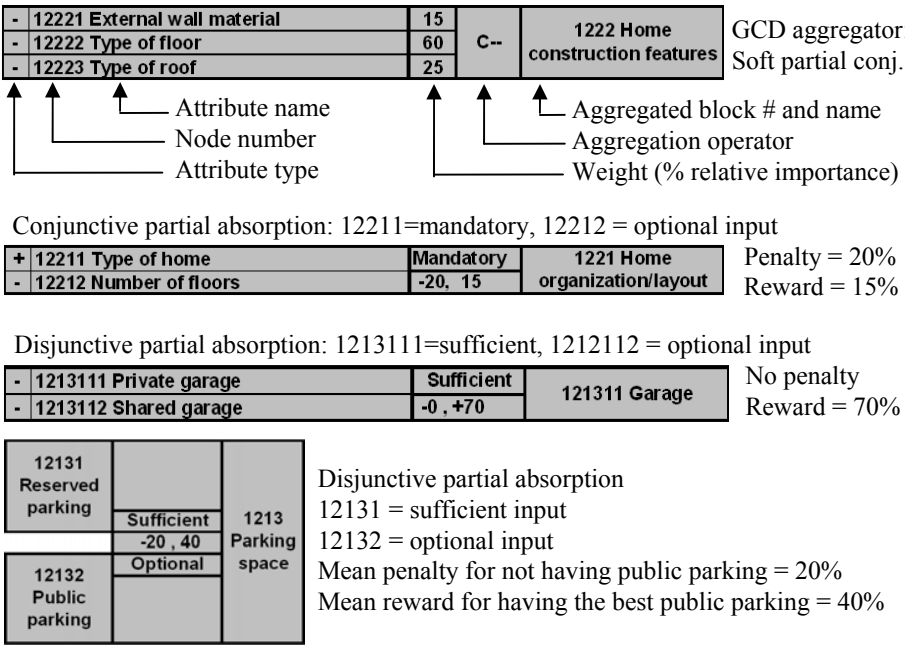


Fig. 5. Explanation of fields in the shade diagram

We also express the soft computing logic relationships by using the conjunctive partial absorption aggregators that aggregate mandatory and optional inputs and the disjunctive partial absorption aggregators that aggregate sufficient and optional inputs [3],[5]. In both cases the properties of these aggregators are determined using the desired level of penalty (decrease of output in the case of unsatisfied optional input) and reward (increase of output in the case of perfectly satisfied optional input). E.g., in the case of parking space (#1213) it is sufficient to have a reserved parking place and the availability of public parking is optional with the mean penalty of -20% and the mean reward of +40%. An obvious advantage of shade diagrams is their rectangular form: similarly to Nassi-Shneiderman structured flow charts, a new shade diagram can be inserted in each rectangular space, making easily readable aggregation structures. Shading of diagrams facilitates the perception of grouping of inputs.

If home costs are $C_1,...,C_N$ then logic aggregation also includes a hard partial conjunction ($\bar{\Delta}$) for aggregating the overall home quality $Q_i = S_i / \max(S_1,...,S_N) \in [0,1]$ and the home affordability $A_i = \min(C_1,...,C_N)/C_i \in [0,1]$ yielding (in the case of equal weights) the overall home values $V_i = Q_i \,\bar{\Delta}\, A_i,\ i = 1,...,N$. In the case of home sale this model can be used to find the *maximum price of our home* C_i^*, so that (even with that price) the home is still the most attractive in a selected area (attains the maximum value $V_i = Q_i \,\bar{\Delta} \min(C_1,...,C_N)/C_i^* \geq Q_j \,\bar{\Delta} \min(C_1,...,C_N)/C_j, \quad \forall j \neq i$).

3 Missingness-Tolerant Aggregation and Verbalization

The LSP criterion function consists of attribute criteria and the suitability aggregation structure and assumes the availability of all n input attribute values. In reality, however, the ORE web sites regularly offer incomplete data about available homes. For example, our experiments with homes available through ORE API in San Francisco show on the average the availability in the range from 50% to 70% of input attributes, as illustrated in Fig. 6. For each of ten zip codes we averaged the availability of attributes for all marketed homes providing reliable insight into the missingness problem. The home attribute data come from various sources: home owner/seller, county records, and broker listing feeds, and some of them are frequently incomplete. So, we have two options: to abandon the idea of home evaluation and selection using ORE data, or to use techniques for penalty-controlled missingness-tolerant aggregation. We use the method presented in [8] where the user can select the degree of penalty $P \in [0,1]$ (or $P \in [0,100\%]$) for missing data, as shown in Fig. 3. Then nonnegative inputs $x_i \geq 0$ correspond to known attributes, and negative inputs denote unknown attributes defined as $x_i = P - 1$. So, $x_i = 0$ denotes either no satisfaction of the corresponding elementary criterion or the maximum penalty assigned to an unknown attribute. In the case of negative suitability we have $0 \leq P < 1$, $-1 \leq x_i < 0$ and the zero penalty yields $x_i = -1$. Our missingness-tolerant aggregation structure maps $[-1,1]^n \rightarrow [0,1]$; for details, see [8].

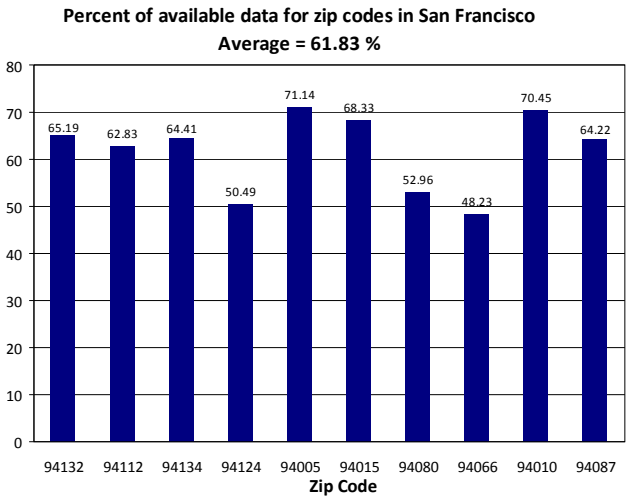


Fig. 6. ORE data availability for ten zip codes in San Francisco

The aggregation of suitability degrees is related to the perception of the importance of inputs. For example, if a homebuyer requires a high degree of simultaneity of the home quality and the home location quality, that requirement necessarily yields a

perception that both the home quality and the location quality are (for that specific homebuyer) very important. Thus, a high andness is a consequence of high overall importance of inputs. Similar situation also holds in the case of high orness. However, while the concepts of andness and orness are familiar to professional decision-makers, the concept of overall importance is familiar to everybody. This fact can be used to derive the andness/orness and other parameters of partial conjunction and partial disjunction from the verbalized perception of the overall importance of inputs. This idea was introduced in [7] and implemented in the LSPhome interface shown in Fig. 3 where users can select verbalized degrees of importance of home location, home quality, and home price. Using the method presented in [7] the selected degrees of importance of home location and home quality are used to derive the andness and weights for the final suitability aggregation block (W_L, W_H, F, Fig. 4). Then, the mean importance of location and home quality and the importance of price are used to derive the andness and the weights of the aggregator that aggregates the overall suitability and the overall affordability and provides the overall home value.

If the user wants a simultaneous satisfaction of k inputs and has the perception that their levels of overall importance (selected from the verbalized importance scale with $L+1$ levels) are $S_1,...S_k$, $S_i \in \{0,...,L\}$, $i=1,...,k, k>1$, then, according to [7], the corresponding andness is interpreted as the mean relative overall importance: $\alpha = (S_1 + ... + S_k)/kL$. Indeed, the perception of importance and the value of andness increase simultaneously and the above model is based on the linear relationship between the andness and the mean overall importance. The verbalized overall importance scale can also be used to derive the degrees of relative importance of inputs. Among three linear models proposed in [7] for computing weights $W_1,...W_k$, the simplest is the proportional scaling model $W_i = S_i/(S_1 + ... + S_k)$, $i=1,...,k$.

4 Experimental Results

Using the LSP methodology presented in previous sections we developed a web application called LSPhome that helps users to find the most suitable home according to their specific criteria. Traditional ORE web sites offer searches of available real estate inventory in the style of the traditional SQL SELECT-FROM-WHERE statement. The user is only allowed to specify a few crisp conditions in the WHERE clause. Such conditions are used as a filter, i.e. as a strictly binary selector that rejects all homes that do not satisfy any of the filtering conditions selected by the user. In a typical case the ORE web sites offer the filter conditions from the following list: (1) home type, (2) price range, (3) minimum number of bedrooms, (4) minimum number of bathrooms, (5) square feet range, (6) lot size range, (7) home age range, (8) time on market, (9) keywords used to select desired features (e.g. pool, patio), and (10) desired location/neighborhood. The filtering method considers all selected filtering conditions as the binary mandatory requirements. For example, if the home type is specified as condo/apartment, then all single family and multi family homes will be rejected. Obviously, the filtering process is useful, but it is not the home evaluation.

The home filtering is merely the partitioning of inventory in two basic groups: homes that are not acceptable and homes that might be acceptable. At this time the customers of ORE web sites do not have the possibility to determine the degree of acceptability (suitability) according to their specific needs. Evaluation and ranking of potentially suitable homes is left to the user and it is done intuitively. Of course, the number of attributes is too big for easy intuitive evaluation, and the process of home selection is usually stressful and time consuming.

The primary advantages of the soft computing approach and the LSP method with respect to the traditional filtering process are the evaluation and ranking of homes according to user needs, the reduction of search/decision time, and the justification of proposed decisions; that increases the confidence and improves the experience of homebuyer (and/or home seller). Of course, the central problem is how to define the user needs. The number of home attributes that we used (36) is a typical value and it is difficult to reduce it without losing the credibility of evaluation results. On the other hand, it is not reasonable to ask an average homebuyer to specify 36 elementary criteria (or fuzzy set membership functions) followed by an advanced aggregation structure. Thus, we proposed a hybrid approach: the user specifies 9 crucial requirements using the LSPhome interface shown in Fig. 3 and the remaining parts of the LSP criterion are prefabricated (fixed, reflecting average general requirements). In this way we combine the simplicity of specification of requirements and the breadth of covering relevant attributes. In particular, a significant advantage of our method is the integration of the home quality and the location quality attributes, with the possibility to conveniently adjust the relative importance of home quality versus the location quality. The location quality analysis is based on data about all points of interest provided by Google using techniques developed in [6] and [15].

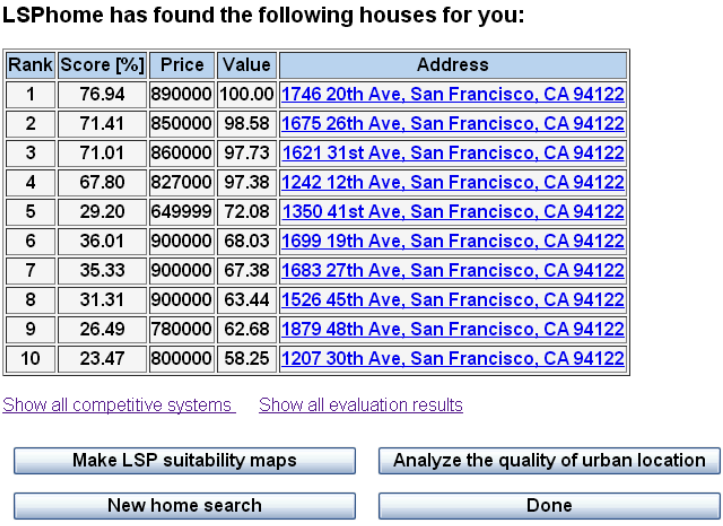


Fig. 7. An example of typical LSPhome evaluation results

A typical example of the summarized home evaluation and selection results generated by the LSPhome system is shown in Fig. 7 ("score" denotes the overall suitability). The user looking for a home in the vicinity of the 19th Avenue in San Francisco is given the ranking of 10 homes selected by LSPhome. The first four homes satisfy more than 2/3 of user requirements and other have too low suitability scores. The overall value is computed as a hard partial conjunction of the normalized suitability and normalized affordability: $V_i = W(\min(C_1,...,C_N)/C_i) \ \bar{\Delta}\ (1-W)(S_i/\max(S_1,...,S_N)) \in [0,1]$, $i = 1,...,N$. The weight W denotes the relative importance of affordability compared to the relative importance of home quality. It is computed from the importance of low price selected by the homebuyer using the LSPhome interface (Fig. 3). The results in Fig. 7 show the normalized relative value $V_i^{(norm)} = 100V_i / \max(V_1,...,V_N)),\ i = 1,...,N$, so that the top ranking home is rated 100%. In our example the four leading homes differ for less than 3%, while others have significantly lower values. Consequently, in this case the user is expected to focus on the four best options, compare homes using the values and the suitability of attributes (Fig. 8 and Fig. 9) and expand the investigation using suitability maps or a detailed analysis of the quality of urban location.

Walkability	47.0	54.0	49.0	49.0	23.0	44.0	32.0	82.0	53.0	40.0
Shopping and dining	76.0	79.0	79.0	84.0	55.0	83.0	66.0	87.0	80.0	81.0
Health support	38.0	42.0	64.0	66.0	41.0	59.0	35.0	74.0	64.0	63.0
Suitability for children	64.0	34.0	56.0	59.0	36.0	65.0	38.0	89.0	56.0	69.0
Suitability for seniors	40.0	39.0	61.0	59.0	28.0	58.0	51.0	79.0	63.0	54.0
Walking distance from the ideal location	1.76	1.75	0.02	0.65	1.29	0.72	1.92	1.01	0.26	0.88
Total internal living area of home	1350.0	982.0	1700.0	1385.0	1152.0	2266.0	1647.0	1046.0	2190.0	1686.0
Outer usable area belonging to home	1650.0	2014.0	1700.0	1611.0	1500.0	206.0	474.0	1046.0	2190.0	1912.0
Number of bedrooms	2.0	2.0	3.0	2.0	2.0	5.0	4.0	3.0	4.0	2.0
Number of bathrooms	2.0	1.0	2.0	1.0	1.0	4.0	2.0	1.5	3.0	1.1
Kitchen	50.0	40.0	60.0	*****	0.0	0.0	40.0	*****	*****	*****
Dining room/area	0.0	1.0	0.0	*****	1.0	0.0	1.0	*****	*****	*****
Living/family room	1.0	0.0	1.0	*****	0.0	1.0	0.0	*****	*****	*****
Breakfast room/area	1.0	1.0	1.0	*****	0.0	0.0	0.0	*****	*****	*****
Home office	1.0	0.0	0.0	*****	0.0	0.0	1.0	*****	*****	*****
Laundry	70.0	70.0	100.0	*****	0.0	0.0	35.0	*****	*****	*****
Walk-in closets	1.0	0.0	1.0	*****	0.0	0.0	0.0	*****	*****	*****
Pantry	0.0	0.0	0.0	*****	0.0	0.0	0.0	*****	*****	*****
Auxiliary utility areas	0.0	0.0	0.0	*****	0.0	0.0	0.0	*****	*****	*****
Private garage	0.0	3.0	3.0	*****	3.0	3.0	3.0	3.0	*****	*****
Shared garage	0.0	0.0	0.0	*****	0.0	0.0	0.0	0.0	*****	*****
Reserved uncovered parking space	0.0	0.0	0.0	*****	0.0	0.0	0.0	0.0	*****	*****
Street parking next to home	1.0	0.0	0.0	*****	0.0	0.0	0.0	0.0	*****	*****
Street parking close to home	0.0	0.0	0.0	*****	0.0	0.0	0.0	0.0	*****	*****
Paid public parking	0.0	0.0	0.0	*****	0.0	0.0	0.0	0.0	*****	*****
Type of home	4.0	4.0	1.0	4.0	4.0	4.0	4.0	1.0	1.0	4.0
Number of floors	1.0	1.0	2.0	*****	*****	1.0	2.0	*****	*****	*****
External wall material	1.0	1.0	*****	*****	*****	2.0	1.0	*****	*****	*****
Type of floor	5.0	3.0	2.0	*****	1.0	5.0	1.0	*****	*****	*****
Type of roof	1.0	1.0	*****	*****	0.0	0.0	0.0	1.0	*****	*****
Home energy supply	0.0	1.0	1.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0
Source of energy for heating	3.0	*****	*****	*****	*****	3.0	3.0	*****	*****	*****
Type of heating system	2.0	*****	*****	*****	*****	4.0	4.0	*****	*****	*****
Type of cooling system	0.0	0.0	*****	*****	0.0	0.0	0.0	0.0	*****	*****
Home age	97.0	109.0	12.0	77.0	90.0	84.0	65.0	108.0	12.0	81.0
Last modification/improvement	33.0	*****	*****	*****	*****	7.0	*****	5.0	*****	*****

Fig. 8. Attributes of ten competitive homes showing typical cases of missing data

1213	Parking space	33.60	75.50	75.50	*****	75.50	75.50	75.50	*****	75.50	*****
12132	Public parking (first-come, first-served)	75.00	0.00	0.00	*****	0.00	0.00	0.00	*****	0.00	*****
122	Home features	43.37	30.30	62.74	63.18	32.64	53.16	48.79	61.72	30.48	62.15
1225	Home age and maintenance	22.92	9.17	90.00	35.83	25.00	49.26	45.83	32.50	42.49	90.00
1221	Home organization/layout	94.28	94.28	58.87	100.00	100.00	94.28	100.00	100.00	50.00	50.00
12131	Reserved parking	0.00	94.48	94.48	*****	94.48	94.48	94.48	*****	94.48	*****
121311	Garage	0.00	100.00	100.00	*****	100.00	100.00	100.00	*****	100.00	*****
1212	Type of rooms/area	0.00	0.00	0.00	58.35	0.00	0.00	0.00	61.02	87.53	100.00
1211	Area belonging to home	91.86	71.75	100.00	93.76	81.07	83.91	88.77	100.00	74.01	100.00
121321	Free public parking	75.00	0.00	0.00	*****	0.00	0.00	0.00	*****	0.00	*****
1224	Home temperature regulation	50.37	0.00	*****	*****	0.00	64.32	64.32	*****	0.00	*****
1222	Home construction features	77.23	68.46	50.00	*****	44.22	81.13	46.46	*****	70.00	*****
12121	Primary rooms	0.00	0.00	0.00	58.35	0.00	0.00	0.00	61.02	87.53	100.00
12122	Additional space and storage	58.34	2.64	29.83	*****	0.00	0.00	4.27	*****	*****	*****
121222	Storage and auxiliary areas	29.83	0.00	29.83	*****	0.00	0.00	0.00	*****	*****	*****
121221	Additional space	93.64	26.09	29.83	*****	0.00	0.00	42.16	*****	*****	*****
11	QUALITY OF LOCATION	65.60	60.69	72.02	71.94	46.80	70.65	53.58	69.67	86.08	72.84
111	Suitability of neighborhood	56.89	50.65	60.06	61.61	34.48	60.17	42.84	59.36	82.96	61.68
12252	Last modification/improvement	29.75	*****	*****	*****	*****	79.00	*****	*****	85.00	*****
12251	Home age	19.17	9.17	90.00	35.83	25.00	30.00	45.83	32.50	10.00	90.00
12243	Type of cooling system	0.00	0.00	*****	*****	0.00	0.00	0.00	*****	0.00	*****
12242	Type of heating system	60.00	*****	*****	*****	*****	100.00	100.00	*****	*****	*****
12241	Source of energy for heating	90.00	*****	*****	*****	*****	90.00	90.00	*****	*****	*****

Fig. 9. A fragment of evaluation results showing the missing attribute and subsystem data

A typical problem of missing data is visible in Fig. 8 (obtained using the "show all competitive systems" option in Fig. 7) and in Fig 9 (obtained using the "show all evaluation results" option in Fig. 7). Out of ten competitive homes only two homes have the complete attribute data. All other data are incomplete. Furthermore, the missing attribute data propagate through the aggregation tree and some subsystems (e.g. garage and reserved parking) have missing values shown in Fig. 9.

In order to deal with missing data evaluators must decide about the most suitable value of the missingness penalty parameter. The effects of missingness penalty are shown in Fig. 10. In all cases increasing the missingness penalty causes a decrease of the overall suitability. For the maximum penalty the overall suitability for missing nonmandatory attributes is positive, and for missing mandatory attributes it is zero.

Fig. 10. Overall suitability as a function of missingness penalty

The selection of missingness penalty is based on the decision maker missingness tolerance level. Indeed, the missing data can be intentionally hidden because they are inconvenient, or they can be unknown to all data providers. In the case of suspected inconvenient data it is justifiable to apply the highest penalty. In the case where we have reasons to believe that the unknown attributes are satisfied (e.g. the house with missing parking data is in a residential district that is known to have free public parking space) we may select a lower penalty value. To decide about the most appropriate missingness penalty it is suitable to first plot and analyze the overall suitability curves similar to those shown in Fig. 10. The suitability functions in Fig. 10 are strictly concave and the penalty of 80% should be applied if we want to get the overall suitability that is approximately halfway between the extreme values.

The evaluation results (Fig. 7) offer the possibility for detailed investigation of the suitability of home location and its neighborhood. The evaluation tools [6], [15] (also available at www.seas.com) provide suitability maps, which are geographic maps with an overlay showing the distribution of suitability degrees. Fig. 11 shows the suitability map for walkability (possibility to access selected points of interest by walking) where the suitability degrees are presented as numeric values on top of a Google map with selected points of interest. We define walkability as a conjunctive partial absorption of a set of mandatory points of interest and a set of optional points of interest. For the best home proposed in Fig. 7, the walkability is 53%. The potential homebuyer can also investigate the suitability of selected neighborhood for business, children, entertainment, shopping, etc., or make his/her own suitability map.

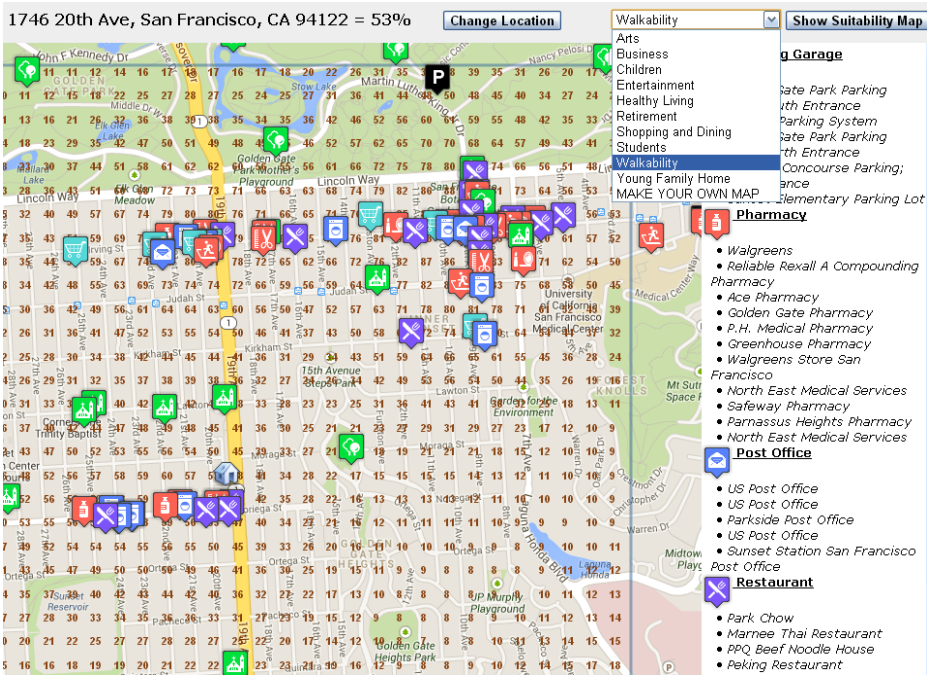


Fig. 11. Suitability map on top of a Google map in the vicinity of the selected home

Suitability maps based on points of interest provided by Google give useful information about the suitability of neighborhood, but they do not include physical, environmental, and safety aspects of the neighborhood. Such information can be collected from other sources (e.g. government) and used for additional analysis of the suitability of location. Fig. 12 shows a sample of such an analysis (based on the analyzer of the quality of urban locations developed in [15] and activated as an option in Fig. 7). The quality of urban locations is analyzed using suitability maps based on 11 diverse attributes presented in Fig. 12. In the given point of the best home reported by LSPhome the quality of location is 64%. This value can be compared with the presented distribution of the location quality in the whole city (the best values around 70% and the mean value of 55.41%). Thus, the neighborhood of the selected home is notably above the city average and not too far from the best locations in the city.

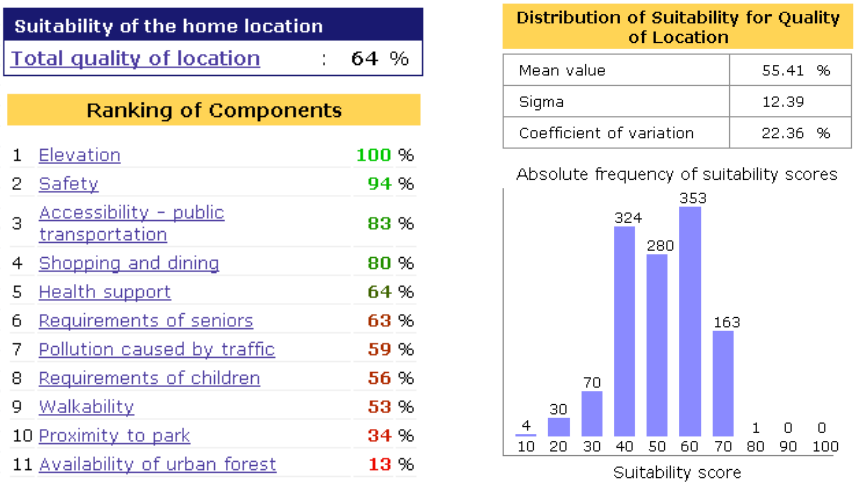


Fig. 12. Quality of urban locations based on 11 diverse attributes

5 Conclusions

Evaluation and selection of homes is essentially a soft computing logic problem. ORE web sites offer data that enable the use of customized compensative logic criteria for fast ranking of available homes. This paper shows a way such criteria can be designed using the LSP method, and implemented in a software tool available over the Internet. Specific problems related to online buying and selling of homes include the missingness-tolerant aggregation and the use of the verbalized concept of overall importance to derive the andness/orness and weights of partial conjunction and partial disjunction aggregators. Soft computing decision methods are a way to significantly improve both the efficiency and the customer experience in online real estate.

References

1. Beliakov, G., Pradera, A., Calvo, T. (eds.): Aggregation Functions: A Guide for Practitioners. STUDFUZZ, vol. 221. Springer, Heidelberg (2007)
2. Bullen, P.S.: Handbook of means and their inequalities. Kluwer (2003)
3. Dujmović, J.J.: Preference logic for system evaluation. IEEE Transactions on Fuzzy Systems 15(6), 1082–1099 (2007)
4. Dujmović, J.J., Larsen, H.L.: Generalized conjunction/disjunction. International J. Approx. Reas. 46, 423–446 (2007)
5. Dujmović, J.J., Nagashima, H.: LSP method and its use for evaluation of Java IDE's. International J. Approx. Reas. 41(1), 3–22 (2006)
6. Dujmović, J.J., De Tré, G.: Multicriteria Methods and Logic Aggregation in Suitability Maps. International Journal of Intelligent Systems 26(10), 971–1001 (2011)
7. Dujmović, J.J.: Andness and Orness as a Mean of Overall Importance. In: Proceedings of the IEEE World Congress on Computational Intelligence,-Brisbane, Brisbane, Australia, June 10-15, pp. 83–88 (2012)
8. Dujmović, J.: The Problem of Missing Data in LSP Aggregation. In: Greco, S., Bouchon-Meunier, B., Coletti, G., Fedrizzi, M., Matarazzo, B., Yager, R.R., et al. (eds.) IPMU 2012, Part III. CCIS, vol. 299, pp. 336–346. Springer, Heidelberg (2012)
9. Gini, C., Barbensi, G., Galvani, L., Gatti, S., Pizzetti, E.: Le Medie. Unione Tipografico-Editrice Torinese, Torino (1958)
10. Grabisch, M., Marichal, J.-L., Mesiar, R., Pap, E.: Aggregation Functions. Cambridge Univ. Press (2009)
11. National Association of Realtors, NAR Fact Sheet (2013), `http://www.realtor.org/for-the-media/nar-fact-sheet`
12. Torra, V., Narukawa, Y.: Modeling Decisions. Springer (2007)
13. Trulia (2013), `http://www.trulia.com/`
14. Wikipedia, Multiple listing service (2013), `http://en.wikipedia.org/wiki/Multiple_Listing_Service`
15. Yokoohji, R.: LSP Suitability Maps for the Quality of Urban Locations. MS thesis. San Francisco State University, Department of Computer Science, Culminating Experience Report SFSU-CS-CE-12.09 (2012)
16. Zillow (2013), `http://www.zillow.com/`

Constraints Preserving Genetic Algorithm for Learning Fuzzy Measures with an Application to Ontology Matching

Mohammad Al Boni[1], Derek T. Anderson[2], and Roger L. King[1]

[1] Center for Advanced Vehicular Systems
Mississippi State University, MS 39762 USA
mma201@msstate.edu, rking@cavs.msstate.edu
[2] Electrical and Computer Engineering Department
Mississippi State University, MS 39762 USA
anderson@ece.msstate.edu

Abstract. Both the fuzzy measure and integral have been widely studied for multi-source information fusion. A number of researchers have proposed optimization techniques to learn a fuzzy measure from training data. In part, this task is difficult as the fuzzy measure can have a large number of free parameters ($2^N - 2$ for N sources) and it has many (monotonicity) constraints. In this paper, a new genetic algorithm approach to constraint preserving optimization of the fuzzy measure is present for the task of learning and fusing different ontology matching results. Preliminary results are presented to show the stability of the leaning algorithm and its effectiveness compared to existing approaches.

Keywords: Fuzzy measure, fuzzy integral, genetic algorithm, ontology matching.

1 Introduction

The fuzzy integral (FI), introduced by Sugeno [14], is a powerful tool for data fusion and aggregation. The FI is defined with respect to a fuzzy measure (FM). The FM encodes the worth of different subsets of sources. Successful uses of the FI include, to name a few, multi-criteria decision making [6], image processing [15], or even in robotics [12].

It is well-known that different FMs lead the FI, specially the Choquet FI, to behave like various operators (max, min, mean, etc.) [15]. In some applications, we can define the FM manually. However, in other settings, it may only be possible to define the densities (the measures on just the singletons) and a measure building technique is used e.g., a S-Decomposable measure such as the Sugeno λ-fuzzy measure. It is also very difficult, if possible at all, to specify a FM for relatively small N (number of inputs), as the lattice already has $2^N - 2$ free parameters! (e.g., for $N = 10$ inputs, $2^{10} - 2 = 1022$ free parameters). Also, input sources can be of different nature. For example, we may fuse values from sensors

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_9,

with algorithms outcomes. In such problems, it is hard to determine the importance of each input data source. Thus, many data-driven learning methods are used to learn the measure. One method is to use a quadratic program (QP) to learn the measure based on a given data set [6]. Although the QP is an effective approach to build the measure using a data set, its complexity is relatively high and it does not scale well [10]. Other optimization techniques have been used to reduce the complexity and improve performance, e.g., genetic algorithms (GA) [2]. The goal of this article is to design and implement a GA to learn the full FM from ontology similarities. The main contribution of this article includes: the investigation of new constraint preserving operators (crossover and mutation) in a GA that overcome drawbacks of prior work. The application of this theory is to learn a FM to fuse multiple similarity techniques in the context of ontology matching.

2 Related Work in Ontologies

A number of works have been put forth in the literature regarding combining several ontology matching techniques. In [18], Wang et al. used Dempster Shafer theory (DST) to combine results obtained from different ontology matcher. Each result is treated as a mass function and combined using the Dempster's Combination Rule. It is not clear that DST is the correct fit because in DST one typically only has access to a limited amount of evidence in different focal sets. However, the authors appear to have access to all information in the form of ontology term matching matrices. Other papers, such as [9,17], have used a GA to learn the weight of each matching algorithm for an operator like a weighted sum. Herein, an improved set of constraint preserving GA operators are put forth and more powerful non-linear aggregation operator, the FI, is investigated.

3 Fuzzy Integral and Fuzzy Measure

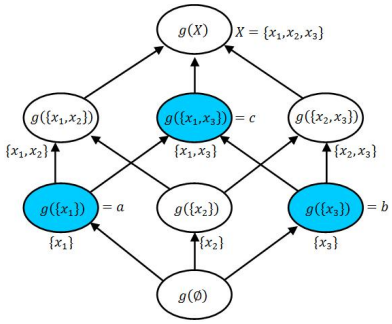
For a set of N input sources, $X = \{x_1, x_2, ..., x_N\}$, the discrete FI is:

$$\int_s h \circ g = \bigvee_{i=1}^{N} \left(h(x_{\pi_{(i)}}) \wedge G(x_{\pi_{(i)}}) \right), \tag{1}$$

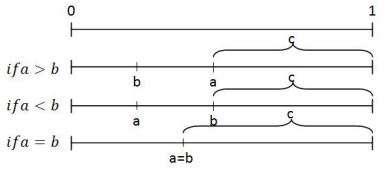
where h is the partial support function, $h : X \to [0, 1]$, $h(x_{\pi_{(i)}})$ is the evidence provided by source $\pi_{(i)}$, π is a re-permutation function such that $h(x_{\pi_{(1)}}) \geq h(x_{\pi_{(2)}}) \geq ... \geq h(x_{\pi_{(N)}})$. The value $G(x_{\pi_{(i)}}) = g(\{x_{\pi_{(1)}}, x_{\pi_{(2)}}, ..., x_{\pi_{(i)}}\})$ is the measure of a set of information sources. Note, $\int_s h \circ g$ is bounded between $\left[\bigwedge_{i=1}^{N} h(x_i), \bigvee_{i=1}^{N} h(x_i) \right]$ and it can be explained in words as "the best pessimistic agreement".

The FM can be discussed in terms of its underlying lattice. A lattice is induced by the monotonicity constraints over the set 2^N. Each node or vertex in the

lattice is the measure for a particular subset of sources, e.g., $g(\{x_1, x_3\})$. Figure 1a is an illustration of the FM for $N = 3$. Formally, a FM, g, is a set-valued function $g : 2^X \rightarrow [0, 1]$, where $g(\emptyset) = 0$, $g(X) = 1$, and for $A, B \subseteq X$, such that $A \subseteq B$, $g(A) \leq g(B)$ (the monotonicity constraint).



(a) Lattice view of a FM.



(b) Interval view of a monotonicity constraint.

Fig. 1. (a) FM lattice for $N = 3$ and (b) possible values for $g(\{x_1, x_3\})$ for a given $g(\{x_1\})$ and $g(\{x_3\})$

4 Preserving FM Constraints in a Genetic Algorithm

Many previous attempts exist to address constraint satisfaction in a GA. Typically, a penalty function is used to reduce the fitness of solutions that violate constraints [11]. However, even if infeasible solutions are close to an ideal minimum, it is not clear why or how to really balance the cost function to eventually help it converge to a quality valid solution. In the FM, we have an exponentially increasing number of constraints in N. It is a highly constrained problem. It is not likely that we will end up obtaining a valid FM for such a heavy constrained problem (or that one such infeasible solution could simply be made a quality feasible solution at the end). Therefore, penalty function strategy becomes very risky to use and will restrain the search space. Many researchers avoid dealing with the FM constraints by designing a GA that learn only the densities. Then, the rest of the lattice is populated using a measure deriving technique such as the Sugeno λ-fuzzy measure. However, a better solution is to design a more intelligent set of GA operators for the constraints in a FM so we can efficiently search just the valid FM space and operate on valid FMs. Since all chromosomes in a population are always valid (started and remained valid by using our crossover and mutation operators), we did not have to modify the selection process. Only crossover and mutation need be defined to preserve the monotonicity constraints.

4.1 Constraint Preserving Crossover for the FM

In this subsection, a new GA crossover operation is described that preserves the monotonicity property of the FM. First, g($\emptyset$) and g(X) need not be considered as they are 0 and 1 by definition. It is the remaining $2^N - 2$ lattice elements that we are concerned with. In order to ensure a valid FM after crossover, we decompose the FM into a set of interval relations that present the allowable bounds for crossover. For example, consider a set $\{x_1, x_3\}$ of sources, $\{x_1\}$ and $\{x_3\}$, such that $g(\{x_1\}) = a$, $g(\{x_3\}) = b$ and $g(\{x_1, x_3\}) = c$. We know from the FM that $max(a, b) \leq c$. Figure 1b illustrates the valid interval ranges that a, b and c can posses in order to remain a valid (credible) FM.

Definition 1 (Intersected and non-inclusive intervals). Let $d = [d_1, d_2]$, $e = [e_1, e_2]$ be two intervals such that $d \cap e \neq \emptyset$. Let d be the smaller of the two intervals, i.e., $d_2 - d_1 < e_2 - e_1$. If $d_1 \leq e_1$ or $d_2 \geq e_2$ then we call d and e intersected and non-inclusive intervals.

Note, this property is needed herein in order to identify candidate chromosomes to perform crossover on.

Definition 2 (Random density crossover). Let g_1 and g_2 be two FMs. Let $\leftrightarrow$ denote the random selection and swapping of one density from g_1, $b_1 = g_1(x_i)$, and g_2, $b_2 = g_2(x_j)$, where i,j are random numbers in $\{1, 2, .., N\}$.

Note, by itself, $\leftrightarrow$ does not guarantee a valid FM.

Definition 3 (Repair operator for $\leftrightarrow$). Let $\Leftrightarrow$ (explained in Prop 1) denote an operation to repair a violation of the monotonicity constraint caused by $\leftrightarrow$.

Even though we discuss crossover of densities ($\leftrightarrow$), $\Leftrightarrow$ has the result that it impacts multiple layers in the FM. As such, while we only cross two densities, we are in fact changing many values in a FM. It is important to note that our $\Leftrightarrow$ makes use of existing values in the FM to repair any violations in the other FM. This means no "new" information is injected, rather the values in two FMs are swapped (in the classical theme of crossover).

Proposition 1: Let g_1 and g_2 be two FMs with N input sources and let the measure on the densities $\{\{x_1\}, \{x_2\}, ..., \{x_N\}\}$ be $\{g_1^1, g_1^2, ..., g_1^N\}$ and $\{g_2^1, g_2^2, ..., g_2^N\}$ respectively. Furthermore, let each measure be sorted individually such that $g_1^{\pi_1(1)} \leq g_1^{\pi_1(2)} \leq ... \leq g_1^{\pi_1(N)}$ and $g_2^{\pi_2(1)} \leq g_2^{\pi_2(2)} \leq ... \leq g_2^{\pi_2(N)}$ where π_1,π_2 are re-permutation functions. Furthermore, we make the assumption that each corresponding sorted sub-interval is intersected and non-inclusive between g_1 and g_2 (Def.1). Then, all admissible $\leftrightarrow$ (Def.2), followed by $\Leftrightarrow$ (Def.3), operations are guaranteed to not violate the FM monotonicity property.

Proof: This proposition is proved by considering all possible enumerable cases. First, we divide the problem into identical sub-problems, and we prove the

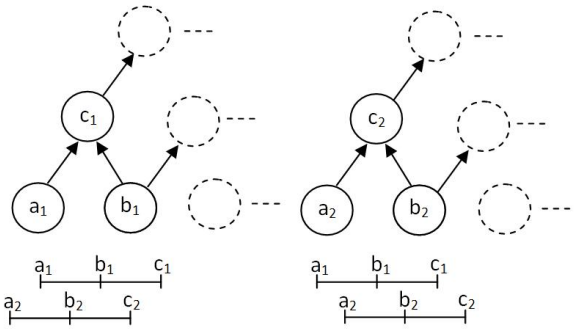


Fig. 2. Two FMs with possible range conditions for swapping values across measures

proposition for the case of one interval pair. Next, this single case is extended to the case of multiple corresponding interval pairs.

First, g_1 and g_2 are individually sorted. Then, the intervals $[g_1^{\pi_1(1)}, g_1^{\pi_1(N)}]$ and $[g_2^{\pi_2(1)}, g_2^{\pi_2(N)}]$ are divide into sub-intervals, each with two sources i.e., $\{[g_1^{\pi_1(1)}, g_1^{\pi_1(2)}], [g_1^{\pi_1(2)}, g_1^{\pi_1(3)}]$, ..., $[g_1^{\pi_1(N-1)}, g_1^{\pi_1(N)}]\}$ and $\{[g_2^{\pi_1(1)}, g_2^{\pi_1(2)}], [g_2^{\pi_1(2)}, g_2^{\pi_1(3)}]$, ..., $[g_2^{\pi_1(N-1)}, g_2^{\pi_1(N)}]\}$. Next, we look at the case of a single corresponding interval pair between g_1 and g_2. Let $a_1 = g_1^{\pi_1(1)}$, $b_1 = g_1^{\pi_1(2)}$, $a_2 = g_2^{\pi_2(1)}$ and $b_2 = g_2^{\pi_2(2)}$. Since $[a_1, b_1], [a_2, b_2]$ are intersected and non-inclusive intervals (Def.1), then we have two cases (see Figure 2):

$$a_2 \leq a_1 \leq b_2 \leq b_1 \tag{2}$$

$$a_1 \leq a_2 \leq b_1 \leq b_2 \tag{3}$$

For case 2, there are four admissible crossover operations:
(1) $a_1 \leftrightarrow a_2$: no need to swap c_1 and c_2 and the new intervals are $[a_2, b_1], [b_1, c_1]$ and $[a_1, b_2], [b_2, c_2]$.
(2) $a_1 \leftrightarrow b_2$: Since $b_2 \leq b_1$, the new intervals are $[b_2, b_1], [b_1, c_1]$ and $[a_2, a_1]$, $[a_1, c_2]$. No need to swap c_1 and c_2.
(3) $b_1 \leftrightarrow a_2$: Swap $c_1 \Leftrightarrow c_2$. The new intervals are $[a_2, a_1], [a_1, c_2]$ and $[b_2, b_1]$, $[b_1, c_1]$.
(4) $b_1 \leftrightarrow b_2$: Swap $c_1 \Leftrightarrow c_2$. The new intervals are $[a_1, b_2], [b_2, c_2]$ and $[a_2, b_1]$, $[b_1, c_1]$.
Case 3 is proved the same way i.e., perform $\leftrightarrow$ then check the resulted intervals and apply $\Leftrightarrow$ when required. Because $\leftrightarrow$ is applied on one density in each FM, the proof of the first sub-interval case is easily generalized to the other cases.
Since Def.1 holds, we randomly select one density in g_1 to be swapped with a different randomly swapped density in g_2. Then, $\Leftrightarrow$ is iteratively repeated up the lattice to fix all violations resulting from the $\leftrightarrow$ operator. At each layer in the lattice, which means for all measure values on sets of equal cardinality, intervals are compared and $\Leftrightarrow$ is applied when required.

4.2 Constraints Preserving Mutation

In mutation, a random value in the lattice will be changed. In order not to violate the monotonicity constraint, the new value of a randomly selected node, should be greater than or equal the maximum measure of the nodes coming into it and less than or equal the minimum measure of those it goes into at the next layer. This is the only required modification to the traditional mutation by restricting the new value between a minimum and maximum thresholds.

5 Application: Ontology Matching with the Fuzzy Integral

Ontologies are used in different knowledge engineering fields, for example the semantic web. Because it is widely used, many redundant ontologies have been created and many ontologies may fully, or partially, overlap. Matching these redundant and heterogeneous ontologies is an on-going research topic. A good survey is presented in [1]. However, the no-free-lunch theorem was proven, i.e., no algorithm can give the optimal ontology matching among all knowledge domains, due to the complexity of the problem [1]. In this section, we implement an algorithm to combine results from different similarity matching techniques using the FI. So is the claim, if we cannot outright solve it, then aggregate multiple methods to help improve the robustness of our approach.

Ontology matching is a process, applied on two ontologies, that tries to find, for each term within one ontology (the source), the best matched term in the second ontology (the destination). This problem can be approached using several mechanisms based on: string normalization, string similarity, data-type comparison, linguistic methods, inheritance analysis, data analysis, graph-mapping, statistical analysis, and taxonomy analysis [9]. In this paper, we fused results from existing matchers: FOAM ([4]), FALCON ([7]) and SMOA ([13]).

5.1 Genetic Algorithm Implementation

Encoding: Each chromosome is a vector of [0,1]-valued numbers that map lexographically to a FM e.g., $(g_1, g_2, ..., g_N, g_{1,2}, ..., g_{1,...,N-1})$. The length of a chromosome is $2^N - 2$ (where there are N similarity matching algorithms).
Fitness Function: Semantic precision and recall are widely used to evaluate the performance of ontology matching algorithms [5].They are an information retrieval metrics [16] that have been used for ontology matching evaluation since 2002 [3]. Fmeasure is a compound metric that can reflect both precision and recall, and it is given by the formula:

$$Fmeasure = \frac{2 * precision * recall}{precision + recall}, \tag{4}$$

where precision is given by:

$$precision = \frac{|A \cap R|}{A}, \tag{5}$$

and recall is given by:

$$recall = \frac{|A \cap R|}{R}, \tag{6}$$

R is the reference ontology and A is the alignment resulted from the matching technique. First, the most fitted chromosome (FM) is used with the FI to calculate A. Then, A is used in computing precision and recall. Lastly, Fmeasure is obtained from the resulted precision and recall. The goal of our GA is to maximize the Fmeasure. We linearly scaled the plot (Fig 3) to aid visual display.

Crossover: Crossover is performed in two phases. The first phase checks the intersection property between the parents' chromosomes so they will result in a valid offspring. The second phase performs Def.2 and 3.

Selection: We adopted traditional roulette wheel selection.

Stop Condition: A maximum number of iterations is used as the stopping condition.

5.2 Experimental Results and Evaluation

Several tests were conducted on the I^{3}CON [8] data set. We compared the performance of our tool with existing tools using the precision, recall and Fmeausre (see figures 5a, 5b, 5c). Although our approach gives lower precision than FOAM, FALCON and SMOA individually, it gives a better Fmeasure. This occurs because we combine several results which reduces precision but increases recall considerably. Our tool provides a precision of 1 if and only if the precision of all combined sources is 1. Also, our tool tends to give better results than GOALS. GOALS uses an OWA operator to combine results from different matchers. We conducted two experiments on the "AnimalsA.owl" and "AnimalsB.owl" using different OWAs. With an $OWA = [0.3, 0.4, 0.3]$ ($OWA = [0.7, 0.2, 0.1]$ respectively) we get an Fmeasure of 0.8571 (0.875 respectively) while our tool returns a value of 0.8936. Inconsiderate (underestimated or overestimated) selection of OWA values dose affect the performance of the system. Since different matchers give different results on different domains, it is difficult to choose the optimal selection of OWA that gives the best results. Our system solves this problem by using the GA to learn the weights. We can learn an OWA if its needed, or any other aggregation operators, but it does not need to be decided up front.

Also, figures 3a, 3b, 3c, show the convergence of the GA using 10,20, and 30 iterations with crossover rate of 0.1, 0.2 and 0.3. Each graph is the average fitness value of the population at each iteration. We also report, at each x-location, the full range bounds (max and min). By comparing the results from different configurations, we found that a 10% crossover rate with 10 iterations was enough to achieve the desired results herein. With such configurations, the GA reached convergence with minimal computational effort. Also, we found that the range bounds decreases, which reflects convergence over time. Moreover, we found that all of the plots decrease at some point. This is caused by mutation, in

which we are exploring a bigger space which may not have a high fitness value. To determine the performance of the constraint preserving crossover, we conducted two studies. Study 1 shows the average number of chromosomes that have a

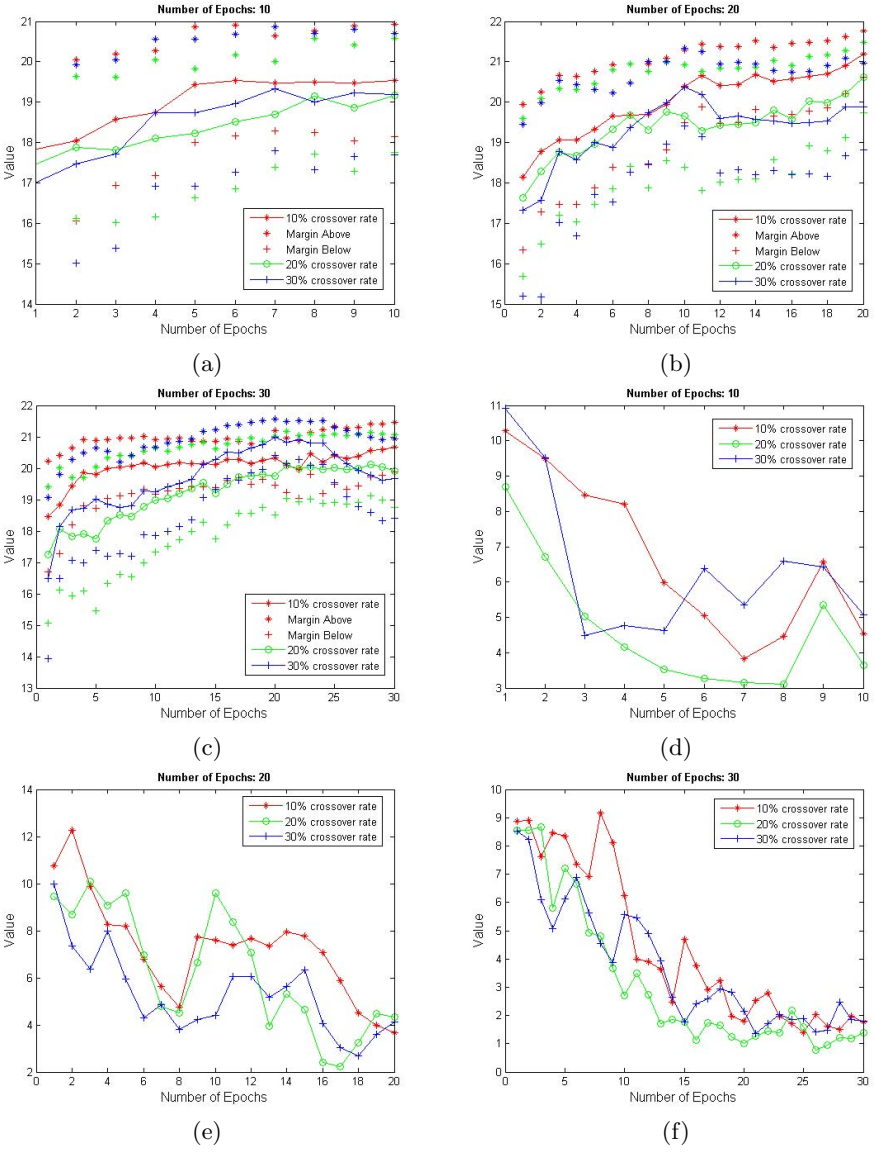


Fig. 3. (a)(b)(c) Analysis of GA behavior and performance for different number of epochs (10,20,30) and different crossover rates (10%, 20%, 30%.) (d)(e)(f) Average number of chromosomes with a valid interval property.

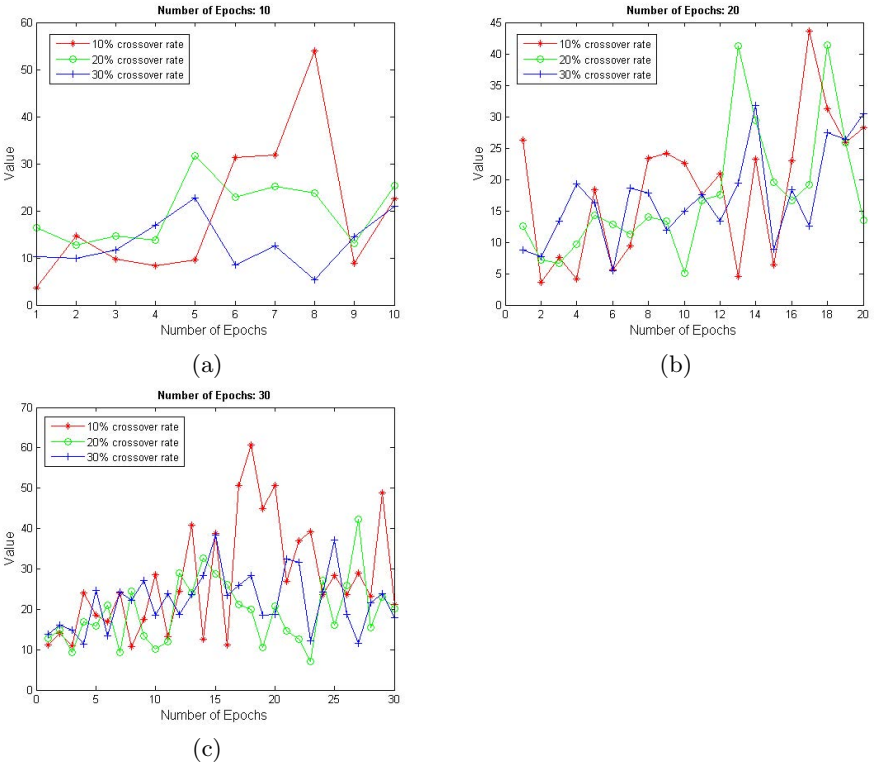


Fig. 4. (a)(b)(c) Average number of attempts to find a chromosome with a valid interval property

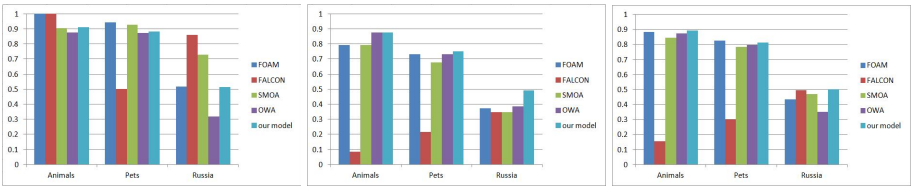


Fig. 5. (a)(b)(c) Ontology matching evaluation metrics: precision, recall, Fmeasure

valid interval property (see figures 3d, 3e, 3f). This reflects the applicability of our crossover model. We notice that the average number of suitable mates tend to decrease at each iteration. That happens because chromosomes will converge more at each iteration and will likely violate the interval property. Study 2 is conducted on the average number of attempts to find a suitable mate to crossover with (see figures 4a, 4b, 4c). We found that this number tends to increase in

average at each iteration because fewer chromosomes will have a valid interval property. Thus, we have less of a chance to find a valid candidate pair and the number of attempts increases.

6 Conclusion and Future Work

In this paper we propose a new constraint preserving GA for learning FMs. Specifically, we proposed a new crossover and mutation operation and different ontology matching algorithms were fused using the FI learned by the GA. We showed that our framework can give satisfactory results on different study domains. However, a few challenges need to be extended for future work, e.g., we would like to explore the use of the shapely index to determine the importance of each matching algorithm. Also, we will study the use of more than two ontologies to support the matching between ontologies. Also, the decreasing trend of the average number of matches raises an important question. What if we had to run the algorithm for a long time, would we reach a starvation (a case in which we can't find a valid candidate to crossover with)? If yes, do we need to relax our constraints? If we could not, could we design a new operator that can insert new valid candidates to keep the algorithm moving? Passing multiple populations? Point being, need to revisit the harsh constraints on learning. Likewise, we would like to explore the use of island GAs on different populations which may help increase the chance of success and diversity.

Acknowledgments. The first author would like to acknowledge and thank the Fulbright scholarship program for its financial support of the work performed herein.

References

1. Amrouch, S., Mostefai, S.: Survey on the literature of ontology mapping, alignment and merging. In: 2012 International Conference on Information Technology and e-Services (ICITeS), pp. 1–5. IEEE (2012)
2. Anderson, D.T., Keller, J.M., Havens, T.C.: Learning fuzzy-valued fuzzy measures for the fuzzy-valued sugeno fuzzy integral. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. LNCS, vol. 6178, pp. 502–511. Springer, Heidelberg (2010)
3. Do, H.-H., Melnik, S., Rahm, E.: Comparison of schema matching evaluations. In: Chaudhri, A.B., Jeckle, M., Rahm, E., Unland, R. (eds.) NODe-WS 2002. LNCS, vol. 2593, pp. 221–237. Springer, Heidelberg (2003)
4. Ehrig, M., Sure, Y.: Foam-framework for ontology alignment and mapping-results of the ontology alignment evaluation initiative. In: Workshop on Integrating Ontologies, vol. 156, pp. 72–76 (2005)
5. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. In: Proc. 20th International Joint Conference on Artificial Intelligence (IJCAI), pp. 348–353 (2007)
6. Grabisch, M.: The application of fuzzy integrals in multicriteria decision making. European Journal of Operational Research 89(3), 445–456 (1996)

7. Hu, W., Qu, Y.: Falcon-ao: A practical ontology matching system. Web Semantics: Science, Services and Agents on the World Wide Web 6(3), 237–239 (2008)
8. I. Interpretation and I. Conference. Ontology Alignment Dataset (2004), http://www.atl.lmco.com/projects/ontology/i3con.html (accessed June 4, 2013)
9. Martinez-Gil, J., Alba, E., Aldana-Montes, J.F.: Optimizing ontology alignments by using genetic algorithms. In: Proceedings of the Workshop on Nature Based Reasoning for the Semantic Web, Karlsruhe, Germany (2008)
10. Mendez-Vazquez, A., Gader, P.: Sparsity promotion models for the choquet integral. In: IEEE Symposium on Foundations of Computational Intelligence (FOCI 2007), pp. 454–459. IEEE (2007)
11. Michalewicz, Z.: Genetic algorithms, numerical optimization, and constraints. In: Proceedings of the Sixth International Conference on Genetic Algorithms, vol. 195, pp. 151–158. Morgan Kaufmann, San Mateo (1995)
12. Pan, G.-Y., Wang, W.-Y., Tsai, C.-P., Wang, W.-Y., Tsai, C.-R.: Fuzzy measure based mobile robot controller for autonomous movement control. In: 2011 International Conference on System Science and Engineering (ICSSE), pp. 649–653 (2011)
13. Stoilos, G., Stamou, G., Kollias, S.D.: A string metric for ontology alignment. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 624–637. Springer, Heidelberg (2005)
14. Sugeno, M.: Theory of fuzzy integrals and its applications. PhD thesis, Tokyo Institute of Technology (1974)
15. Tahani, H., Keller, J.M.: Information fusion in computer vision using the fuzzy integral. IEEE Trans on Systems, Man and Cybernetics 20(3), 733–741 (1990)
16. van Rijsbergen, C.J.K.: Information retrieval. Butterworths, London (1975), http://www.dcs.gla.ac.uk/Keith/Preface.html
17. Wang, J., Ding, Z., Jiang, C.: Gaom: genetic algorithm based ontology matching. In: Services Computing, pp. 617–620. IEEE (2006)
18. Wang, Y., Liu, W., Bell, D.: Combining uncertain outputs from multiple ontology matchers. In: Prade, H., Subrahmanian, V.S. (eds.) SUM 2007. LNCS (LNAI), vol. 4772, pp. 201–214. Springer, Heidelberg (2007)

Topology Preservation in Fuzzy Self-Organizing Maps

Mohammed Khalilia[1] and Mihail Popescu[2]

[1] Computer Science
University of Missouri
Columbia, Missouri, USA
mohammed.khalilia@gmail.com
[2] Health Management and Informatics
University of Missouri
Columbia, Missouri, USA
PopescuM@missouri.edu

Abstract. One of the important properties of SOM is its topology preservation of the input data. The topographic error is one of the techniques proposed to measure how well the continuity of the map is preserved. However, this topographic error is only applicable to the crisp SOM algorithms and cannot be adapted to the fuzzy SOM (FSOM) since FSOM does not assign a unique winning neuron to the input patterns. In this paper, we propose a new technique to measure the topology preservation of the FSOM algorithms. The new measure relies on the distribution of the membership values on the map. A low topographic error is achieved when neighboring neurons share similar or same membership values to a given input pattern.

Keywords: Fuzzy self-organizing map, topology preservation, map continuity, relational data.

1 Introduction

Self-Organizing Maps (SOM) is an unsupervised neural network algorithm. SOM tries to map the *s*-dimensional input patterns to a 2-dimensional lattice, preserve the topology of the data, and cluster the neurons that represent similar input patterns, which can be visualized using a 2D or 3D map such as the Unified Distance Matrix (U-Matrix) [1]. Several formulations and modifications were proposed to the classical SOM algorithm, such as the Self-Organizing Semantic Maps [2], Ontological SOM [3], Relational Topographic Maps [4], and WEBSOM [5]. Another class of SOMs is the fuzzy SOM algorithms. The general idea of FSOM is to integrate fuzzy set theory into neural networks to give SOM the capabilities of handling uncertainly in the data. FSOM can also be divided into two categories: object FSOM [6–10] where input patterns are represented as feature vectors and the relational FSOM [11] which handles relational data.

Regardless of the type of SOM algorithm they all share one important feature that is topology preservation. Topology preservation means that neighboring data points in

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_10,

the input space are mapped to nearby neurons in the output space. Once a good mapping is established, SOM can represent the high dimensional input space in a 2-dimensional output map that preserves the topology of the input data. This in turn yields better visualization and reveals more information about the structure and the clusters presented in high dimensional input space. To ensure that SOM has established good mapping, we need to measure or quantify the goodness of SOM. Different measures are proposed to accomplish this goal, such as the quantization error and the topographic error. Those errors are widely used in SOM and while the quantization error was adapted for the object and relational FSOM [11], no formulation is yet proposed to measure the topological preservation or continuity of the map in the FSOM algorithms.

The topographic errors used in SOM are not directly applicable to FSOM due to the fact that FSOM does not assign a unique winning neuron for every object, instead every neuron is a winning a neuron of every object with a varying degree of membership. Therefore, in this work, we propose a technique to measure the topographic error in FSOM algorithms.

The reminder of the paper is organized as follows: Section 2 gives an overview of the fuzzy relational SOM. Section 3 discusses some of the well-known methods to measure the goodness of SOM. Section 4 explains a new approach to measure the topographic error in FSOM. Section 5 presents experimental results and we conclude this paper with remarks and discussion in Section 6.

2 Fuzzy Relational Self-Organizing Maps

In this section we give a very brief overview of the fuzzy relational SOM algorithm (FRSOM) [11] on which the experimental results discussed in section 5 are based on. However, the same technique for evaluating the topology preservation can be used on object FSOM or any FSOM algorithm. For a complete analysis of FRSOM the reader is referred to [11].

Given n input objects $O = \{o_1, \dots, o_n\}$ described by feature vectors $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^s$or by a relational matrix $R = [r_{jk}] = \left[\|x_j - x_k\|^2\right]$[4,11] SOM constructs a lattice or map of c number of neurons (similar to Fig. 1a), that are connected using a neighborhood kernel, h, such the neighborhood between neuron i and j is given by

$$h_{ij} = \exp\left(\frac{-\|a_i - a_j\|^2}{2\sigma^2(t)}\right), \tag{1}$$

where a_i is the coordinate of the ith neuron in the output space (two dimensional space) and σ is a monotonically decreasing neighborhood size. Every neuron has a corresponding s-dimensional weight vector, $m = \{m_1, \dots, m_c\}$ or an n-dimensional coefficient vector in the relational algorithm. One of the goals of the classical crisp SOM algorithm is to assign every s-dimensional input signal, o_k, a winning or a best-matching unit (BMU), w_k, according to

$$w_k = \arg\min_i \|m_i - x_k\|^2 \ \forall 1 \le i \le c \text{ and } 1 \le k \le n . \quad (2)$$

Effectively, SOM assigns a full membership of o_k in neuron w_k,

$$u_{ik} = \begin{cases} 1, \text{if } w_k = i \\ 0, \text{otherwise} \end{cases} . \quad (3)$$

An alternative to this approach is to assign a fuzzy membership for all objects in every neuron as described in [11]. The FRSOM proposed in [11] produces fuzzy partitions $U \in M_{fcn}$ where

$$M_{fcn} = \left\{ U \in \mathbb{R}^{c \times n} \middle| \begin{array}{c} u_{ik} \in [0,1], \\ \sum_{k=1}^{n} u_{ik} > 0, \sum_{i=1}^{c} u_{ik} = 1, \\ \forall\, 1 \le i \le c \text{ and } 1 \le k \le n \end{array} \right\}. \quad (4)$$

Introducing fuzzy memberships to SOM as in FRSOM adds another layer of complexity due to the fact that all neurons are winners of all objects to some degree. Thus, any error measurement made in FRSOM has to factor in all membership values of all input signals in all neurons. In [11] we showed that the quantization error in SOM can be easily adapted to the FRSOM, but this is not the case regarding the topographic error. In the next section we will briefly review two of the major SOM evaluation techniques followed by a new method to evaluate the topology preservation of FRSOM in section 4.

3 Topology Preservation in SOM

Several measures are proposed to measure the goodness of the map. Some measures, such as the quantization error, evaluate the fitness of SOM to the input data. This error calculates the average distance between the input patterns and their corresponding winning neurons [12]. Optimal map is expected to produce a smaller error, which means the input patterns are close to their winning neurons. Quantization error for SOM is shown in (5).

$$qe_c = \sum_{k=1}^{n} \left\| x_k - m_{w_k} \right\| . \quad (5)$$

Similarly, the FSOM quantization error is defined as [11]

$$qe_f = \sum_{i=1}^{c} \sum_{k=1}^{n} u_{ik}^q \left\| x_k - m_{w_i} \right\| . \quad (6)$$

However, the crisp and fuzzy quantization errors in (5) and (6) may not accurately measure the topographic preservation of the map. Instead, one can quantify the relation between the codebook weight vectors and the associated neurons in the map as in the topographic product [12]. This gives a sense on how well the *s*-dimensional

space is mapped to a 2-dimensional lattice [13]. A different approach is to use the topographic error.

The topographic error measures the continuity of the map or how well an input signal preserves the local continuity of the map [12]. When the first and second best-matching units to object o_k are adjacent in the map space, then o_k is said to preserve local map continuity and if they are not adjacent then there is a topological error. To evaluate the overall topology of the map the proportion of input signals for which the first and second best-matching units are not adjacent is measured (7) [12]. A lower error yields a better map and topology.

$$te_c = \sum_{k=1}^{n} adj(o_k), \tag{7}$$

where

$$adj(o_k) = \begin{cases} 1, if\ the\ first\ and\ second\ BMUs\ are\ not\ adjacent \\ 0, otherwise \end{cases}.$$

Another matric for measuring topology preservation in crisp SOM is discussed in [14]. The metric is said to be topology preserving if for any x_i, if x_j is the kth nearest neighbor of x_i, then w_j is the kth nearest neighbor of w_i.

The concept of first and second BMUs is not applicable to FSOM since every unit i is a BMU of every object o_k with a degree u_{ik}. A possible workaround is to harden the fuzzy partition produced by FSOM to find the BMU then compute the topographic error as in (7). Another approach is to consider the two neurons in which o_k has the highest membership as the first and second BMUs. However, neither of these two approaches exploits the membership grade of FSOM. Therefore, a new formulation to measure the local continuity of the map in FSOM is needed to evaluate its goodness and the topology preservation, which is the topic of the next section.

4 Topology Preservation in FRSOM

In FRSOM every neuron is a BMU of every object with a varying degree of membership. Regardless, both the crisp and fuzzy SOM should preserve the topology. Therefore, every pattern presented to FRSOM is also expected to preserve the local continuity of the map. One can consider the first and second neuron with the highest membership to o_k as best and second winning neurons, w_k and w_j. However, this flawed strategy uses only two neurons and discards all other neurons despite the fact other neurons might have high membership to o_k. Relying on two neurons can only give us a false sense of the map continuity. Consider a scenario where the first and second neurons with the highest memberships to o_k, w_k and w_j are immediate neighbors, but the neuron with the third highest membership to o_k is distant from w_k and w_j. A better approach is to use the membership values and utilize all neurons when measuring the topology preservation of FRSOM. More specifically, by looking

at the differences of the membership values between the neurons and their immediate neighbors we can make a conclusion on how well the local topology of the map is preserved.

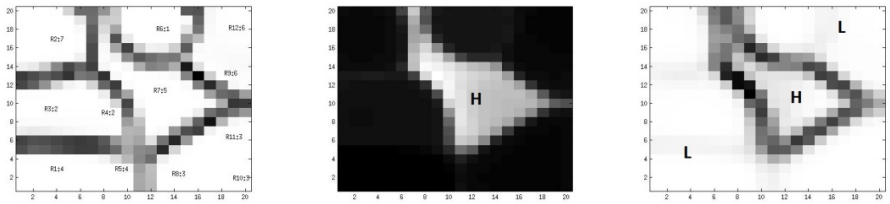
For any given object o_k in FRSOM, we expect neurons with high firing strength to o_k to be concentrated in one region (H region). Also, not all neurons have the same firing strength, as we go further away from the H region, the membership values start to diminish gradually. If the correct data topology is discovered by FRSOM, the H region corresponds to the catchment basin or part of it where o_k belongs the most. In such case, we say that o_k preserves the local continuity of the map. On the other hand, if the neurons of high membership to object o_k are scattered throughout the map or if no H region is identified then the object fails to preserve the topology of the map. For demonstration, Fig. 1a shows the topographic map for Hepta dataset [15] and Fig. 1b shows the H region for some input pattern.

In order to assess how well an object o_k preserves the local continuity of the map we first need to compute the HL-matrix. HL-matrix has the same dimensions as the topographic map and c neurons. A topology preserving HL-matrix includes two main regions, the H region which contains the neurons with high membership to object o_k and the L region containing the rest of the neurons which have low membership values to o_k, as shown in Fig. 1c. Observe that the HL-matrix of o_k represents a snapshot of the U-matrix (Fig. 1a). Adjacent neurons in regions H and L should have similar membership values to o_k. Hence, the difference in the membership values between a neuron i and its immediate neighbors $N(i)$ should be very small with exception to the bordering neurons that separate the H and L regions as shown in Fig. 1c. For a given object o_k we first compute its HL-matrix where the value at every neuron's coordinate is computed as follows

$$HL(i) = \sum_{j \in N(i)} \left| u_{ik} - u_{jk} \right| . \tag{8}$$

$HL(i)$ corresponds to the sum of differences between the membership u_{ik} and the memberships of o_k in $N(i)$. Then that difference is projected on top of the grid position of every neuron. This process is performed for every input pattern. For a small topographic error the value for every neuron $HL(i)$ should be as small as possible, which means that the neuron i and its neighbors $N(i)$ have very similar memberships to the given input pattern.

For an object to preserve the local topology it is imperative that we identify a single region labeled H. Failure in identifying a single region H will cause the topographic error to increase and possibly reaching its maximum value. This technique is stricter than the topographic error in (7). Here we want to ensure that two adjacent neurons have similar membership to o_k, which is somewhat similar to (7), but in addition we would like to ensure that o_k preserves the local continuity within a specific region of the map.



(a) (b) (c)

Fig. 1. (a) FRSOM topographic map of the Hepta dataset, (b) H region for some input pattern o_k and (c) HL-matrix of same input pattern

For more accurate evaluation of the topology preservation it is recommended that we normalize the HL-matrix as follows

$$NHL(i) = \frac{HL(i)}{\sum_{j=1}^{c} HL(j)} . \tag{9}$$

Two important reasons for this normalization: first, it sets an upper bound on the topographic error, similar to (7) the maximum error is 1. Second, normalization is crucial when comparing the topographic errors across different maps. Once the normalized HL-matrix is computed, the final topographic error of a single object o_k will depend on the neurons identified in the region labeled H. The error is simply the sum of values enclosed in the H region of the NHL-matrix (10). As the values in the H region get smaller, so does the topographic error. Meaning that adjacent neurons in the H region share similar memberships to o_k.

$$te_f(k) = \sum_{i \in H} NHL(i) . \tag{10}$$

The final topographic error of the map is computed as the average topographic error overall the objects as

$$te_f = \frac{1}{n} \sum_{k=1}^{n} te_f(k) . \tag{11}$$

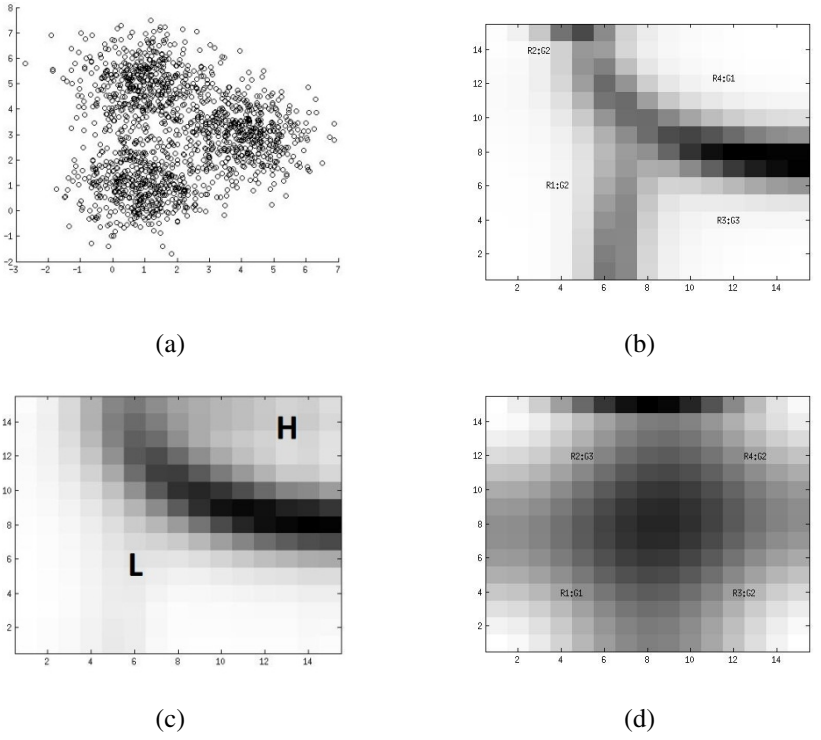
Few remarks to point out about the proposed measure (11): first, the only way for a map to result in a zero topographic error is when the values in the H region are zeros. In other word, when neuron $i \in H$ and its neighbors $N(i)$ have an identical membership to o_k. Second, an HL–matrix may not contain a unique H region. In this situation the topographic error can reach its maximum, which is the sum of all values in the NHL-matrix ($te_f = 1$).

5 Experimental Results

5.1 Fuzzy Topographic Error on O3G

The overlapping three Gaussian (O3G) dataset contains three clusters of size 500 each (Fig. 2a). Clusters in O3G have larger variance which causes overlapping. We setup FRSOM with initial (σ_0), final neighborhood radius (σ_f), initial fuzzifier (q_0), final fuzzifier (q_f), map dimensions and number of epochs to be 2, 0.5, 1, 2, 15x15 and 10, respectively. The resulting topographic map is shown in Fig. 2b.

From Fig. 2c it is clear that the HL-matrix for some given pattern contains the two H and L regions, which is an indication that it preserves the local continuity of the map.

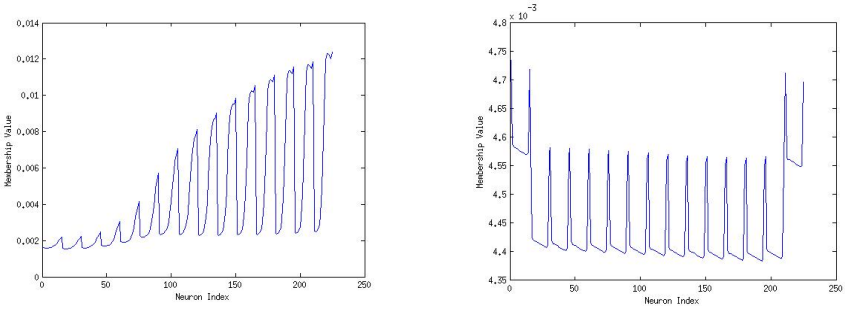


(a) (b)

(c) (d)

Fig. 2. (a) O3G dataset, (b) topographic map produced by FRSOM when $\sigma_0 = 2$, (c) HL-matrix for some object o_k and (d) topographic map produced by FRSOM when $\sigma_0 = 4$

In a topology preserving map, such as the one in Fig. 1c, the membership u_{ik} is expected gradually increase while approaching the H region and neurons with the highest membership should be located within the H region as demonstrated in Fig. 3a. On the other hand, a non-topology preserving map as in Fig. 2d we see a more chaotic



(a) (b)

Fig. 3. (a) Behavior of membership values of o_k in a topology preserving map (membership vs. neuron index), (b) behavior of the membership in a non-topology preserving map ($\sigma_0 = 4$)

membership values among the neurons (Fig. 3b) causing te_f to increase. It could also mean that the four regions or corners in Fig. 2d are wrapped around to form one region representing all input patterns, failing to preserve the topology.

Now, let us compare te_c and te_f for the maps in Fig. 1b and Fig. 1d. If we compute te_c for the map in Fig. 2b (Table 1), where the two neurons with the highest membership value to an input pattern are used as the first and second BMU, we find it higher than the te_c in Fig. 2d (Table 1). On the contrary, te_f has increased from 0.32 in Fig. 1b to $te_f = 1$ in Fig. 1d. In this scenario te_f reveals more information about the goodness of the map resulted from FRSOM since we probably expect Fig. 2b to be more topology preserving than Fig. 2d.

Table 1. Behaviour of te_c and te_f when varying σ_0

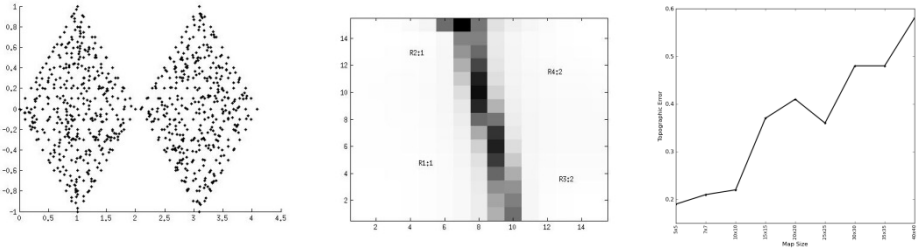
Map	σ_0	te_c	te_f
Fig. 2b	2	0.021 (0.006)	0.32 (0.03)
Fig. 2d	4	0.004 (0.004)	1 (0)

5.2 Fuzzy Topographic Error and Map Dimensions

In this experiment we will use the Two Diamonds dataset from the Fundamental Clustering Problem Suite (FCPS), which contains 800 data points [15] as shown in Fig. 4a. On this dataset we will show how the map dimensions can have an influence on the topographic error. Same parameters used on the O3G dataset will be used for the Two Diamonds with exception to the map dimensions which is set it be 20x20. The resulting topographic map is shown in Fig. 4b.

A smaller map of size 10x10 was also produced for the Two Diamonds dataset. It is not shown since it is very similar to the map in Fig. 4b. We found the overall topological error of the 20x20 map measured to be 0.33. As the map size increases it is likely that the H region increases which in some cases causes an increase in the membership variance among adjacent neurons. On the contrary, 10x10 map might

have lower variance in the memberships among neighboring neurons in the H region and hence a lower topographic error (overall topographic error is 0.28). Overall, as the map size increased the topographic error increased (Fig. 3c). Therefore, it is important to choose a map size suitable for the dataset.



(a) (b) (c)

Fig. 4. (a) Two diamonds dataset, (b) topographic map produced by FRSOM and (c) map size vs. te_f for 5x5, 7x7, 10x10, 15z15, 20x20, 25x25, 30x30, 35x35, 40x40 map dimensions as shown along the x-axis

6 Conclusion

In this paper we presented preliminary results for measuring the topology preservation in fuzzy self-organizing maps. The newly proposed topographic error relies on the membership distribution on the map and in some sense is an extension to the crisp topographic error. The assumption is that adjacent neurons should have similar memberships to a given object o_k. In addition, we presented the HL-matrix. A topology preservation HL-matrix for a given o_k contains two regions, the H region that encompasses the neurons with high membership to o_k and the L region which contains the low membership neurons to o_k. In the results different scenarios were presented to demonstrate how the topographic error behaves when varying the map dimensions. We observed that the topographic error in FSOM tends to be higher than the standard topographic error used in SOM.

One drawback of the proposed measure is its dependence on the map dimensions. For instance, as the map dimensions or size increases so does the topographic error. To overcome this problem, one is expected to specify a map dimension that is suitable to the input dataset. The dependency of the topographic error on the SOM parameters is not necessarily a bad thing. On the contrary, a high topographic error is an indication that the map is not optimal and the parameters require tuning. However, additional experiments are needed to study the influence of other parameters such as the neighborhood size and the fuzzifier, in addition to the map dimensions, on the proposed topographic error.

Furthermore, a more theoretical approach for determining the H region is needed; contrary to the current approach of thresholding the membership values and employing image segmentation to determine the H and L regions.

References

1. Ultsch, A.: Maps for the visualization of high-dimensional data spaces. In: Proc. Workshop on Self organizing Maps (2003), http://www.informatik.uni-marburg.de/~databionics/papers/ultsch03maps.pdf
2. Ritter, H., Kohonen, T.: Self-organizing semantic maps. Biological Cybernetics 61, 241–254 (1989)
3. Havens, T.C., Keller, J.M., Popescu, M., Bezdek, J.C.: Ontological self-organizing maps for cluster visualization and functional summarization of gene products using Gene Ontology similarity measures. In: 2008 IEEE International Conference on Fuzzy Systems (IEEE World Congress on Computational Intelligence), pp. 104–109. IEEE (2008), doi:10.1109/FUZZY.2008.4630351
4. Hasenfuss, A., Hammer, B.: Relational topographic maps. In: Berthold, M.R., Shawe-Taylor, J., Lavrač, N. (eds.) IDA 2007. LNCS, vol. 4723, pp. 93–105. Springer, Heidelberg (2007), http://www.springerlink.com/index/D0664R20V2L83MX5.pdf
5. Kaski, S., Honkela, T., Lagus, K., Kohonen, T.: WEBSOM – Self-organizing maps of document collections. Neurocomputing 21, 101–117 (1998)
6. Corsini, P., Lazzerini, B., Marcelloni, F.: A new fuzzy relational clustering algorithm based on the fuzzy C-means algorithm. Soft Computing 9, 439–447 (2004)
7. Pascual-Marqui, R.D., Pascual-Montano, A.D., Kochi, K., Carazo, J.M.: Smoothly distributed fuzzy c-means: a new self-organizing map. Pattern Recognition 34, 2395–2402 (2001)
8. Chi, S.-C., Kuo, R.-J., Teng, P.-W.: A fuzzy self-organizing map neural network for market segmentation of credit card. In: SMC 2000 Conference Proceedings, 2000 IEEE International Conference on Systems, Man and Cybernetics. Cybernetics Evolving to Systems, Humans, Organizations, and their Complex Interactions, vol. 5, pp. 3617–3622. IEEE (2000)
9. Mohebi, E., Sap, M.N.M.: Hybrid Kohonen Self Organizing Map for the Uncertainty Involved in Overlapping Clusters Using Simulated Annealing. In: 2009 11th International Conference on Computer Modelling and Simulation, pp. 53–58. IEEE (2009), doi:10.1109/UKSIM.2009.28
10. Bezdek, J.C., Tsao, E.C.-K., Pal, N.R.: Fuzzy Kohonen clustering networks. In: 1992 Proceedings of the IEEE International Conference on Fuzzy Systems, pp. 1035–1043. IEEE (1992), doi:10.1109/FUZZY.1992.258797
11. Khalilia, M., Popescu, M.: Fuzzy relational self-organizing maps. In: 2012 IEEE International Conference on Fuzzy Systems, pp. 1–6. IEEE (2012), doi:10.1109/FUZZ-IEEE.2012.6250833
12. Kiviluoto, K.: Topology preservation in self-organizing maps. In: Proceedings of International Conference on Neural Networks (ICNN 1996), vol. 1, pp. 294–299. IEEE (1996)
13. Arsuaga Uriarte, E., Díaz Martín, F.: Topology preservation in SOM. International Journal of Mathematical and Computer Sciences (2005), http://www.ica.luz.ve/~enava/redesn/ebooks/TopologyPreservationInSOM.pdf
14. Bezdek, J.C., Pal, N.R.: An index of topological preservation and its application to self-organizing feature maps. In: Proceedings of 1993 International Conference on Neural Networks (IJCNN 1993), Nagoya, Japan, vol. 3, pp. 2435–2440. IEEE (1993)
15. Ultsch, A.: Clustering with SOM: U* C. In: Proc. Workshop on Self- Organizing Maps, pp. 75–82 (2005), http://www.citeulike.org/group/2572/article/1304144

Designing Type-2 Fuzzy Controllers Using Lyapunov Approach for Trajectory Tracking

Rosalio Farfan-Martinez[1], Jose A. Ruz-Hernandez[2], Jose L. Rullan-Lara[2], Willian Torres-Hernandez[1], and Juan C. Flores-Morales[2]

[1] Universidad Tecnologica de Campeche. Carretera Federal 180 s/n, San Antonio Cardenas, C.P. 24381, Carmen, Campeche. Mexico
{farfan678,williantorreshernandez}@hotmail.com
[2] Universidad Autonoma del Carmen. Calle 56, No. 4 Esq. Avenida Concordia, Col. Benito Juarez, C.P. 24180. Ciudad del Carmen, Campeche, Mexico
{jruz,jrullan}@pampano.unacar.mx, juancafloresm@hotmail.com

Abstract. The paper presents the design of type-2 fuzzy controllers using the *fuzzy Lyapunov synthesis approach* in order to systematically generate the rule base. To construct the rule base, the error signal and the derivative of the error signal are considered. It also presents the performance analysis to determine the value of the separation interval ξ between the upper and lower membership functions of the type-2 fuzzy set used. The controllers are implemented via simulation to solve trajectory tracking problem for angular position and angular velocity of a servo trainer equipment. Simulation results are successful for both cases and shown better performance than those of classical controllers.

1 Introduction

The fuzzy sets were introduced by L. A. Zadeh in the mid-sixties in order to process data affected by non-probabilistic uncertainty [1]. The type-1 fuzzy systems can handle the linguistic variables and experts reasoning and also reproduce the knowledge of systems to control, however, it can not handle uncertainties such as dispersions in linguistic distortion measurements and expert knowledge [2]. On the other hand, type-2 fuzzy systems can handle such kinds of uncertainties and also have the ability to model complex nonlinear systems. In addition, controllers designed using type-2 fuzzy systems achieve better performance than those of type-1. The type-2 fuzzy sets were also originally proposed by Zadeh in 1975 [3].

In [4] a fuzzy logic type-2 based controller using genetic algorithms is performed to control the shaft speed of a DC motor. Genetic algorithms are used to optimize triangular and trapezoidal membership functions. The controller is implemented in a FPGA and its performance is compared with fuzzy logic type-1 and PID controllers.

A type-2 fuzzy controller (T2FC) is designed for an automatic guided vehicle for wall-following in [5]. In this case, T2FC has more robustness to sensor noise and better guidance performance than one of type-1. Another application of

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*, Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_11,

T2FC to mobile robots is presented in [6]. Trajectory tracking is applied first at simulation level and then on a Digital Signal Controller (DSC) of a experimental platform. The reported results show that performance of type-1 controller is poor comparing to type-2 controller.

Some applications of type-2 fuzzy controller in real-time can be also found in literature. For example, the classical inverted pendulum and the magnetic levitation system which are both highly non-linear. In [7], a low-cost microcontroller is used to validate the performance of T2FC for the inverted pendulum. For magnetic levitation system, [8] compared performance of type-1 and type-2 fuzzy controllers and a PID controller. Given that the system is unstable and non-linear, T2FC is showed better performance. Finally, position and velocity type-1 controller are designed in [9]. In this case, stability of both controllers are assured by means of Fuzzy Lyapunov Approach [10]. Results are presented in real time and are compared with classic controllers.

The paper is organized as follows. In section 2 we describe the servo trainer equipment. Then, section 3 presents the control design methodology using the fuzzy Lyapunov approach. Simulation results are presented in section 4. Finally, concluding remarks are presented in section 5.

2 Servo Trainer Equipment

The equipment used as plant to control in this paper is the *CE110 Servo Trainer* from *TQ Education and Training Ltd* [11]. This apparatus is used to help in teaching linear control theory and to implement validate some control algorithms (classical and non classical) in real-time. The equipment have a variable load which is set using a current direct generator, by changes of different inertial load and using the engage a gearbox or by set all of them together. Besides, the apparatus have three modules to introduce some typical nonlinearities.

The mathematical model of servotrainer is set by equations [9]:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\tfrac{1}{T}x_2 + \tfrac{G_1 G_2}{T}u \end{aligned} \tag{1}$$

where $x_1 = \theta$ and $x_2 = \omega$ are the angular position and angular velocity, respectively. The gains G_1 and G_2 are defined by $G_1 = k_i k_\omega$ and $G_2 = k_\theta / 30 k_\omega$ where $k_i = 3.229$ (rev/sec-Volts) is the motor constant, $k_\omega = 0.3$ (Volts/(rev/sec)) is the velocity sensor constant and $k_\omega = 20$ (Volts/rev) is the angle sensor constant. The time constant T change according to size of load: $T = 1.5$ (sec) for small load (one inertial disc); $T = 1$ (sec) for medium load (two inertial discs); $T = 0.5$ (sec) for large load (three inertial discs).

3 Controller Design

3.1 Fuzzy Lyapunov Approach

The goal is to design a control law u such that the velocity and position of servo trainer follows a reference signal y_{ref}. One way of achieving this goal is to choose

a Lyapunov function candidate $V(x)$. Then, this Lyapunov function must meet the following requirements [10]:

$$V(0) = 0 \ , \tag{2}$$

$$V(x) > 0, \ \ x \in N \backslash \{\mathbf{0}\}, \tag{3}$$

$$\dot{V}(x) = \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} \dot{x}_i < 0, \ \ x \in N \backslash \{\mathbf{0}\}. \tag{4}$$

where $N \backslash \{\mathbf{0}\} \in R^n$ is some neighborhood of $\{\mathbf{0}\}$ excluding the origin $\{\mathbf{0}\}$ itself, and $\dot{x}_i$ $(i = 1, 2, \ldots, n)$. If $\{\mathbf{0}\}$ is an equilibrium point of (1) and such $V(x)$ exist, then $\{\mathbf{0}\}$ is locally asymptotically stable.

The conditions (2) and (3) are satisfied by taking such Lyapunov function candidate $V = \frac{1}{2}\left(e^2 + \dot{e}^2\right)$ where e is the tracking error. Differentiating V we have $\dot{V} = e\dot{e} + \dot{e}\ddot{e}$. Substituting $w = \ddot{e}$, is required then:

$$\dot{V} = e\dot{e} + \dot{e}w < 0 \tag{5}$$

Analyzing the equation (5), we can establish four basic fuzzy rules for w such that conditions (4) is satisfied:

- IF e is *positive* AND $\dot{e}$ is *positive* THEN w is *negative big*
- IF e is *negative* AND $\dot{e}$ is *negative* THEN w is *positive big*
- IF e is *positive* AND $\dot{e}$ is *negative* THEN w is *zero*
- IF e is *negative* AND $\dot{e}$ is *positive* THEN w is *zero*

3.2 Type-2 Fuzzy Systems

A fuzzy type-2 system denoted by $^{\approx}A$, is characterized by a membership function type-2 $\mu_{^{\approx}A} = (x, u)$, where $x \in X$, $u \in J_x^u \subseteq [0, 1]$ and $0 < \mu_{^{\approx}A} = (x, u) < 1$. It is defined as follows [12]

$$^{\approx}A = \{(x, \mu_A(x) \mid x \in X)\} = \left[\int_{x \in X} \left[\int_{u \in J_x^u \subseteq [0,1]} f_x(u)/u\right]/x\right] \tag{6}$$

If $f_x(u) = 1$, $\forall u \in [\underline{J}_x^u, \overline{J}_x^u] \subseteq [0, 1]$, membership function type-2 $\mu_{^{\approx}A}$ is expressed by a lower membership function type-1 $\underline{J}_x^u = \underline{\mu}_A(x)$ and upper membership function type-1 $\overline{J}_x^u = \overline{\mu}_A(x)$. Then, $\mu_{^{\approx}A}$ is called an fuzzy type-2 interval, denoted by equation (7)

$$^{\approx}A = \left[\int_{x \in X} \left[\int_{u \in [\underline{\mu}_A(x), \overline{\mu}_A(x)] \subseteq [0,1]} 1/u\right]/x\right] \tag{7}$$

If $^{\approx}A$ is a fuzzy type-2 singleton, then the membership function is defined by equation (8)

$$\mu_{^{\approx}A}(x) = \begin{cases} 1/1, & \texttt{if} \ \ x = x^{'} \\ 1/0, & \texttt{if} x \neq x^{'} \end{cases} \tag{8}$$

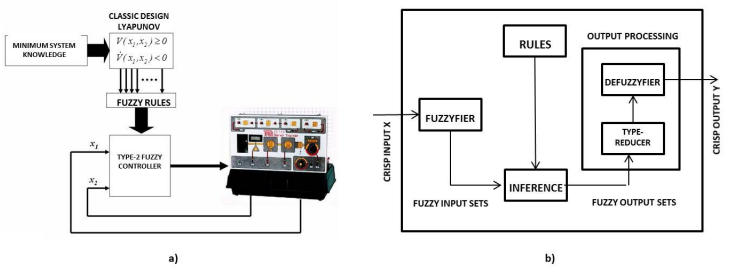


Fig. 1. (a) Control scheme; (b) Components of a type-2 fuzzy system

The type-2 fuzzy systems consist of a fuzzyfier which converts a value from real world into a fuzzy value, a fuzzy inference engine that applies a fuzzy reasoning to obtain a fuzzy output, an output processor comprising a reducer that transforms a fuzzy set type-2 into a fuzzy set type-1 and defuzzyfier which converts a fuzzy value into a precise value (see Fig. 1a).

As mentioned above, membership functions in type-2 fuzzy systems are characterized by having two membership functions of type-1; an upper and a lower membership function. The interval ξ between these two functions can be varied in order to obtain optimal performance. Figure 2a shows such type-2 membership function.

In this paper we have used the Matlab Toolbox developed and described in [12] to implement the type-2 fuzzy system in order to generate values of w. Figures 2b-c shows fuzzy sets for error e, for the derivative of error $\dot{e}$ and for variable w, respectively.

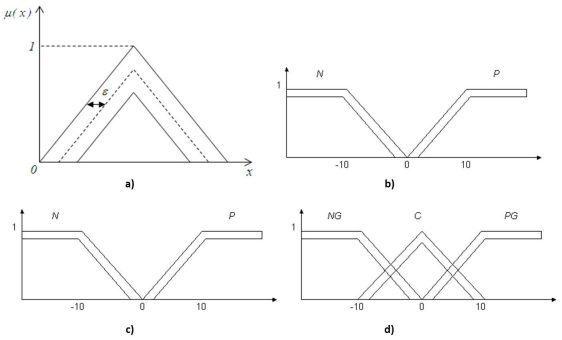


Fig. 2. Type-2 fuzzy sets: (a) Definition of type-2 fuzzy set; (b) Fuzzy set for e; (c) Fuzzy set for $\dot{e}$; (d) Fuzzy set for w

3.3 Mamdani Velocity Controller

The goal is to design a control signal u such that the angular velocity x_2 follows a desired reference signal y_ω. That is $e_\omega \to 0$ as $t \to \infty$ where $e_\omega = x_2 - y_\omega$. According to (5), $\ddot{e}_\omega$ is related to w by $\ddot{e}_\omega = w = \ddot{x}_2 - \ddot{y}_\omega$ and taking into account equation (1), the expression for w is $w_\omega = -\frac{1}{T}\dot{x}_2 + \frac{G_1G_2}{T}\dot{u} - \ddot{y}_\omega$. Finally, solving this equation for control signal u we obtain:

$$u = \frac{T}{G_1G_2}\int\left(\frac{1}{T}\dot{x}_2 + w_\omega + \ddot{y}_\omega\right)dt. \tag{9}$$

3.4 Mamdani Position Controller

The goal is to design a control signal u such that the angular position x_1 follows a desired reference signal y_θ. That is, $e_\theta \to 0$ as $t \to \infty$ where $e_\theta = x_1 - y_\theta$. In this case, $\ddot{e}_\theta$ is related to w by $\ddot{e}_\theta = w = \ddot{x}_1 - \ddot{y}_\theta$. From equation (1), we have that $\ddot{x}_1 = \dot{x}_2$ and the expression for w is $w = -\frac{1}{T}x_2 + \frac{G_1G_2}{T}u - \ddot{y}_\theta$. Then, the control signal u for position tracking is

$$u = \frac{T}{G_1G_2}\left(w_\theta + \ddot{y}_\theta\right) + \frac{1}{G_1G_2}x_2 \tag{10}$$

4 Simulation Results

In this section the *integral of the absolute value of the error* (*IAE*) and the *integral square error* (*ISE*) are used as performance criteria of proposed controllers.

4.1 Velocity Controller

The reference signal that is used is a sinusoidal $y_\omega = 1000\sin 0.4t$ RPM and small load conditions are considered. Ten values of parameter ξ for type-2 fuzzy membership functions vary from 0 to 1 with increments of 0.1. The Table 1 present the ten values obtained for each performance criteria. These results are compared with classic PI controller $u_{PI} = k_p e + k_i \int e dt$ with $k_p = 0.9$ and $k_i = 2.4$ [11].

Table 1. Performance for different values of separation ξ

ξ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
IAE	1.2840	3.6240	5.275	6.597	8.374	7.894	10.38	10.16	12.15	12.92
ISE	0.1022	0.6108	1.271	2.013	3.266	3.204	5.267	5.711	7.904	9.217

Figure 3(Left) shows the trajectory tracking of T2FC designed. Reference signal is well tracked by both controllers. However, PI controller is lagged and its tracking error holds about ± 100 RPM while tracking error obtained with T2FC is bounded by ± 20 RPM and its maximum value is 60 RPM. Is worth mentioning that both control signal is not saturated (see graph on the Up-Right of Fig. 3). In Table 2 the performance of both controller is evaluated by error criterions mentioned above .

4.2 Position Controller

The same loading conditions were considered in the position tracking and the reference signal is the sine signal $y_\theta = 108 \sin 0.3t$ degrees is used. Again, parameter ξ for type-2 fuzzy sets is set to 0.1. Performance of our controller is compared to those of a classical controller with a proportional controller for x_1 ($k_p = 10$) combined with a velocity feedback loop gain with $k_v = 0.01$ [11].

In Figure 4(Left) we can observe the trajectory tracking. Both controllers have acceptable performance in tracking the velocity trajectory. But looking the tracking errors (Up-Right side of Fig. 4) we can observe that at the beginning of the simulation the classical controller oscillate during four seconds whereas the T2FC error converges quickly and smoothly to steady state error value. We can also observe that the signal control from T2FC still bounded and it has not oscillations.

Finally, the Table 2 shows the performance in terms of IAE and ISE error criterions. In both tracking applications, our type-2 fuzzy controllers had proved a good performance of the proposed approach and surpass performance of classical controllers.

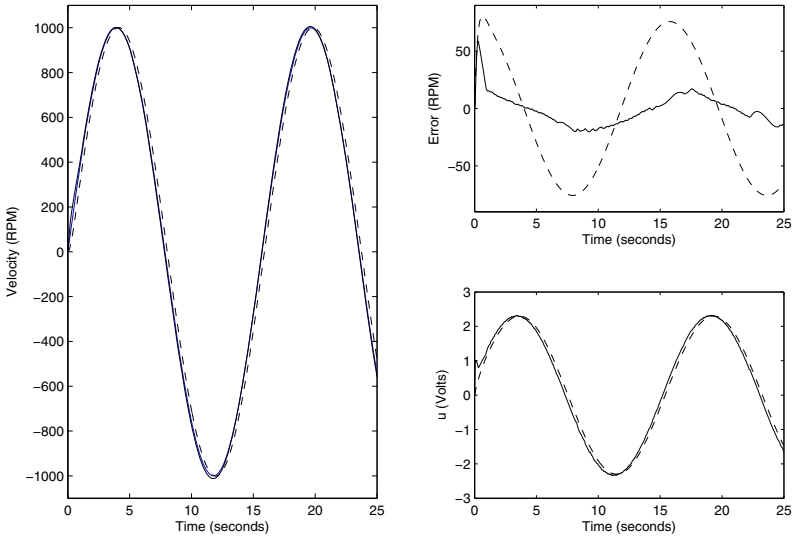


Fig. 3. Velocity Tracking: y_ω (blue line), T2FC (solid line), PI controller (dashed line). Left: Trajectory tracking; Up-Right: error signal; Down-Right: control signal

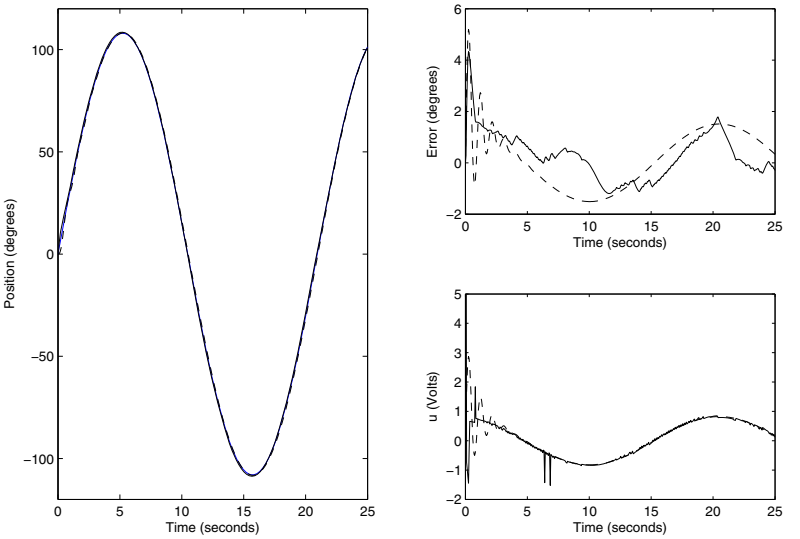


Fig. 4. Position Tracking y_θ (blue line), T2FC (solid line), classical controller (dashed line). Left: Trajectory tracking; Up-Right: error signal; Down-Righ: control signal

Table 2. Performance of trajectory tracking

	Controller	IAE	ISE
Position Tracking	Type-2 ($\xi = 0.1$)	1.025	0.065
	PI ($k_p = 10$, $k_i = 0.01$)	1.409	0.111
Velocity Tracking	Type-2 ($\xi = 0.1$)	1.284	0.1022
	PI ($k_p = 0.9$, $k_v = 2.4$)	6.177	1.871

5 Conclusions

In this paper we have design two type-2 fuzzy controllers using the *fuzzy Lyapunov synthesis approach* in order to systematically generate the rule base. Controllers are designed to solve the position and velocity trajectory tracking problems in a servo trainer system. To tuning the type-2 fuzzy controllers, the separation between upper and lower membership functions is commanded by parameter ξ in steps of 0.1 units. For both controllers, the best tuning was obtained with $\xi = 0.1$.

The performance of our proposed controllers are compared to classical controllers under same simulations conditions for the servo trainer. The *IAE* and *ISE* are used as performance criterions. Simulation results had proved good performances of our proposed approach in position and velocity tracking applications and surpass performances of classical controllers.

Actual research is conducted to test our controllers in medium and full load conditions and also to include nonlinearities that are already available in the servo trainer equipment. In order to demostrate the effectiveness of our approach, authors are motivated to compare performances of type-2 fuzzy controller with performance of type-1 fuzzy controller.

Acknowledgments. The first and the fourth author thank to Universidad Tecnologica de Campeche for financial support and to Faculty of Engineering of the Universidad Autonoma del Carmen for providing the facilities for the use of equipment used in this work.

References

1. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
2. Kwak, H.J., Kim, D.W., Park, G.T.: A New Fuzzy Inference Technique for Singleton Type-2 Fuzzy Logic Systems. International Journal of Advanced Robotic Systems 9, 1–7 (2012)
3. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning-1. Informat. Sci. 8, 199–249 (1975)
4. Maldonado, Y., Castillo, O.: Genetic Design of an Interval Type-2 Fuzzy Controller for Velocity Regulation in a DC Motor. International Journal of Advanced Robotic Systems 9, 1–8 (2012)
5. Yao, L., Chen, Y.S.: Type-2 Fuzzy Control of an Automatic Guided Vehicle for Wall-Following. In: Grigorie, L. (ed.) Theory and Applications, pp. 243–252 (2011) ISBN: 978-953-307-543-3
6. Leottau, L., Melgarejo, M.: An Embedded Type-2 Fuzzy Controller for a Mobile Robot Application. In: Topalov, A. (ed.) Recent Advances in Mobile Robotics, pp. 978–953 (2011) ISBN: 978-953-307-909-7
7. Sierra, G.K., Bulla, J.O., Melgarejo, M.A.: An Embedded Type-2 Fuzzy Processor For The Inverted Pendulum Control Problem. IEEE Latin America Transactions 9(3), 263–269 (2011)
8. Kumar, A., Kumar, V.: Design and Implementation of IT2FLC for Magnetic Levitation System. Advances in Electrical Engineering Systems 1(2), 116–123 (2012)
9. Ruz, J.A., Rullan, J.L., Garcia, R., Reyes, E.A., Sanchez, E.: Trajectory Tracking Using Fuzzy-Lyapunov Approach: Application to a Servo Trainer. In: Castillo, O., Melin, P., Montiel Ross, O., Sepulveda Cruz, R., Pedrycz, W., Kacprzyk, J. (eds.) Theoretical Advances and Applications of Fuzzy Logic and Soft Computing, pp. 710–718. Springer, Heidelberg (2007)
10. Margaliot, M., Langholz, G.: Fuzzy Lyapunov-based approach to the design of fuzzy Controllers. Fuzzy Sets and Systems, Elsevier 106, 49–59 (1999)
11. TecQuipment LTD: CE110 Servo Trainer, User's Manual, England (1993)
12. Castro, J.R., Castilo, O., Melin, P., Martinez, L.G., Escobar, S., Camacho, I.: Building Fuzzy Inference Systems with the Interval Type-2 Fuzzy Logic Toolbox. In: Melin, P., et al. (eds.) Analysis and Design of Intelligent Systems using Soft Computing Techniques, pp. 53–62. Springer, Heidelberg (2007)

Decentralized Adaptive Fuzzy Control Applied to a Robot Manipulator

Roberto Canto Canul[1], Ramon Garcia-Hernandez[1], Jose L. Rullan-Lara[1], and Miguel A. Llama[2]

[1] Universidad Autonoma del Carmen
Ciudad del Carmen, Campeche, Mexico
{rp_robert}@hotmail.com, {rghernandez,jrullan}@pampano.unacar.mx
[2] Instituto Tecnologico de la Laguna
Torreon, Coahuila, Mexico
mllama@itlalaguna.edu.mx

Abstract. In this work it is presented the design of a decentralized adaptive fuzzy control. In this scheme it is suppose a system with unknown parameters and on-line fuzzy identifier, which uses an adaptive law to adjust the unknown parameters in order to build a model of an assumed unknown nonlinear system. The applicability of the proposed approach is illustrated via simulations by trajectory tracking control of a two degrees-of-freedom robot manipulator.

1 Introduction

Robotics and mechatronics represent strategic areas for every country with aspirations to modernity. The scientific development has been growing not only in the amount of products coming to market every day, it also has increased in complexity, this gives its place to the development of new intelligent control techniques because in some cases, classic control theory is not enough to get adequate results in the control of these devices, or simply Intelligent control solves the problem in a better manner.

Fuzzy control is a technique developed based on fuzzy logic and linguistic statements generated from the experience of the system operator to be controlled and not necessarily requires of a mathematical model.

Moreover, adaptive control consist in a control with adjustable parameters and a mechanism for its adjustment on-line [2]. This technique can be combined with fuzzy control obtaining a better performance because the control adjustment is done during the execution of the process and responds to its variations, it also requires less knowledge about the process since the adaptive mechanism could help to find the appropriate fuzzy sets or the fine adjustment automatically, in other words, it helps to learn the plant dynamics.

Fuzzy adaptive control is based on the input-output feedback linearization technique, which is often used in fuzzy systems with adaptive parameters either as a function approximator or in order to compute the control law with minimum knowledge of the plant.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_12,

Finally, there is another technique known as decentralized control which main feature is that once identified the stages of the process, we design n controllers corresponding to each stage; making this, we give robustness to the process because the responsibility is not only for one single controller but is distributed in each one of them. Another advantage is the flexibility on the system because if we need to increase or decrease the number of process or degrees-of-freedom of the system, we only need to add a proportional number of stages of control without affecting the stages already working at the time.

Trajectory tracking is a very common problem for the control engineers not only in industry but also in educational institutions, several methods have been developed through the years but in this section only some examples with the adaptive fuzzy method will be mentioned: in [3] it is designed a centralized adaptive fuzzy controller for trajectory tracking and its performance is tested in a two degrees-of-freedom robot manipulator in simulation; the main objective in the mentioned work is to prove the adaptive part of the controller by changing parameters during the operation of the plant simulating the possible variations in a real robot. Another contribution in this field is shown in [4], where the controller designed in [3] is implemented in a real robot manipulator and the testing is done in real-time obtaining favorable results in trajectory tracking. In [5] it is combined the adaptive fuzzy control with neural networks and other techniques, for designing a decentralized adaptive fuzzy applied to a 2DOF robot manipulator, the main difference between the 2 works mentioned before is an identification block with a neural network to learn on-line the dynamics for the plant.

In the first section of this work, it is presented an introduction of decentralized adaptive fuzzy control and the reasons why it is interesting its study. In the second section is described a type of decentralized system which is applied in the testing of the controller designed. In section 3, the mathematical sustenance for the decentralized adaptive fuzzy control is presented. In section 4 this control is applied to a two DOF robot manipulator and the results in trajectory tracking are presented in section 5.

2 Decentralized Systems

Let consider a large-scale system which is constituted of nonlinear subsystems in the nonlinear block-controllable (NBC) form with disturbance term consisting of r blocks [6]:

$$\begin{aligned}
\dot{x}_i^1 &= f_{i1}(x_i^1) + B_{i1}(x_i^1)x_i^2 + \Gamma_{i1k}(\bar{x}_k^1) \\
\dot{x}_i^2 &= f_{i2}(x_i^1, x_i^2) + B_{i2}(x_i^1, x_i^2)x_i^3 + \Gamma_{i2k}(\bar{x}_k^1, \bar{x}_k^2) \\
&\vdots \\
\dot{x}_i^q &= f_{iq}(x_i^1, x_i^2, \ldots, x_i^q) + B_{iq}(x_i^1, x_i^2, \ldots, x_i^q)x_i^r \\
&\quad + \Gamma_{iqk}(\bar{x}_k^1, \bar{x}_k^2, \ldots, \bar{x}_k^q) \\
\dot{x}_i^r &= f_{iq}(x_i^1, x_i^2, \ldots, x_i^q) + B_{iq}(x_i^1, x_i^2, \ldots, x_i^q)x_i^r \\
&\quad + \Gamma_{iqk}(\bar{x}_k^1, \bar{x}_k^2, \ldots, \bar{x}_k^q)
\end{aligned} \tag{1}$$

where q=3,...,r=1

$$\dot{x}_i^r = f_{ir}(x_i) + B_{ir}(x_i)u_i + \Gamma_{irk}(\bar{x}_i) \tag{2}$$

where $x_i = [x_i^{1T} x_i^{2T} \dots x_i^{rT}]^T$, $x_i^q \in R^{n_{iq}\text{x}1}$, $\bar{x}$ is the state vector of the k-th subsystem ($i = 1, \dots N, 1 \leq k \leq N, k \neq i$) and the rank of $B_{iq} = n_{iq}, \forall x_i^q \in D_{x_i^q} \subset R^{n_{iq}}$. The interconnection terms

$$\begin{aligned}
\Gamma_{i1k} &= \sum_{k=1,k\neq i}^{N} \gamma_{i1k}(\bar{x}_k^1) \\
\Gamma_{i2k} &= \sum_{k=1,k\neq i}^{N} \gamma_{i2k}(\bar{x}_k^1, \bar{x}_k^2) \\
&\vdots \\
\Gamma_{iqk} &= \sum_{k=1,k\neq i}^{N} \gamma_{iqk}(\bar{x}_k^1, \bar{x}_k^2, \dots, \bar{x}_k^q) \\
\Gamma_{i1k} &= \sum_{k=1,k\neq i}^{N} \gamma_{irk}(\bar{x})
\end{aligned} \tag{3}$$

reflect the interaction between the i-th and k-th subsystem; they are bounded by nonlinear functions γ_{irk} and enter the system as no matching condition disturbances, f_{ir} and B_{ir} are smooth and bounded functions, $f_{ir}(0) = 0$ and $B_i(0) = 0$. The integers $n_{i1} \leq n_{i2} \leq ... \leq n_{ir}$ define the different subsystem structures, and $\sum_{q=1}^{r} n_{iq} = n_i$.

3 Design of the Decentralized Adaptive Fuzzy Control

An indirect adaptive fuzzy controller consists in an input-output linearization scheme which starts by finding a direct relation between the output y and the control signal u. For this, there is a methodology described in [7], [8], [9], among others, where it is specified the necessary conditions to obtain an input-output linearization and so, guarantee that the system is linear and controllable, from a coordinate transformation and a state feedback. For the case of the decentralized adaptive fuzzy controller it will be considered a n order nonlinear system described by

$$\begin{aligned}
\dot{x}_i^1 &= x_i^2 \\
\dot{x}_i^2 &= x_i^3 \\
&\vdots \\
\dot{x}_i^{n-1} &= f_{n-i}(x^{n-1}) + g_i(x^{n-1})x^{n-1} \\
\dot{x}_i^n &= f_n(x^n) + g_i(x^n)u \\
y &= x
\end{aligned} \tag{4}$$

which has input-output linearization, is in the normal form and has relative degree $r{=}n$. Applying the control law

$$u_i = \frac{1}{g_i(x_i)}\left[-f_i(x_i) + {k_i}^T e_i + y_m^{(n)}\right] \tag{5}$$

with $k_i = [k_{in}...k_{i1}]^T$ and the error $e_i = y_{im} - x_i$ and $e_i = [e_i^1 e_i^2 ... e_i^n]^T = [e_i \dot{e}_i ... e_i^{(n-1)}]^T$, then the equation in closed-loop turns to

$$e_i^{(n)} + k_{i1} e_i^{(n-1)} + ... + k_{in} e_i = 0 \tag{6}$$

$$e_i^{(n)} + {k_i}^T e_i = 0 \tag{7}$$

We suppose the functions $f_i(x_i)$ and $g_i(x_i)$ are unknown, so the adaptive fuzzy controller will approximate them using the functions $\hat{f}_i(x_i)$ and $\hat{g}_i(x_i)$ making this, the control law in (5) become

$$u_i = \frac{1}{\hat{g}_i(x_i)}\left[-\hat{f}_i(x_i) + {k_i}^T e_i + y_m^{(n)}\right] \tag{8}$$

Due to these functions are universal approximators and they can approximate uniformly the nonlinear function with an arbitrary precision, we use fuzzy systems to obtain the necessary functions $\hat{f}_i(x_i)$ and $\hat{g}_i(x_i)$ in the control law, where

$$\hat{f}_i(x_i) = \frac{\sum_{l_1}^{p_1} \cdots \sum_{l_n}^{p_n} \bar{y}_{f_i}^{l_1 \ldots l_n} \left(\prod_{r=1}^{n} \mu_{A_r^{l_r}}(x_r)\right)}{\sum_{l_1}^{p_1} \cdots \sum_{l_n}^{p_n} \left(\prod_{r=1}^{n} \mu_{A_r^{l_r}}(x_r)\right)} \tag{9}$$

$$\hat{g}_i(x_i) = \frac{\sum_{l_1}^{q_1} \cdots \sum_{l_n}^{q_n} \bar{y}_{g_i}^{l_1 \ldots l_n} \left(\prod_{r=1}^{n} \mu_{B_r^{l_r}}(x_r)\right)}{\sum_{l_1}^{q_1} \cdots \sum_{l_n}^{q_n} \left(\prod_{r=1}^{n} \mu_{B_r^{l_r}}(x_r)\right)} \tag{10}$$

The fuzzy systems in (9) and (10) have singleton fuzzifier, max-prod inference and center of area defuzzifier. In order to improve the approximations results, some free parameters are proposed, these parameters can change on-line with the operation of the system. These free parameters replace the output fuzzy sets which normally are fixed in a non adaptive scheme. Replacing $\bar{y}_{if}^{l_1 \ldots l_n}$ with the free parameter $\theta_{if} \in \prod_{r=1}^{n} p_m$ in $\hat{f}_i(x_i)$; and replacing $\bar{y}_{ig}^{l_1 \ldots l_n}$ with the free parameters $\theta_{ig} \in \prod_{i=1}^{n} q_i$ in $\hat{g}_i(x_i)$, then (9) and (10) can be written as

$$\hat{f}_i(x_i|\theta_{if}) = |\theta_{if}^T \xi_i(x_i) \tag{11}$$

$$\hat{g}_i(x_i|\theta_{ig}) = |\theta_{ig}^T \eta_i(x_i) \tag{12}$$

with the vectors $\xi_i(x_i)$ of dimension $\prod_{r=1}^{n} p_r x 1$, and $\eta_i(x)$ of dimension $\prod_{r=1}^{n} q_r x 1$ which elements are

$$\xi_{i\, l_1 \ldots l_n}(x_i) = \frac{\prod_{r=1}^{n} \mu_{A_r^{l_r}}(x_r)}{\sum_{l_1}^{p_1} \cdots \sum_{l_n}^{p_n} \left(\prod_{r=1}^{n} \mu_{A_r^{l_r}}(x_r)\right)} \tag{13}$$

$$\eta_{i\,l_1\ldots l_n}(x_i) = \frac{\prod_{r=1}^{n} \mu_{B_r^{l_r}}(x_r)}{\sum_{l_1}^{q_1}\cdots\sum_{l_n}^{q_n}\left(\prod_{r=1}^{n} \mu_{B_r^{l_r}}(x_r)\right)} \tag{14}$$

so, the control law for the decentralized adaptive fuzzy control is defined for the i-th subsystem by

$$u_i = \frac{1}{\hat{g}_i(x_i\,|\theta_{ig}\,)}\left[-\hat{f}_i(x_i|\theta_{\,if}) + {k_i}^T e_i + y_{i\,m}^{(n)}\right] \tag{15}$$

3.1 Adaptive Law

The mentioned free parameters θ_{if} and θ_{ig} are obtained with the methodology described in [2]. Substituting equation (15) in (4), it is obtained the closed-loop equation for the i-th subsystem, described by

$$e_i^{(n)} = -{k_i}^T e_i + \left[\hat{f}_i(x_i|\theta_{\,if}) - f_i(x_i)\right] + \left[\hat{g}_i(x_i\,|\theta_{ig}\,) - g_i(x_i)\right] u_i \tag{16}$$

Let consider

$$\Lambda_i = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -k_{i\,n} & -k_{i\,n-1} & -k_{i\,n-2} & \cdots & -k_{i1} \end{bmatrix}, \qquad b_i = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{17}$$

so (16) is written in vectorial form as

$$\dot{e}_i = \Lambda_i e_i + b_i\left\{\left[\hat{f}_i(x_i|\theta_{\,if}) - f_i(x_i)\right] + \left[\hat{g}_i(x_i\,|\theta_{ig}\,) - g_i(x_i)\right] u_i\right\} \tag{18}$$

for n=1, the optimal parameters are

$$\theta_{if}^* = \arg_{\theta_{if}\in^{\min}_{\prod_{r=1}^{n}\,p_i}}\left[\sup_{X\,\in\,R}\left|\hat{f}_i(x_i|\theta_{\,if}) - f_i(x_i)\right|\right] \tag{19}$$

$$\theta_{ig}^* = \arg_{\theta_{ig}\in^{\min}_{\prod_{r=1}^{n}\,q_i}}\left[\sup_{X\,\in\,R}\left|\hat{g}_i(x_i\,|\theta_{ig}\,) - g_i(x_i)\right|\right] \tag{20}$$

So $\hat{f}_i\left(x_i\left|\theta_{if}^*\right.\right)$ and $\hat{g}_i\left(x_i\left|\theta_{ig}^*\right.\right)$ are the best max-min approximators among all the ones obtained from (9) and (10). Defining the approximation error for the i-th subsystem as

$$w_i = \left[\hat{f}_i(x_i\,|\theta_{if}^*\,) - f_i(x_i)\right] + \left[\hat{g}_i(x_i\,|\theta_{ig}^*\,) - g_i(x_i)\right] u_i \tag{21}$$

so, the equation (18) can be written as

$$\begin{aligned}\dot{e}_i &= \Lambda_i e_i + b_i \left\{ \left[\hat{f}_i(x_i|\theta_{if}) - \hat{f}_i(x_i \left|\theta^*_{if}\right.) \right] + \left[\hat{g}_i(x_i\,|\theta_{ig}) - \hat{g}_i(x_i\,|\theta^*_{ig}) \right] u_i \right\} \\ \dot{e}_i &= \Lambda_i e_i + b_i \left[\left(\theta_{if} - \theta^*_{if} \right) \xi_i(x_i) + \left(\theta_{ig} - \theta^*_{ig} \right) \eta_i(x_i) u_i + w_i \right] \end{aligned} \tag{22}$$

In Fig.1 it is presented the block diagram for a decentralized adaptive fuzzy control scheme.

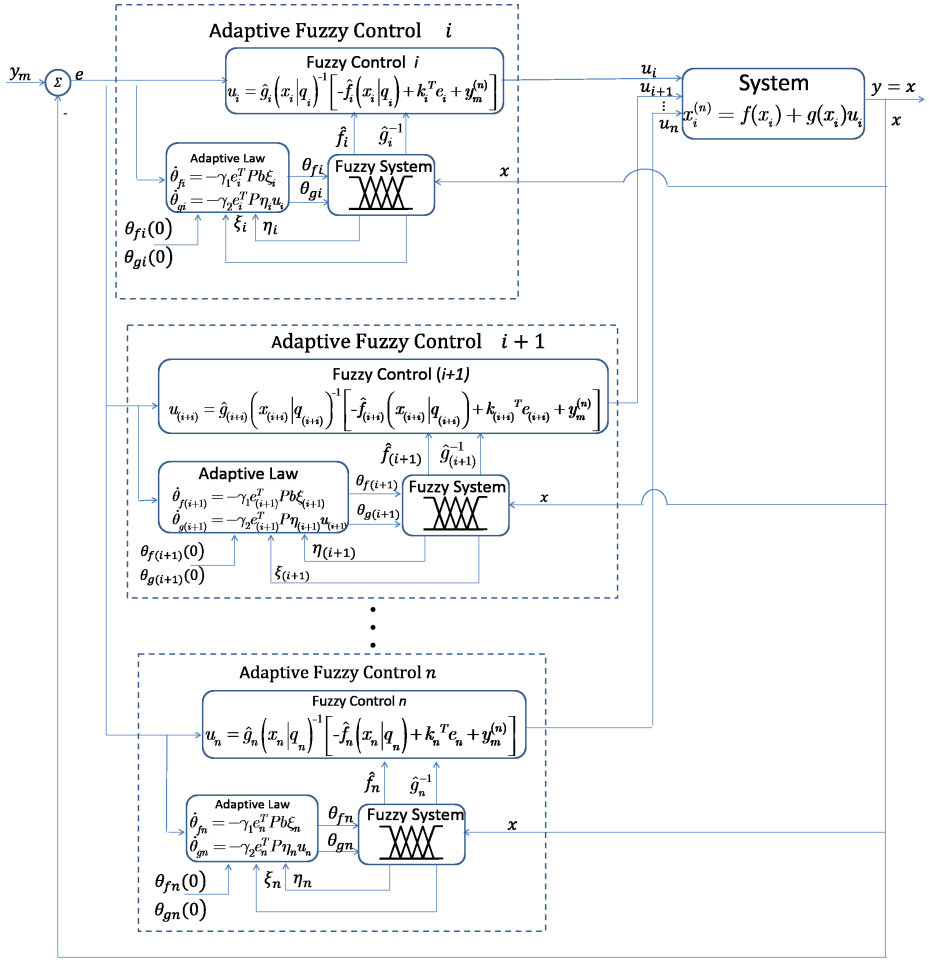


Fig. 1. Block Diagram for the Decentralized Adaptive Fuzzy Control

4 Application to a Two Degrees-of-Freedom Robot Manipulator

4.1 Description of the Robot Manipulator

In order to evaluate in simulation the performance of the proposed control algorithm, we use a two DOF robot manipulator moving in the vertical plane. This robot is armed with two electromechanical actuators whose allow the robot to move and also indicate the angular position of the joints. The dynamic model for a n degrees-of-freedom robot is given by the equation [10].

$$\tau = M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) + \tau_f(\dot{q}) \tag{23}$$

where $M(q)$ is the inertia matrix which is positive defined and symmetric, $C(q,\dot{q})$ is the matrix of Centripetal Forces and Coriolis, $g(q)$ is the gravitational torque vector, $\tau_f(\dot{q})$ is the vector of friction including the viscous friction,τ is the vector of external torque applied to the robot and q, $\dot{q}$, $\ddot{q}$ are the vectors of position, velocity and acceleration, respectively. Accelerations vector is as follows

$$\ddot{q} = -M(q)^{-1}[C(q,\dot{q})\dot{q} + g(q) + \tau_f(\dot{q})] + M(q)^{-1}\tau \tag{24}$$

which is in the form (4), this system has input-output linearization and the decentralized adaptive fuzzy control can be expressed as

$$\ddot{q}_i = F_i(q,\dot{q}) + G_i(q,\dot{q})\tau_i \tag{25}$$

where $F_i(q,\dot{q}) = -M(q)^{-1}[C(q,\dot{q})\dot{q} + g(q) + \tau_f(\dot{q})]$ and $G_i(q,\dot{q}) = -M(q)^{-1}$, then the control law by input-output linearization in (5) applied to a two DOF robot manipulator is

$$\tau = G_i(q_i,\dot{q}_i)^{-1}\left[\ddot{q}_{di} + k_v\ddot{\tilde{q}}_i + k_p\dot{\tilde{q}}_i - F_i(q_i,\dot{q}_i)\right] \tag{26}$$

$$\tau = M(q)\left[\ddot{q}_{di} + k_v\ddot{\tilde{q}}_i + k_p\dot{\tilde{q}}_i - F_i(q_i,\dot{q}_i)\right] + C(q,\dot{q})\dot{q} + \tau_f(\dot{q}) + g(q) \tag{27}$$

The equation (27) is known as "Computed Torque Control" [11]. The functions $f_i(x_i)$ and $g_i(x_i)$ are unknown, replacing them with the fuzzy systems described in (11) and (12), we obtain the control law for the i-th link of the robot

$$\tau = \hat{G}_i(q_i,\dot{q}_i)^{-1}\left[\ddot{q}_{di} + k_v\ddot{\tilde{q}}_i + k_p\dot{\tilde{q}}_i - \hat{F}_i(q_i,\dot{q}_i)\right] \tag{28}$$

for i=1,2 the approximators are

$$\begin{aligned}\hat{F}_1(q,\dot{q}\,|\theta_{1f}) &= \hat{f}_1(q_1,\dot{q}_1\,|\theta_{1f})\\ \hat{F}_2(q,\dot{q}\,|\theta_{2f}) &= \hat{f}_2(q_2,\dot{q}_2\,|\theta_{2f})\end{aligned} \tag{29}$$

$$\begin{aligned}\hat{G}_1(q_1,\dot{q}_1\,|\theta_{1g}) &= \left[\hat{g}_{11}(q_1,\dot{q}_1\,|\theta_{g11})\ \hat{g}_{12}(q_1,\dot{q}_1\,|\theta_{g12})\right]\\ \hat{G}_2(q_2,\dot{q}_2\,|\theta_{2g}) &= \left[\hat{g}_{21}(q_2,\dot{q}_2\,|\theta_{g21})\ \hat{g}_{i22}(q_2,\dot{q}_2\,|\theta_{g22})\right]\end{aligned} \tag{30}$$

4.2 Design Parameters

Applying the decentralized adaptive fuzzy control, the closed-loop system for each link of the robot reduces to linear equations where k_{ip} and k_{iv} determine the pole placement, which are selected on the left side of the plane S so the system is stable. In this case, the position and velocity constants are selected as:

$$\begin{array}{ll} k_{ip1} = 13264[1/seg^2] & k_{iv1} = 230[1/seg^2] \\ k_{ip2} = 16074[1/seg^2] & k_{iv2} = 240[1/seg^2] \end{array}$$

Finally applying the control law to a two DOF robot we obtain

$$\ddot{\tilde{q}}_i = -k_{iv}\dot{\tilde{q}}_i - k_{ip}\tilde{q}_i + \hat{F}_i(q_i, \dot{q}_i \,|\theta_{if}) + \hat{G}_i(q_i, \dot{q}_i \,|\theta_{ig})\tau_i - F_i(q_i, \dot{q}_i) + G_i(q_i, \dot{q}_i)\tau_i$$

which can be represented as

$$\frac{d}{dt}\begin{bmatrix} \tilde{q}_i \\ \dot{\tilde{q}}_i \end{bmatrix} = \dot{e}_i = A_i e_i + B_i \quad \text{where} \quad e_i = \begin{bmatrix} \tilde{q}_i \\ \dot{\tilde{q}}_i \end{bmatrix} = \begin{bmatrix} q_{di} - q_i \\ \dot{q}_{di} - \dot{q}_i \end{bmatrix},$$

$$\text{and} \quad A_i = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\text{diag}\,(k_{ip}) & -\text{diag}\,(k_{iv}) \end{bmatrix} \text{ with } \mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ \mathbf{0} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix},$$

$$\text{and} \quad B_i = \begin{bmatrix} 0 \\ \hat{F}_i(q_i, \dot{q}_i \,|\theta_{if}) - \hat{F}_i(q_i, \dot{q}_i \,|\theta^*_{if}) + \left[\hat{G}_i(q_i, \dot{q}_i \,|\theta_{ig}) - \hat{G}_i(q_i, \dot{q}_i \,|\theta^*_{ig})\right]\tau_i + w_i \end{bmatrix}$$

The decentralized adaptive fuzzy control scheme applied to a two degrees-of-freedom robot manipulator is summarized in the Fig.2.

4.3 Desired Trajectories

For simulation, the following trajectories were selected [11].

$$\begin{bmatrix} q_{1d}(t) \\ q_{2d}(t) \end{bmatrix} = \begin{bmatrix} b_1(1 - e^{-d_1 t^3}) + c_1(1 - e^{-d_1 t^3})\sin(w_1 t) \\ b_2(1 - e^{-d_2 t^3}) + c_2(1 - e^{-d_2 t^3})\sin(w_2 t) \end{bmatrix} \tag{31}$$

where $b_1 = \pi/4[rad]$, $c_1 = \pi/18[rad]$, $d_1 = 2$ and $w_1 = 15[rad/s]$ are parameters for the desired position in the first joint of the robot; $b_2 = \pi/3[rad]$, $c_2 = 25\pi/36[rad]$, $d_2 = 1.8$ and $w_2 = 1.75[rad/s]$ are parameters for the desired position in the second joint.

5 Simulation Results

In Fig.3(a) and Fig.3(b), the trajectory tracking results for the robot with the decentralized adaptive fuzzy control are presented. In Fig.4(a) and Fig.4(b), the tracking error for each joint of the robot manipulator are presented. Finally, in Fig.5(a) and Fig.5(b) the applied torques to each joint are shown.

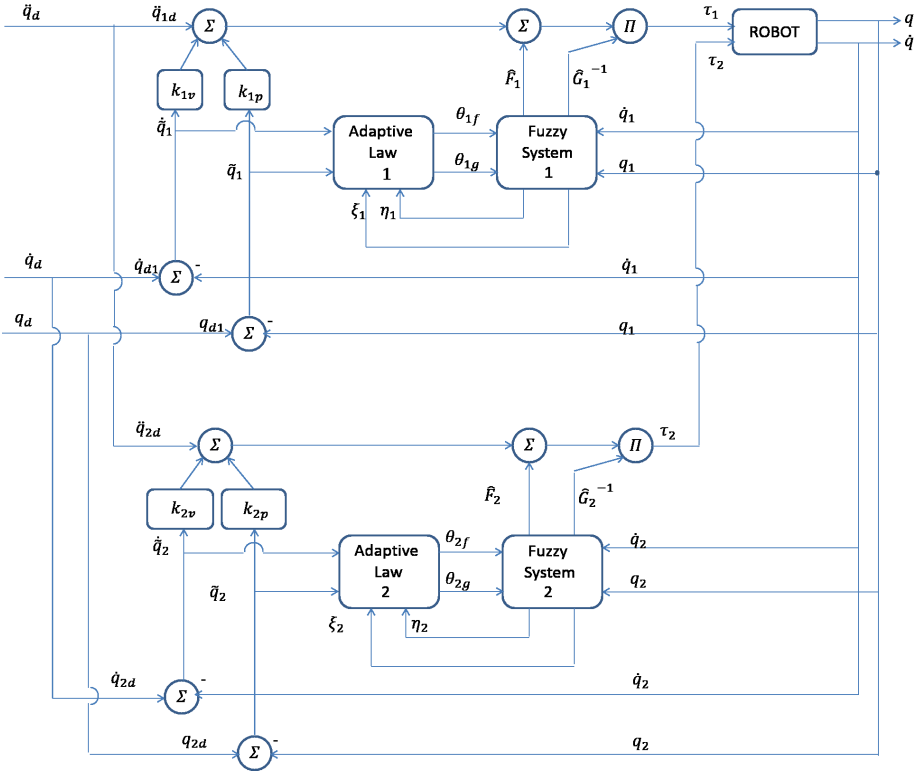
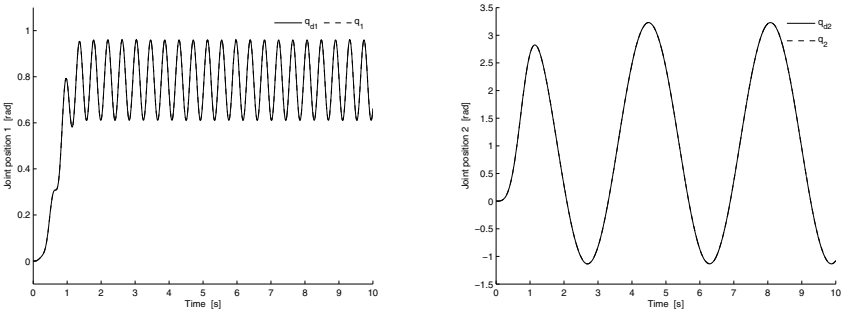
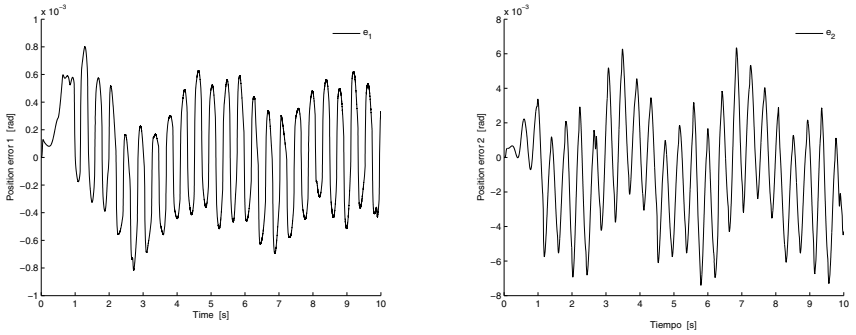


Fig. 2. Block diagram of the Controller applied to a 2 DOF robot manipulator



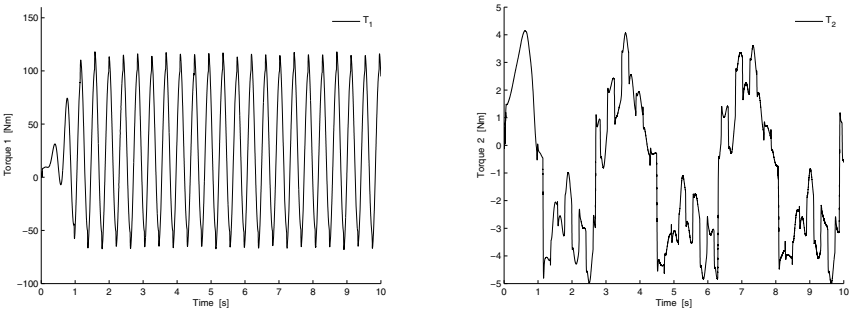
(a) Joint 1 (b) Joint 2

Fig. 3. Trajectory tracking for joints 1 and 2



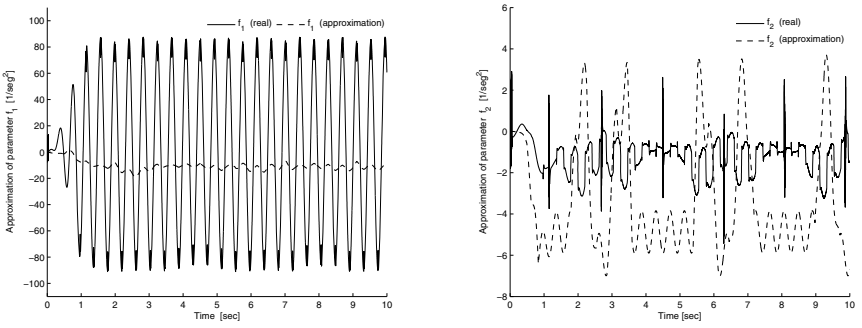
(a) Joint 1 (b) Joint 2

Fig. 4. Tracking errors for joints 1 and 2



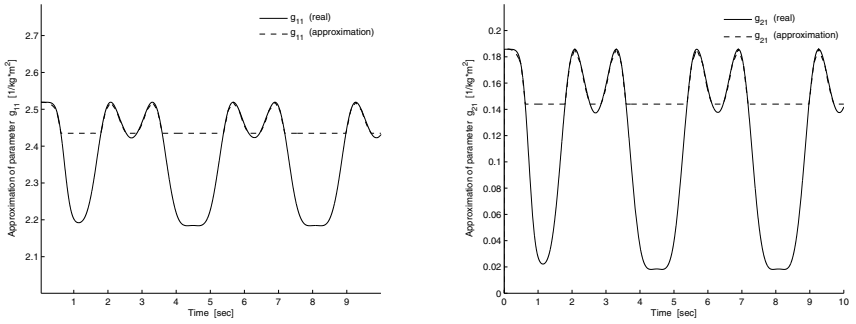
(a) Joint 1 (b) Joint 2

Fig. 5. Applied torque to joints 1 and 2



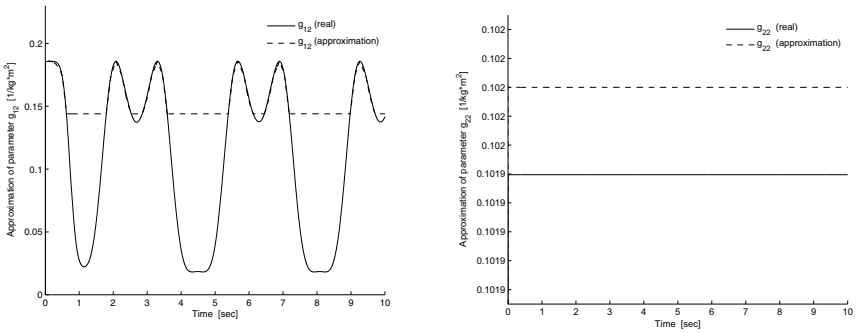
(a) $\hat{f}_1$ (b) $\hat{f}_2$

Fig. 6. Approximations of $\hat{f}_i(x_i)$



(a) $\hat{g}_{11}$ (b) $\hat{g}_{21}$

Fig. 7. Approximations of $\hat{g}_i(x_i)$



(a) $\hat{g}_{12}$ (b) $\hat{g}_{22}$

Fig. 8. Approximations of $\hat{g}_i(x_i)$

The fuzzy system in the controller is used to approximate the unknown parameters in the functions $\hat{f}_i(x_i)$ and $\hat{g}_i(x_i)$. The approximations computed for each joint by the fuzzy systems are presented in Fig.6(a) and Fig.6(b) corresponding to the approximations of the elements of $\hat{f}_i(x_i)$ and Fig.7(a), Fig.7(b); Fig.8(a) and Fig.8(b) corresponding to the elements of $\hat{g}_i(x_i)$.

A performance analysis for the controller using the Euclidean norm or $\mathcal{L}_2$ norm, is defined by

$$\mathcal{L}_2[e] = \sqrt{\frac{1}{T - t_0} \int_{t_0}^{T} e^T e \, dt} \tag{32}$$

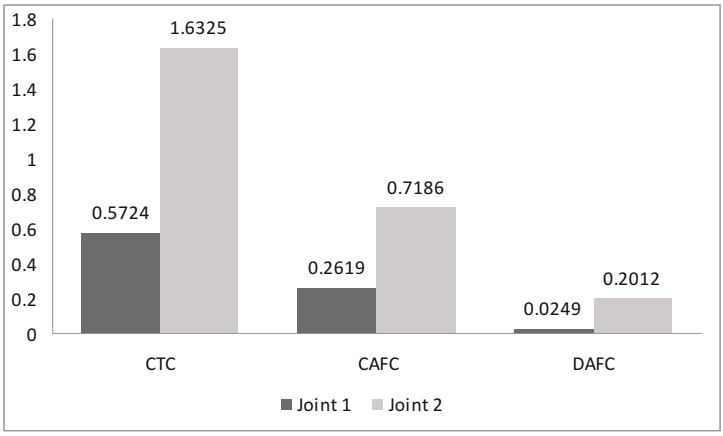


Fig. 9. $\mathcal{L}_2$ norm for each joint with different control techniques

this norm was applied to the tracking error in each joint of the robot manipulator. For this case, $\mathcal{L}_2$ norm is computed as

$$\mathcal{L}_2[e_i] = \sqrt{\frac{1}{T - t_0} \int_{t_0}^{T} {e_i}^2 \, dt} \tag{33}$$

where $e_i = q_{id} - q_i$

Equation (33) was used for testing the performance of the designed controller, we compare the results obtained applying the $\mathcal{L}_2$ norm to the Decentralized adaptive fuzzy control (DAFC) with the results obtained with another 2 techniques: "Computed Torque Control" (CTC) and "Centralized adaptive fuzzy control" (CAFC). The results in Fig.9 how that tracking errors for the controller designed in this paper decreases to 20% in comparison with the other controllers.

6 Conclusions

In this work a decentralized adaptive fuzzy control was designed. The results for trajectory tracking applied to a two DOF robot manipulator were presented, obtaining satisfactory results. The fuzzy part of the controller is responsible of estimate the internal parameters of the robot whose are supposed unknown. This approach allows us to control the plant even without prior knowledge. These estimations may differ from the real parameters without affecting the control objective.

Acknowledgments. First author acknowledges the scholarship from CONACyT-Mexico to pursue his postgraduate studies. Authors thank to Universidad Autonoma del Carmen (UNACAR), Instituto Tecnologico de la Laguna (ITL) and Programa de Mejoramiento del Profesorado (PROMEP) for the support given in this research work.

References

1. Wang, L.: A course in fuzzy systems and control, USA (1992)
2. Jang, J., Sun, C., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A computational approach to learning and machine intelligence. Prentice Hall (1997)
3. Tellaeche, J.: Diseño de un controlador difuso adaptable para un robot manipulador de dos grados de libertad de transmision directa. Instituto Tecnologico de la Laguna, Mexico (2003)
4. Centeno, H.: Control difuso adaptable de sistemas mecatrónicos: Aplicacion experimental a un carro péndulo y a un robot manipulador de 2GDL. Instituto Tecnologico de la Laguna, Mexico (2008)
5. Garcia-Hernandez, R., Sanchez, E., Bayro-Corrochano, E., Santibanez, V., Ruz-Hernandez, J.: Real-time decentralized neural block control: application to a two DOF robot manipulator. International Journal of Innovative Computing, Information and Control 7(3), 1075–1085 (2011)
6. Benitez, V., Sanchez, E., Loukianov, A.: Decentralized adaptive recurrent neural control structure. Engineering Applications of Artificial Intelligence 20(8), 1125–1132 (2007)
7. Slotine, J., Wang, L.: Applied nonlinear control. Prentice Hall, USA (1991)
8. Khalil, H.: Nonlinear systems. Prentice Hall, USA (2002)
9. Isidori, A.: Nonlinear control systems. Springer, Great Britain (1995)
10. Spong, M., Hutchinson, S., Vidyasagar, M.: Robot modelling and control. John Wiley & Sons Inc. (2005)
11. Kelly, R., Santibañez, V.: Control de Robots Manipuladores. Prentice Hall, Madrid (2003)

Modeling, Planning, Decision-Making and Control in Fuzzy Environment

Ben Khayut, Lina Fabri, and Maya Abukhana

Department of R&D at Intelligence Decisions Technologies Systems, IDTS Israel

Abstract. The use of proposed technology is oriented for *persons*, who want *to interact* with systems *to make relevant decisions* in real-time, *fuzzy conditions*, heterogeneous *subject areas* and *multi-lingual communication*, where the *situations are unknown in advance*, *fuzzy structured* and *not clearly regulated*. The *essence* of the technology consists in the *situational control* of fuzzy *data*, *information* and *knowledge*, extracted from texts in different *natural languages*, dissimilar *subject areas* for *situational fuzzy control* of the object in *Intelligent* real-time system. The technology is formalized using *Fuzzy Logic*, *Situational Control* theories and is defined by methods of *knowledge representation*, *situational data control*, *fuzzy logic inference*, *knowledge modeling*, *generalization* and *explanation knowledge*, *dialogue control*, *machine translation* and others.

1 Introduction

In order to control an object it is required to know its structure, the purpose of its existence and its control criteria [5]. The task becomes more complicated when there is a need to control objects in real time, in situations unexpected in advance, using data, information and knowledge, presented in a variety of natural languages and subject areas. In these circumstances, arises a problem of decision making in fuzzy environment [3] based on the data, information and knowledge.

The solution to this problem in this article is focused on the approach of *modeling*, *planning* and *controlling* of linguistic and subject area data, information, knowledge, fuzzy inference and others, by mapping the objectives and constraints in fuzzy environment.

The *novelty* of the technology consists in using of:

- fuzzy modeling and situational fuzzy control of fuzzy data, information and knowledge for implementing an automatic fuzzy inference and finding on that basis a correct, accurate, timely and adequate solution, taking into account the current situation and impact of fuzzy environment;
- the conclusion for planning of control actions on the controlled object and realizing in such way of fuzzy control of the object in the fuzzy environment;
- the processors for *creating* and *synthesis* of images, concepts and meaning, extracted from texts in various natural languages and subject areas, and *serialization* them in bases of data, information and knowledge;

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*, Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_13,

– the bases for organizing of *multi-lingual* human communication through fuzzy dialogue, generalization and explanation of knowledge in the Intelligent fuzzy control system.

There are a lot of interested ideas, methods and algorithms for decision-making in a fuzzy environment, as example in:

– [12] are described two models and methods of Decision-making in one particular subject area of Power Engineering;
– [13] are considered adjustable autonomous agents, that possess partial knowledge about the environment. In a complex environment and unpredictable situations these agents are asked the help of human on base of the model, called HHP-MDP (Human Help Provider MDP) and requests, which are set in advance;
– [14] are reviewed the basic concepts, related to decision-making in fuzzy environments, ontological control for system of systems (SoS) engineering applications, and use ModelSim to simulate such process.

The comparative analysis of these and other works, associated with our work, showed, that there is no integrated linguistic approach to the problem of *situational fuzzy control* in a *fuzzy environment*, including the techniques of *situational control of fuzzy* data, information and knowledge, modeling, planning, decision-making and *situational fuzzy control* of the *object*, based on the achievements of Fuzzy Logic, Situational Control, Artificial Intelligence, Linguistics and others.

2 Main Results

Modeling decisions is defined as construction of a new conceptual situation and a state of controlled objects (fuzzy data, information, knowledge, inference and others), which meets the criteria in the internal and external levels of fuzzy environment. *Planning decisions* is defined as a use of modeling results to create a sequence of alternative decisions that are suitable to the situation and a state of these environments. *Decision-making* is defined as a process of modeling fuzzy logic inference [7] for selecting the relevant decision from a limited number of alternative fuzzy decisions. *Control* is the process of using the modeling results of planning and decision-making in fuzzy environment, in order to implement a control action on the objects (data, information, knowledge, solutions and others) to shift them and their control system to a new state that matches a specified criterion.

To realize targets of Manager (and/or system) to control of organizational object K^R on base of chosen relevant and pertinent decision A^N from a large number of alternatives A, the control task, which is defined by formula [4]

$$\langle A, E, S, T \rangle, \tag{1}$$

can be represented by model (2), [5], [6], (Figure 1, Figure 2),

$$T = \langle A^N, K^R, (S_i : Q_j \overset{u,x,z}{\Longrightarrow} Q_i; I) \rangle, \tag{2}$$

where:

E: Environment of task of decision-making and control (a set of alternatives);
S: System of preferences of the decision maker (DM) (objectives and criteria);
T: The actions of the DM over A of the selection and ordering of alternatives A^N (identification of objective and criterion).

This action of DM is considered as modeling and control data process , which realizes the modeling and generalization of knowledge [8], [9] fuzzy logic inference [7], control dialog [10] and other means and methods of situational control of fuzzy data using natural language (NL) and other languages.

In this case, the approach of *knowledge modeling* is the process of forming, processing and actualization of a semantic network of frames – concepts, which are units of knowledge and data organized in a databases (DB) and knowledge bases (KB), and formally are presented by the module $A^N(G^D, G^Z, G^Y)$ and the model $K^R(R^D, R^Z, R^Y, A^N)$.

The components in these rations are indexed by i, j, k, and are, respectively, the ordinal numbers of: modulated *generalized linguistic variables* $(i = 1, \ldots, l)$, serialized in DBs and KBs on the *extensional* level H of modeling knowledge; *terms* of *subject, linguistic* and other knowledge about indicators $(j = 1, \ldots, s)$ on the *intentional* level I of modeling knowledge; DBs and KBs $(k = I, \ldots, r)$ on the *reformative* level P of knowledge modeling. D, Z, and Y, respectively, identifiers of modulated data information and knowledge frames.

G^D, G^Z, and G^Y, respectively, identifiers of accumulation segments of *subject, linguistic, behavioral* and others data, information and knowledge.

R^D, R^Z, and R^Y, respectively, identifiers of the *rules* of *modeling subject, linguistic, behavioral* and other frame-based representations of data, information and knowledge about indicators and mentioned *linguistic variables* (situations).

A^N are modulated indicators and mentioned *linguistic variables* (generalized situations).

K^R is a formal system of data, information and knowledge modeling, a model of generalized knowledge module, working on base of *rules* R^D, R^Z, R^Y and *relations* R^H, R^I, R^P in the considered segments G^D, G^Z, G^Y of DBs (KBs).

The set of modules are implemented on H, I, and P levels of modeling data, information and knowledge in subject areas by considered methods.

H is the level of purposeful modeling, organized by the use of *fuzzy relations* R^H on the set of components (terms) GLV (corteges in DBs and KBs).

I is the level of purposeful modeling, organized by use of *fuzzy relations* R^I, allowing to determine the membership of the GLV to the certain concept of a particular subject area.

P is the level of purposeful modeling, organized by use of *fuzzy relations* R^P on the set of *fuzzy relations* R^H, R^I, allowing them to determine compliance of set of components GLV (selection of certain concepts) to a particular subject area. The *modeling of fuzzy logic inference* [7] is realized on basis of fuzzy fragments (parcels) of natural and other languages. For this are developed heuristic algorithms, realized by modules K^R of modeling knowledge using rules R^D, R^Z, R^Y,

facts (situations) and their subsets (segments) $G^D, G^Z, G^Y, R^D, R^Z, R^Y$, extracted from the considered BDs and KBs.

According [6], the ratio

$$T = \langle A^N, K^R, \mu_L^R, (S_i : Q_j \overset{u,x,z}{\Longrightarrow} Q_i; I)\rangle \tag{3}$$

represents a formalized model of *modeling of fuzzy logic inference* in the decision making system, where μ_L^R is the *characteristic (logical) function* of Fuzzy Logic.

The result of the reference is a multi-dimensional generalized *rule - relation* R (computational generalized frame) with the area of values $\mu^R : x \to L_x$. The R is defined on H, I and P levels as chain of parcels μ^R each of which forms the group of coded (computing) values. These values are used by the algorithms of the systems for resolution inductive and deductive *fuzzy inference* in each of the observed levels.

Thus, the multi-dimensional relations (generalized rules) R^D, R^Z, R^Y and the values b^H, b^I, b^P of functions μ^H, μ^I, μ^P, mapped by the DM and his support group, respectively, on the considered levels are the result of *modeling fuzzy logic inference* in the *Situational Data Control* system [6].

This system, interactively interacts with DM or/and *Decision Making System* (DMS) (2), to generate the *fuzzy relevant decision* from the number of the selected alternatives (Figure 2).

The process of *generalization and explanation of knowledge* [9] is considered as a search process for a target logical function (situation) by DMS, which uses for that the mentioned *logic inference* and semantic knowledge network. Consequently, the ration of (3) can be represented as

$$T = \langle A^N, K^R, \mu_L^R(\mu_G^R), (S_i : Q_j \overset{u,x,z}{\Longrightarrow} Q_i; I)\rangle \tag{4}$$

The extents of compliance of the numerical and verbal estimates for concepts x from X are determined by fuzzy sets $A^\alpha = \{x, \mu_A(x) \geq \alpha_A\}$ and relations $R_\alpha = \{x, \mu_A(x) \geq \alpha_R\}$ on the level of knowledge modeling and fuzzy logic inference.

In this case, $\mu^P(x) = \bigvee(\mu^H(x) \bigwedge \mu^I(x)) \bigwedge Poss(a/a_\alpha)$ is a composite rule of *generalization* and knowledge *explanation* in the considered *Situational Data, Information and Knowledge Control* (SDIKC) system, where the rule $\mu^P(x)$ is interpreted as a desired logic function, which identifies the *disjunction* of *conjunctions* of modulated logical functions $\mu^H(x), \mu^I(x)$ and $Poss(a/a_\alpha)$.

Establishing the values of α - level for A_α (in certain, and various distributive lattices L), "nearest" to the top of the curves μ_A at each of the considered levels of the modeling of knowledge, - leads (using μ^P) to unite all of "tops" α-slices and to the formation of the resulting fuzzy set of unique (generalized) and relevant data, information and knowledge.

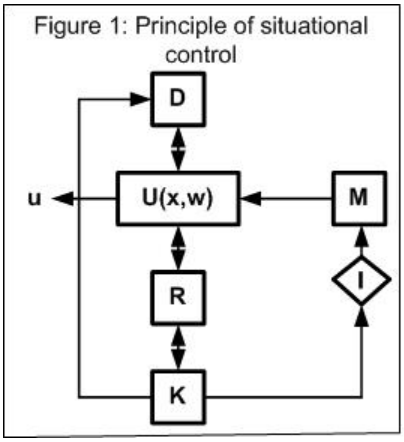
Similarly, the *Dialog control* is based on the ended operations of fuzzy *logic inference, generalization* of knowledge and *control* actions of SDIKC. His model is presented in [10] by (5) as

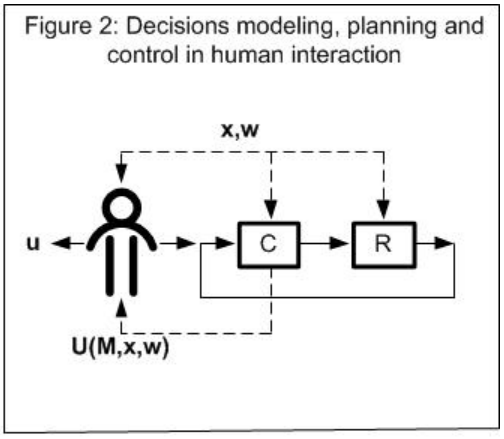
$$T = \langle A^N, K^R, \mu_L^R(\mu_G^R(\mu_D^R)), (S_i : Q_j \overset{u,x,z}{\Longrightarrow} Q_i; I)\rangle, \tag{5}$$

where $\mu_D^R = f(((\mu_L^R(\mu_G^R)), u, x, w)$.

The μ_G^R depends on other fuzzy logic functions μ_D^R, μ_L^R and managed of environmental influences *u,x,w* on the object with the target to make relevant decisions on control, respectively, at the P, I, and H levels of their modeling [6], [7].

$S_i : Q_j \overset{u,x,z}{\Longrightarrow} Q_i; I$ - determines the *elementary act* of control in the process of modeling and selecting the relevant decision, that *transforms* the control system in the new situation S_i, which characterizes its new state Q_l, after the state Q_j was shifted to Q_l (Figure 1, Figure 2).





The systems R and C (Figure 2) are using the *Principle of situational control* and are, respectively, control systems for *modeling* and *planning* decisions.

Together with the Manager they represent the *Decision Making System*. The R (analyst, reviewer) and C (expert, approver) together with Manager (decision maker) define a *Organizational Object Management System* (OOMS).

The Manager interacts with C using Natural Language (NL), sends control actions and receives in response to the situation a few number of relevant alternatives. The result of the interaction between C and R - is obtaining by C (in response to the situation) a few number of improved alternatives.

The C and R are implementing the process of modeling and chosen relevant decisions [7] using SDIKC system.

The *x,w* were fuzzy influences, where the first of them is accessible for estimation, but the second is not.

The formula (5) represents the OOMS in heterogeneous subject fields by using the multi-lingual communication.

The mentioned above K^R is a model of generalized knowledge module K (Figure 1), which is functioning on base of R^D, R^Z, R^Y rules and knowledge segments A^N, represented in DBs and KBs.

I - is a SDIKC system. The main its function in the model M - is interpretation by using K the feed-back reactions from the environment influences x, w and control object $U(x, u)$.

M - is *Intellectual Interface* of data, information, knowledge, decisions and *control dialog*. It contains procedures for recognition of reasons, regularities, fixing the facts and their interpretation.

D and R - components of *Linguistic Processor* (LP). The first is the *Interpreter* and the second is the *Synthesizer*. The LP realizes transformation target's actions, which are expressed in NL and other languages. The method of the LP realization is represented in [11].

Each Manager's action me be interpreted as Data, Information and Knowledge Modeling processes [8], fuzzy logic inference [7], dialogue control [10], generalization knowledge [9] and other methods, which are support the processes of modeling, planning and control decisions in fuzzy environment.

3 Conclusions

The proposed methods and technology are oriented for using in the autonomous (Smart) Information Management Systems, which are operating in a fuzzy environment, interacting with people and other systems in different languages and dissimilar subject areas, where the situations and factors of influence on the control object cannot be determined and structured in advance.

References

1. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–359 (1965)
2. Zadeh, L.A.: The role of fuzzy logic in modeling, identification and control. University of California at Berkeley 15(3), 191–203 (1994)
3. Bellman, R.E., Zadeh, L.A.: Decision-making in a fuzzy environment. 1970: Management Science 14, 141–164 (1970)
4. Borisov, A.N., Alekseev, A.V., Glushkov, V.I., et al.: Processing of fuzzy information in decision-making systems, Moscow. Communication (1989)
5. Pospelov, D.A.: Situational control, Moscow. Science (1986)
6. Khayut, B.Z., Pechersky, Y.N.: Situational Data Control, Moscow. VINITI, Deposited manuscript, 29 pages (1987)
7. Khayut, B.Z.: Modeling of fuzzy logic inference in the decision making system. Institute of Mathematics of the Moldavian Academy of Science, Year book, Collection of papers, Modeling Management systems, Issue of 110, pp. 134–144. Kishinev (1989)
8. Khayut, B.Z., Pechersky, Y.N.: em Knowledge modeling in decision support system, Moscow. Deposited manuscript, 11 pages (1990)
9. Khayut, B.Z., Pechersky, Y.N.: Knowledge generalization in decision support system, Kishinev. Problems of Hardware development, pp. 91–92 (1989)
10. Khayut, B.Z., Pechersky, Y.N.: Approach to the Dialogue control in the situational decision making system, Kishinev. Interactive tools in automated management systems, Collection of papers, pp. 145,146 (1988)

11. Khayut, B.Z., Nikolaev, G.N., Popeskul, A.N., Chigakovsky, V.A.: Realization Linguistic Data Base for Machine translation foreign languages texts using DBMS INES. In: International Machine Translation Symposium, All-Union Translation Center, Collection of papers, Moscow, pp. 236–237 (1983)
12. Ekel, P.Y., Kokshenev, I.V., Parreiras, R.O., Alves, G.B., Souza, P.M.: Fuzzy Set Based Models and Methods of Decision Making and Power Engineering Problems, Delaware, USA. Engineering, vol. 5(5A), 11 pages (2013)
13. Cote, N., Canu, A., Bouzed, M., Mouaddib, A.: Humans-Robots Sliding Collaboration Control in Complex Environments with Adjustable Autonomy. In: IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT), Macau, vol. 2, pp. 146–153 (2012)
14. Liang, H., Kisner, W., Sepehri, N.: Decision-making in fuzzy environments using ontological control with fuzzy automation. In: 9th IEEE International Conference on Cognitive Informatics (ICCI), USA, pp. 542–549 (2010)

Knowledge Integration for Uncertainty Management

Jane Booker[1], Timothy Ross[2], and James Langenbrunner[3]

[1] Booker Scientific
Fredericksburg, TX USA
jane@bookercj.com
[2] Department of Civil Engineering
University of New Mexico
Albuquerque, NM USA
ross@unm.edu
[3] Los Alamos National Laboratory
Los Alamos, NM USA
jrl@lanl.gov

Abstract. Knowledge integration is based upon gathering and aggregating all available data, information, and knowledge from theory, experience, computation and similar applications. Such a "waste nothing" approach becomes important when the underlying theory is difficult to model, when observational data are sparse or difficult to measure, or when uncertainties are large. An inference approach is prescribed, providing common ground for many kinds of uncertainties arising from the sources of data, information and knowledge. These sources are integrated using a modified Saaty's Analytic Hierarchy Process (AHP). A fusion physics application illustrates how to manage the uncertainties in the inference-based integration approach. Zadeh membership functions and possibility distributions contribute to this management.

Keywords: Knowledge integration, Uncertainty management, Analytic hierarchy process (AHP), Possibility distribution, Membership function.

1 Introduction

Scientific and engineering applications have traditionally relied upon repeated test, experimental and observational data for the formulation and confirmation of theory, and for making predictions. Recently complex computational physical models have provided an additional source of knowledge for those activities especially when data are difficult to obtain. However, computations rely on the strength of the theory from which they are constructed and on the field data from which they are validated. Additional knowledge can be gained from considering similar applications, related studies, experience and expertise.

Data, information and knowledge form an interdependent foundation upon which prediction is built, and decisions are made. All these sources and their uncertainties can be combined at a macro (or system) level using Saaty's AHP [1]

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_14,

to supply the combining weights. Implementing AHP requires pairwise comparisons of the sources. Evaluating the utility and relevance of each source relative to others is a manageable task for experts. The knowledge integration approach is a user-friendly method for uncertainty management, applicable to a wide variety of problems. The elegance of this integration approach is best described using an application, one involving inertial confinement fusion (ICF) experiments.

ICF experiments were designed to establish overall reproducibility of a thermonuclear fusion reaction yield from imploding glass capsules. The ICF thermonuclear reaction (1) is for deuterium (D) and tritium (T) ion-reactants, and neutron (n) and alpha (α) particles are output. However, according to theory and historical experimental studies, there are other potential reaction outputs, with rare occurrence. Instead of producing n and α, ${}^5He$ and gamma rays (γ) are produced in approximately every 10^5 reactions. Despite their rare occurrence, the γ's carry more reaction-history information than n's because n-velocity is variable whereas γ-velocity is not variable. This is an uncertainty management issue: quantity versus quality of information.

$$\begin{aligned} D + T &\rightarrow n + \alpha \\ D + T &\rightarrow {}^5He + \gamma. \end{aligned} \tag{1}$$

Of interest is the ratio of occurrence of the two reactions in (1), called the branching ratio (BR). DT BR testing is not necessarily considered data sparse, and it has a long history of studies using diverse experimental protocols. Recently published results [2] conducted two additional relevant test programs to confirm their DT BR results: i) a similar reaction using D with ${}^3He$ instead of T, and ii) an experimental configuration inserting "pucks" of differing materials in the beam in front of the DT capsules. These experiments were designed to minimize some of the high uncertainty issues arising from experimental protocols, measurements, diagnostics, and inferences being made to determine the BR. The large variability in the historical studies is partially due to these issues.

The first two columns of Table 1 show the DT BR recent experimental results of [2]. The first two left columns, labeled as DT BR data, differ according to two different diagnostic measuring protocols: gas Cherenkov detector (GCD) and gamma reaction history (GRH). The next two columns, D^3He and "Puck" data, contain the results from the confirmation studies i) and ii). These four columns indicate large variability although not as large as the two columns of historical data. The far right column has the single theoretical value from experts.

The question is: can the integration of the all the values in Table 1 be beneficial for managing the large uncertainties in the DT BR determination, even though those uncertainties originate from a lack of knowledge rather than from a lack of tests? The answer is yes, as demonstrated in the next section.

2 Knowledge Integration Applied to DT Branching Ratio

The first step in knowledge integration is to identify all available sources of relevant data, information and knowledge. Five different sources are represented

Table 1. Values of DT branching ratio contents of the 5 boxes in Figure 1

DT BR data (GCD)	DT BR data (GRH)	D^3He data	"Puck" data	History data	History data cont.	Theory
3.88e-05	3.65e-05	3.29e-05	5.59e-05	9.81e-06	5.7e-05	5.0e-05
3.64e-05	3.73e-05	3.35e-05	7.26e-05	2.3e-05	5.9e-05	
3.54e-05	3.65e-05	3.26e-05	7.09e-05	9.52e-05	7e-05	
3.90e-05	3.43e-05	5.13e-05	4.08e-05	1.92e-04	5.6e-05	
3.89e-05	3.61e-05	4.36e-05	4.02e-05	2.66e-04	7.6e-05	
4.01e-05	3.31e-05		7.28e-05	2.13e-05	7.8e-05	
4.05e-05	3.15e-05		5.32e-05	2.1e-04	1.06e-04	
3.50e-05	3.11e-05		4.78e-05	7.1e-05	1.2e-04	
3.80e-05	3.28e-05		5.43e-05	5.3e-05	2.11e-04	
3.70e-05	3.06e-05		5.71e-05	5e-05	6.9e-05	
3.49e-05			5.08e-05	4.4e-05	4.5e-05	
3.75e-05						

by boxes for the DT BR application in Figure 1. The recent DT BR experiments [2] are represented by the upper right (UR) box. Similar, analogous, or relevant experiments to the UR box are represented by the upper left (UL) and lower right (LR) boxes. These are the D^3He and "puck" relevant test programs listed in Section 1 as i) and ii). Other knowledge, theory, expertise, history, and first principles are represented by historical studies in the bottom central (BC) box, and a theoretical value is lower left (LL) box.

The integration is not restricted by the number, contents, or type of knowledge in the boxes. For example, the 5-box configuration in Figure 1 differs from the "4-box paradigm" [3] designed for data-sparse applications. Those four boxes were the test data, computations for those tests, a data-rich similar application, and the computations for those. In Figure 1, computer code outputs from physical models are shown as interior boxes for the UR, LR and UL boxes, indicating that the outputs are not at the same macro level (or the same units) as the 5 boxes, but are used instead in the DT BR determination.

To integrate the data, information, and knowledge contents in the 5 boxes requires understanding about making inferences and the uncertainties induced from those inferences. As defined in [3], *Inference is the difference between what is desired to know or obtain (unobservable or unmeasured) and what can be known or obtained (observable or measured).* When evaluating the degree of similarity of the other boxes to the UR box, the expert must have an understanding of inferences being made, that is, an understanding of the degree of similarity attached to the arrow connecting each box to the UR box (i.e., arrows a, b, d and i). Arrows a, b and i represent analogical inference: the process of inferring the DT BR data from the D^3He, "puck" and historical experiments. Arrow d represents theory inference: the process of inferring the DT BR data from theory or first principles. The degree of similarity must also be evaluated for the inferences being made between all the other pairs of boxes. Arrows c, g and j are

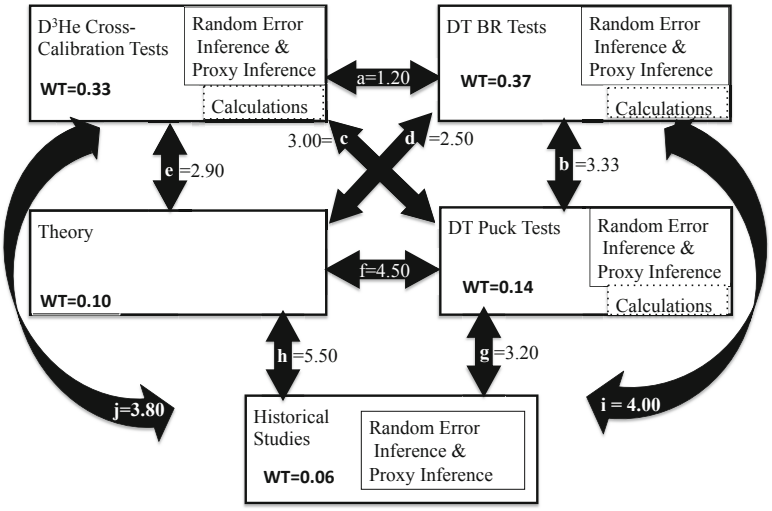


Fig. 1. DT BR 5-box configuration with inference arrows

analogical inference, and arrows e, f, and h are theory inference. Again, understanding inferences represented by these arrows is important.

As shown in Figure 1, inferences can be inside boxes. The most common one is statistical inference: the process of inferring the population of value from a representative sample (the experiments). Because some computations are involved in calculating DT BR from experiments, these induce validation inference: the process of inferring the data from the model outputs. Often quantities that can be observed or measured are not the ones of interest. The process of inferring the desired, but unobservable quantities, from observable quantities is proxy inference. BR is not directly observable, but is inferred from observables.

The next step is to specify factors, issues, models, etc. that affect the degree of similarity between box pairs. For example, differing experimental protocols and diagnostics are important for evaluating arrows out of the History box. The

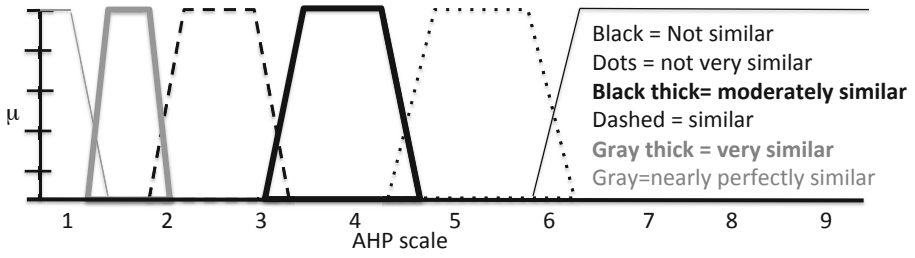


Fig. 2. Fuzzy membership functions for 9-point AHP scale

degree of similarity is then scored using a linguistic-numerical scale based upon Saaty's original 9-point scale [1]. The modification of this scale for similarity can be expressed as Zadeh membership functions in Figure 2. Once the score for each arrow in Figure 1 is determined using formal expert knowledge elicitation [4], the scores are used to construct the upper and lower triangles of a 5x5 triangular matrix, $A_{5,5}$, shown in (2). The lower triangular values in (2) are the inverses of the upper triangular values, and the diagonals are defined as 1.

$$A_{5,5} = \begin{pmatrix} 1 & 1.20 & 3.33 & 2.50 & 4.00 \\ 0.83 & 1 & 3.00 & 2.90 & 3.80 \\ 0.30 & 0.33 & 1 & 4.50 & 3.20 \\ 0.40 & 0.35 & 0.22 & 1 & 5.50 \\ 0.25 & 0.26 & 0.31 & 0.18 & 1 \end{pmatrix} \tag{2}$$

The eigenvector for the principal eigenvalue from the decomposition of $A_{5,5}$ provides the five weights assigned to the boxes. Studies [5] have shown that using a singular value decomposition (SVD) produces weights with less scaling and consistency problems. The SVD decomposition of the 5x5 matrix $A_{5,5}$ is according to (3), where $D_{5,5}$ is a diagonal matrix whose elements are the singular values, d_i, and matrices $U_{5,5}$ and $V_{5,5}$ contain the left and right singular vectors. Matrices $U_{5,5}$ and $V_{5,5}$ are orthonormal having the property: $U^T U = I$ and $V^T V = I$.

$$A_{5,5} = UDV^T = \sum_{i=1}^{5} d_i u_i v_i^T \tag{3}$$

The first singular value for $A_{5,5}$ is $d_1 = 10.83$. The 5 weights, w_i, are calculated from the first columns of $U_{5,5}$ and $V_{5,5}$, u_1 and v_1, as

$$w_j = u_{j,1} + 1/v_{j,1}. \tag{4}$$

After normalization of the weights in (4), they are applied to the box contents to produce the integrated estimate. The weights can be rescaled so that the contents of the UR box have a weight of 1.0. Regardless of scaling, the AHP weights inherently contain information about the inferences being made in combining the contents of the other boxes with the UR box.

A convenient way to summarize the contents of the boxes is to use summary statistics, e.g., the mean or median for central measures. The weights can also be applied to combine uncertainties internal to the boxes. Uncertainties such as the standard deviation or range of values are convenient to use for data. For the case of little or no data in the UR box, the inferences being made in using the other boxes is of primary importance.

For the application here where the UR box contains a reasonable number of experimental results, the confirmatory studies, historical studies and theoretical value are directly incorporated into the DT BR estimation and its uncertainty estimation according to their degree of similarity to the UR box. Using a different weight determination, e.g. determine the weights to minimize the uncertainty, loses the information about the inferences being made.

3 Box Integration and Uncertainty Management

Table 2 contains the results of the box knowledge integration for the DT BR application. Only the statistical inference uncertainty is considered and quantified as standard deviations. Analyzing other inference uncertainties is beyond the scope of this paper. While both eigenvector and SVD weights are shown in Table 2, the integrated box means and standard deviations are calculated using the SVD weights. The integrated mean (IM) is the weighted average of all the data in Table 1 according to

$$IM = \sum_{i=1}^{61} w_i x_i / \sum_{i=1}^{61} w_i. \tag{5}$$

Similarly, an integrated uncertainty (IU) estimate can be calculated from the uncertainties in the boxes, which are quantified as standard deviations in the last column of Table 2. IU is a weighted combination of the standard errors of the box means,S, using the SVD weights in (6). There are other ways to combine the standard deviations in the boxes, and some are featured in Table 3.

$$IU = \sqrt{\sum_{i=1}^{5} w_i S_i^2 / \sum_{i=1}^{5} w_i} \text{ where } S_i = s_i/\sqrt{n_i}. \tag{6}$$

Table 2. AHP weights and box statistics

Box	Eigen Normalized Weights	SVD Normalized Weights	Mean, m_i	n_i	Standard Deviation, s_i
DT BR data	0.33	0.37	3.60e-05	22	0.28e-05
D3He data	0.30	0.33	3.88e-05	5	0.84e-05
"Puck" data	0.20	0.14	5.60e-05	11	1.17e-05
Theory	0.12	0.10	5.00e-05	1	0
Historical data	0.05	0.06	9.01e-05	22	6.87e-05
Integrated			4.44e-05		0.44e-05

In Table 3, approach 1 is a simple combination (equal weights) of all 61 values in Table 1. The large standard deviation divided by the mean results in an 80% error. However, if one were to consider the standard error of the mean of the 61, 0.61e-05, then the % error is a little more than 10%. The problem with this approach is that the data and theory values in the other boxes are not identical in physics or experimental implementation to those in the UR box. So this approach is not adequate. Approach 2 is a variance reduction attempt using statistical analysis of variance (ANOVA) with box as the grouping variable. Some reduction is gained because box-to-box variability is significant at the 1% level, accounting for about one quarter of the total variability from the first row. Approach 3 is

Table 3. Comparison of 5 box knowledge integration to alternative methods

Analysis Approach	Mean or π=1	Standard Deviation or Error	% Error
1. Equal weights, all boxes' contents	5.96e-05	4.76e-05	80%
2. ANOVA, box grouping	5.95e-05	4.25e-05	71%
3. Weighted ANOVA, box grouping	4.44e-05	1.06e-05	24%
4. Box integration, SVD weights	4.44e-05	0.44e-05	10%
5. Possibility distribution function	4.02e-05 - 4.05e-05	NA	NA

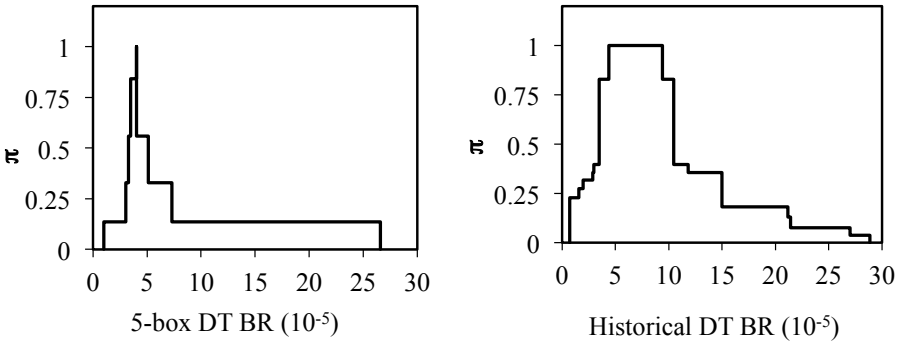


Fig. 3. 5-box π-*df* (left) and historical data π-*df* (right)

another ANOVA applying the SVD weights to each datum. With the weights, about one third of the total variability is explained with the boxes, indicating the importance of the weights. The root mean square errors for approaches 2 and 3 are listed in Table 3. Notice the downward shift in the weighted mean compared to the unweighted mean. Again, the weights represent the degree of applicability of the other boxes to the UR box. Approach 4 is the 5-box integration resulting in a 10% error, the ratio of the integrated uncertainty (6) to the integrated mean (5).

Approach 5 represents an alternative way of combining the contents of the 5 boxes and their uncertainty. The constructed possibility distribution function, π-*df*, represents uncertainty as possibility (π) rather than probability (as when using standard deviations). The π-*df* on the left side of Figure 3 is constructed from (maximum - minimum) ranges of values in the 5 boxes [6]. The theory range was elicited as $\pm$20% of the theoretical value. Only the interval where π=1 in the π-*df* is listed in Table 3. The long right tail is due to the large range of results from the diverse historical experiments. Uncertainty management dictates further investigation of the large uncertainty in the various historical studies. Again, the use of a π-*df* is appropriate as shown on the right side of Figure 3. Here π=1 for values [4.4e-05, 9.4e-05]. The box integration approach could be

applied to the historical data to obtain an integrated mean and uncertainty for the History (BC) box. The number of boxes for the historical studies would be determined by the experimental physicists, but could be as simple as having a box for each of the 22 studies or for the 9 different experimenters.

4 Concluding Remarks

The box integration approach was originally developed as a "waste nothing" analysis tool in data-sparse applications; however it also performs well as an uncertainty management tool as illustrated in Table 3. Uncertainty management is not only uncertainty reduction, which may not be practical; it involves understanding the nature and sources of uncertainty. By defining the relationships between the sources of data, information and knowledge in the boxes as inferences, a commonality is established for handling the uncertainties associated with these inferences. Establishing all uncertainties as inference uncertainties is also part of uncertainty management. The task of quantifying the inference uncertainties is manageable using an appropriate mathematical theory, e.g. probability or possibility theory and relying upon the inferences being made between and within the boxes. That complete task is a work in progress.

The box knowledge integration approach serves as an inference and uncertainty estimation tool [3], and as a knowledge integration method. The number and contents of the boxes is flexible enough to handle a variety of sources of data, information and knowledge. Examples of additional boxes include, different computer models covering subsets of applicable domains, subsets of observational data including isolation of outliers, and subsets of experts' knowledge from differing areas of expertise. The AHP weight determination is convenient to use by experts relying upon their comparisons of degrees of similarity between pairs of boxes to characterize the inferences between then. The use of AHP permits expandability to problems with hierarchical structures, which would involve hierarchical box configurations. The DT BR application box structure can be expanded to include integration of the studies inside the History box (BC).

Other benefits from the box approach include the recovery of information from incomplete tests, and the analysis of multiple quantities of interest. The DT BR application has quantities of interest from observables embedded within the four data boxes. In some cases, these intermediate quantities were available for tests which are not listed in Table 1. Because important information for BR determination was missing, these additional tests were incomplete and omitted. A hierarchical structure permits analysis of the multiple and intermediate quantities, recovering the information content of any incomplete tests.

This document is released to the public as Los Alamos National Laboratory report, LA-UR-13-26617.

References

1. Saaty, T.L.: Analytic Hierarchy Process: Planning, Priority Setting, and Resource Allocation. McGraw-Hill, New York (1980)
2. Kim, Y., et al.: Determination of the Deuterium-Tritium Branching Ratio Based on Inertial Confinement Fusion Implosions. Physical Review C 85(6) (2012)
3. Langenbrunner, J.R., Booker, J.M.: Hemez. F.H., Ross, T.J.: Model Choice Considerations and Information Integration Using Analytical Hierarchy Process. In: Proceedings of the Sixth International Conference on Sensitivity Analysis of Model Output, Milan, Italy, July 19-22 (2010)
4. Meyer, M.A., Booker, J.M.: Eliciting and Analyzing Expert Judgment: A Practical Guide. SIAM Press (2001)
5. Gass, S.I., Rapcsak, T.: Singular Value Decomposition in AHP. European Journal of Operational Research 154, 573–584 (2004)
6. Donald, S.: Development of Empirical Possibility Distributions in Risk Analysis. Ph.D. Dissertation, University of New Mexico, Department of Civil Engineering, Albuquerque, NM (2003)

Co-reference Resolution in Farsi Corpora

Maryam Nazaridoust[1,*], Behrouz Minaie Bidgoli[2], and Siavash Nazaridoust[3]

[1] Department of Information Technology, University of Qom
P.O. Box 3719676333, No.52, 24th Avenue, 30 Metri Keyvanfar, Qom, Iran
Nazaridoust@gmail.com
[2] Department of Computer Engineering,
Iran University of Science and Technology, Tehran, Iran
b_minaei@iust.ac.ir
[3] Department of Industrial Engineering, Islamic Azad University, Lenjan, Iran
nazaridoustn@gmail.com

Abstract. Natural Language Processing (NLP) includes Tasks such as Information Extraction (IE), text summarization, and question and answering, all of which require identifying all the information about an entity exists in the discourse. Therefore a system capable of studying Co-reference Resolution (CR) will contribute to the successful completion of these Tasks. In this paper we are going to study process of Co-reference Resolution and represent a system capable of identifying Co-reference mentions for first the time in Farsi corpora. So we should consider three main steps of Farsi Corpus with Co-reference annotation, system of Mention Recognition and its domain, and the algorithm of predicting Co-reference Mentions as the basis of our study. Therefore, in first step, we prepare a Corpus with suitable labels, and this Corpus as first Farsi corpus having Mention and Co-reference labels can be the basis of many researches related to mention Detection (MD) and CR. Also using such corpus and studying rules and priorities among the mentions, we present a system that identifies the mentions and negative and positive examples. Then by using learning algorithm such as SVM, Neural Network and Decision Tree on extracted samples we have evaluated models for predicting Co-reference mentions in Farsi Language. Finally, we conclude that the performance of neural network is better than other learners.

Keywords: Co-reference Resolution, Mention Detection, SVM, Neural Network and Decision Tree, Farsi Corpus.

1 Introduction

One of the characteristics of the discourse is that you can talk about one or more entities freely in a context and use different kinds of phrases to mention each entities, such as pronoun (he), Nominal (scientist), Name (Lotfali Askarzadeh), or a noun phrase (founder of fuzzy logic), so that the replication of phrases is reduced and Eloquence of the subject is increased.

* Corresponding author.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_15,

This property leads to the creation of potential chains of all noun phrases that refer to a same entity in the text (e.g. he, scientist, Lotfali Askarzadeh, founder of fuzzy logic which all refer to Professor Zadeh). One of the important goals of Extracting Information is the identification of these chains in the text which are performed under a process called Co-reference Resolution [1]. Information Extraction is one of the most important applications of NLP that is under the attention of researchers in two recent decades. The task of Information Extraction (IE) is such that, at first raw texts are received as input and finally processed data are returned as output. The frame obtained from IE can be useful for tasks such as data mining, question and answering, language understanding, summarizing, and information retrieval. Each IE system can be consisted of modules respectively such as Tokenization, POS tagger, Parser, Named Entity Recognition (NER), MD and CR. For instance, consider the following sentence:

[Professor Lotfali Askarzadeh] is [founder of fuzzy logic], [he] enters human logic and nature language in mathematics.

Using the term entity in Co-reference Resolution proposes the question that what are considered as entities? Various categorizing have been presented for different entities; e.g. AEC[1] has suggested a 7-entities Type for Person, Organization, Location, GPE[2], Facilities, Weapon, and Vehicle, with many Sub-Type (e.g. Individual, Group) and a class for each entity[2], and most of the researchers take all these entities or sometimes some of them as their policies.

The term co-reference was used for the first time by Hirschman and Chinkor in MUC[3] conference [3,4]. Many other researchers and they consider two phrases as Co-reference when both of the expressions exactly refer to a same entity in real world. Then, Co-reference Resolution task was the base of many researchers' works. So that some of researchers such as Lee [5], Kobdani [6], and Bunescu [7] have obtained over 80% of accuracy in English language.

Generally, proposed methods for detection of co-referential mentions are divided into 2 total groups of linguistics and machine learning. So we need linguistic knowledge in linguistic methods. Extracting this knowledge from text is a timely process and with error. First linguistic algorithms related to Co-reference Resolution were proposed in late of 1970s, in which semantic linguistics has been used [8, 9]. Then, over the time and development of linguistic corpus, these methods were substituted by statistical methods. In statistical methods, required knowledge is obtained by great corpus and statistical methods, and needs less linguistic knowledge than previous method, and also leads to better results [10].

2 Lotus Corpus

Labelling a corpus by Co-reference information is important linguistically and computationally. From linguistic view, a Corpus which is marked by Co-reference

[1] Automatic Content Extraction.
[2] Geo political.
[3] Message Understanding Conference.

information, provide knowledge about the kind of relationship between 2 Co-reference mention and frequency of different kinds of Co-reference relations. From computational view, such corpus are appropriate for developing and examining automatically trained systems. Samples of applying such corpus to develop and assess presented systems are [10..20]. In addition to mentioned applications, the corpus with Co-reference information can be useful for assessing systems that are only based on linguistic information and do not use statistical information.

The number of corpus marked by co-reference noun phrases and is available for everyone is limited. MUC and ACE are corpus that languages like English, Arabic, and Chinese have been marked extensively in training and test systems of determining noun phrases. On the other side, since there was no such statue for Farsi language, we prepare a proper corpus called Lotus in order to utilize machine learning techniques in Co-reference Resolution. For this purpose, we consider 50 quite long texts from Bijan Khan[4] Corpus as the base and marked given noun phrases.

```
1  DOCUMENT ID=11
2  #   DELM    O
3  روزنامه Name_Single B-NOMH-ORG-MED-GEN-EN107-
4  ابرار   Name_Plurar I-NAM-ORG-MED-SPC-EN107
5  با  P   O
6  عنوان   Name_Single O
7  "   DELM    O
8  قوه Name_Single B-NOMH-ORG-GOV-GEN-EN46-
9  قضاييه  ADJ_SIM I-NOMPOS-ORG-GOV-GEN-EN46
10 و   CON O
11 بلاتكليفي    Name_Single O
12 مطبوعات Name_Plurar B-NOMH-ORG-MED-GEN-EN24
13 توقيف شده   ADJ_SIM O
14 "   DELM    O
15 نوشته   ADJ_INO O
16 است V_PRE   O
17 :   DELM    SID
18 اظهارات Name_Plurar O
19 سخنگوي  Name_Single B-NOMH-PER-TLT-GEN-EN108-
20 قوه Name_Single I-NOMPOS-PER-TLT-GEN-ENA108-
21 قضاييه  ADJ_SIM I-NOMPOS-PER-TLT-GEN-ENA108
22 در  P   O
```

Fig. 1. Example of lotus corpus

[4] http://ece.ut.ac.ir/dbrg/bijankhan/

In Lotus corpus, which is a partly expansion of Bijan Khan, it is sufficient to specify main references related to Persons, Locations, Organizations, and GPE entities. In order to specify the mention type, entity type, entity sub-type, entity class, and referential code, we follow a series of certain laws. These laws are presented according to information of [4,20] Which customized regarding to the Farsi language properties. A section of one of these corpus' texts has been shown in fig.1.

3 Determination of Appropriate Feature Vectors

Generally, before application of machine learning techniques in the process of Co-reference Resolution, wide range of linguistic properties is considered in this process. Extraction of some of properties like semantic properties and knowledge-domain properties is a timely with error process.

After development the application of machine learning techniques in Co-reference Resolution, the properties need much linguistics were replaced by simple linguistic properties and statistical properties. Reported results about rich languages like English are obtained well through statistical methods. But since existing linguistic statistical tools in Farsi are very limited, only the properties have been used in this research that can be calculated simply by existing tools and are appropriate for training algorithms. Some of the properties used in this research are in accordance with presented properties in [10]. The list of selected properties can be observed with their description for each pair of mentions in table1 that shows our feature vectors.

4 Extracting Positive and Negative Examples

After determining feature vectors, now we should determine needed positive and negative examples for machine learning algorithm. Positive examples are created by pairing Co-referenced mentions and negative examples refer to the pairs that are not co-referenced. The number of extracted negative examples was more than positive one and it leads to an imbalance in training data. So, the number of negative examples reduced by applying some limitations; for instance, when both mentions are pronoun and pronominal, they wouldn't be pair. Or in long texts, we consider the limitation of pairing up to 100 words domain. Finally, about 19% of extracted examples are positive and about 81% of them are negative.

5 Determining an Appropriate Learning Algorithm

Performed theoretical studies on machine learning show that none of deductive algorithms works usually better than another. In order to choose an appropriate learner for learning a language (you can refer to[22]), the more learner is proper for the properties of that certain area, inferred model by the learner will be expand well to the new data of the area. In languages like English, Arabic, and Chinese the ground for such comparisons have been provided by creating comprehensive research corpuses like MUC, ACE, OntoNote etc. on which different methods have been examined. But such comparison has not been found for Farsi. Therefore, to choose an appropriate

learner to learn a Farsi problem, different learning methods should be tested practically for a unique statue. For this purpose, all samples obtained from previous stage are divided into 2 training group (80%) and experimental group (20%). Then, 3 basic learning algorithms were trained in two environments of Clementine and Mallet by training group. And finally, models obtained by experimental group are assessed.

Table 1. Feature vector of the mention pair

	feature	values	description
1	Num-I-RepeatD	1,2,3,…	The number of repeated of 1st mention in the document?
2	Num-I-RepeatS	1,2,3,…	The number of repeated of 1st mention in the last sentence?
3	Num-J-RepeatD	1,2,3,…	The number of repeated of 2nd mention in the document?
4	Num-J-RepeatS	1,2,3,…	The number of repeated of 2nd mention in the last sentence?
5	I-Length	1,2,3,…	The number of word forming 1st mention?
6	J-Length	1,2,3,…	The number of word forming 2nd mention?
7	DIST-S	1,2,3,…	Distance between two mentions (sentences)?
8	DIST-W	1,2,3,…	Distance between two mentions (words)?
9	I-Pronoun	T/F	Is first mention pronoun?
10	J-Pronoun	T/F	Is second mention pronoun?
11	STR-Match	T/F	Is the head of first mention match with the second?
12	Number	T/F	Is the number of first mention match with the second?
13	Proper-Name	T/F	Is the first mention Name?
14	Proper-Name	T/F	Is the second mention Name?
15	APP-Match	T/F	Is the second mention the alias of the first one?
16	Entity-Match	T/F	Is the entity type of first mention match with the second?
17	Sub-Entity-Match	T/F	Is the entity subtype of first mention match with the second?

6 Evaluation

6.1 Evaluation Criteria

To assess learning machine, the comparison of 2 learners is needed to investigate their performance. In this regard, there are different assessment tools including MUC, B^3, CEAF, Balance etc. Each of these measures can reflect different behaviour or results for a system [23].

Since the aim is that learning machine identifies Co-referenced mentions correctly, the results of the experiments are examined in the form of criteria of precision (1), recall (2) and F1 measure (3) for positive data[24]. These criteria examine the ability of the system to identify positive examples.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|} \tag{1}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|} \tag{2}$$

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

6.2 Evaluation's Results

Each of basic algorithms under different conditions have been examined and compared to select the most appropriate learner. Learning machine of support vector machine (SVM) has been investigated with different cores (RFB, circle, and polynomial with degree 2 to 8). The results of investigating this algorithm indicate that both criteria of precision and recall will increase in the core of polynomial by increasing the degree of polynomial. And consequently F1 criteria will goes up. Generally, SVM learner has core polynomial with degree 8 at the best conditions.

On the other side, methods based on neural network like Perceptron have different layers, and finally the units will be connected to each other by different weights. In this research, 6 methods of building neural network models were examined and compared (include rapid, dynamic, multiple, pruning, full pruning, and RFBN methods). RBFN method has better performance than others.

At last, the comparison of assessment results from 2 methods have been said and decision tree has been shown in table2.

Table 2. Result of models

F1	Recall	Precision	
39.40	**34.66**	**36.39**	**NN**
30.38	31.44	29.38	**SVM**
28.60	22.41	39.55	**DT**

7 Conclusion and Recommendations

Unlike languages with rich sources like English, Farsi needs a few resources in addition to its special uncertainties. For instance, one of important challenges for Co-reference resolution is the absence of statistical analyst and NER in Farsi. So, we introduce a new statue which has Co-reference resolution labels. Then, with regarding to the 17 introduced properties, existing negative and positive examples in Lotus corpus are extracted and stored in our relational database. As it was mentioned above, the performance of learning algorithms implementation is no such that we can prefer one to another. Thus, in order to select an appropriate learner for Co-reference resolution in Farsi, the results from several basic learning algorithms were examined and compared on extracted samples. Finally, we conclude that the performance of neural network is better than other learners.

References

1. Deemter, K.V., Kibble, R.: On coreferring: coreference in MUC and related annotation schemes. Computational Linguistics 26(4), 629–637 (2000)
2. ACE (Automatic Content Extraction) English Annotation Guidelines for Entities. Version 6.06 2008.06.13
3. Hirschman, L., Chinchor, N.: ۱۹۹۸, Muc-۷ coreference task de_nition. Version ۳. In: Proceedings of the Seventh Message Understanding Conference (MUC-۷),
4. Chinchor, N.A.: Overview of MUC-7/MET-2. In: Proceedings of the Seventh Message Understanding Conference, MUC-7 (1998), http://www.itl.nist.gov/iad/894.02/relatedprojects/muc/proceedings/muc7toc.html
5. Lee, H., Peirsman, Y., Chang, A., Chambers, N., Surdeanu, M., Jurafsky, D.: Stanford's multi-pass sieve coreference resolution system at the conll-2011 shared task. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, Shared Task, pp. 28–34 (2011)
6. Kobdani, H., Schutze, H., Schiehlen, M., Kamp, H.: Bootsrapping Coreference Resolution Using Word Association. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, June 19-24, pp. 783–792 (2011)
7. Bunescu, R.: An Adaptive Clustering Model that Integrates Expert. In: Proceedings of the 20th European Conference for Artificial Intelligence (ECAI-2012). Short paper, Montpellier, France (2012)
8. Strube, M., Hahn, U.: Functional centering-grounding referential coherence in information structure. Computational Linguistics 25(3), 309–344 (1999)

9. Sidner, C.: Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse. PhD thesis, Massachusetts Institute of Technology (1979)
10. Soon, W., Ng, H., Lim, D.: A machine learning approach to Coreference resolution of noun phrases. Computational Linguistics 27(4), 521–544 (2001)
11. Aone, C., Bennett, S.W.: Applying Machin Learning to Anaphora Resolution.
12. Aone, C., Bennett, S.W.: Evaluating automated and manual acquisition of anaphora resolution strategies. In: Proceedings of the 33rd Annual Meeting of the Association for Omputational Linguistics, Cambridge, Mass, June 26-30, vol. 30, pp. 122–129 (1995)
13. Cardie, C., Wagstaff, K.: Noun phrase Coreference as clustering. In: Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods (1999)
14. Fisher, F., Soderland, S., Mccarthy, J., Feng, F., Lehnert, W.: Description of the umass system as used for muc-6. In: Proceedings of the Sixth Message Understanding Conference (MUC-6), pp. 127–140 (1995)
15. Kummerfeld, J.K., Bansal, M., Burkett, D., Klein, D.: Mention Detection, Heuristics for the Onto Notes annotations (2010)
16. Strube, M., Rapp, S., Müller, C.: The influence of minimum edit distance on reference resolution. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, pp. 312–319 (2002)
17. McCarthy, J.: A Trainable Approach to Coreference Resolution for Information Extraction. PhD thesis, Department of Computer Science, University of Massachusetts, Amherst MA (1996)
18. Ng, V., Cardie, C.: Identifying anaphoric and Non-Anaphoric Noun Phrase to Improve Coreference Resolution. In: Proceedings of the 19th International Conference on Computational Linguistics, pp. 1–7. Coling (2002)
19. Ng, V., Cardie, C.: Bootstrapping Coreference classifiers with multiple machine learning algorithms. In: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, pp. 113–120 (2003)
20. Ng, V., Cardie, C.: Improving machine learning approaches to Coreference resolution. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002), pp. 104–111 (2002c)
21. van Deemter, K., Kibble, R.: ۲۰۰۰. On coreferring: coreference in MUC and related annotation schemes. *Computational Linguistics*, ۲۶(۴):۶۲۹–۶۳۷
22. Mooney, R.: Comparative experiments on disambiguating word senses, An illustration of the role of bias in machine learning. In: Brill, E., Church, K. (eds.) Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 82–91 (1996)
23. Tetreault, J.: Empirical evaluations of pronoun resolution. PhD thesis, University of Rochester, Cited on page(s) (2005)
24. Stede, M.: Discourse Processing, Synthesis lectueres On Human language Tecnology (2011)

Real-Time Implementation of a Neural Inverse Optimal Control for a Linear Induction Motor

Victor G. Lopez[1], Edgar N. Sanchez[1], and Alma Y. Alanis[2]

[1] CINVESTAV Unidad Guadalajara
Apartado Postal 31-438, Plaza La Luna, C.P. 45091
Guadalajara, Mexico
{vlopez,sanchez}@gdl.cinvestav.mx
[2] CUCEI, Universidad de Guadalajara
Av. Revolucion 1500, Col. Olimpica, C.P. 44430
Guadalajara, Mexico
almayalanis@gmail.com

Abstract. This paper presents the real-time application of a discrete-time inverse optimal control to a three-phase linear induction motor (LIM) in order to achieve trajectory tracking of a position reference. A recurrent high-order neural network (RHONN) is employed on-line to determine the model of the motor. The equipment and software employed are described as well as real-time trajectory tracking results.

1 Introduction

To implement a control law for a given system, its mathematical model is usually needed. In real-time applications, this control law may not behave as desired because the mathematical model is not usually exact; there are always internal and external disturbances, uncertain parameters and unmodelled dynamics. Neural networks can be employed for nonlinear system identification [1].

Recurrent high-order neural networks (RHONN) are a generalization of the first-order Hopfield network [2]. A RHONN model is able to adjust is parameters on-line and allows to incorporate a priori information about the system structure [1]. This fact motivates to employe the RHONN for identification of the plant to be controlled. The Extended Kalman Filter (EKF) form the basis of a second-order neural network training method [3], where the network weights become the states to be estimated.

Inverse optimal control deals with the problem of determining a control law for a given system such that a cost function is minimized and solving the associated Hamilton-Jacobi-Bellman (HJB) equation, needed in regular optimal control design, is not required [4], [5]. For the inverse approach, a stabilizing feedback control law, based on a priori knowledge of a Control Lyapunov Function (CLF), is designed first and then it is established that this control law optimizes a cost functional.

The inverse optimal control algorithm requires that the full state vector is available. This is not always the case, and then an observer must be implemented.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_16,

High order siliding modes observers, as the Super Twisting-based observer, are widely used because of their atractive features as high robustness and finite-time convergence [6].

The linear induction motor (LIM) is a linear electric actuator on which the electrical energy is turned into mechanical translational movement. LIMs present advantages with respect to other types of motors. They develop magnetic forces directly between the mobile element and the stationary element, without the need of physical contact between both elements, which would restrict the system dynamics. Then, LIM can reach higher speed and reduces undesirables vibrations [8]. For these reasons the LIM has been employed widely in industrial applications such as the transportation, steel, textile, nuclear and space industries [9].

Different kinds of neural network-based controllers have been designed for applications in linear induction motors, as in [10], [11], [12], [13] and many others, where multi-layer perceptrons are employed for the control design. In this paper, the main advantages of the proposed scheme are the employment of the neural weights in a controller with an optimal approach and a significant reduction of the quantity of neural weights required for the system identification, by means of the employment of a RHONN.

In the following, Section 2 present mathematical preliminaries for the neural identification and the inverse optimal control. Sections 3 and 4 describe the application of these algorithms to the linear induction motor. Section 5 presents the procedure and results of the real-time implementation and Section 6 exposes the conclusions of the work.

2 Linear Induction Motor Identification

To identify the LIM model, a RHONN identifier [1], [14] is proposed as:

$$\begin{aligned}
x_{1,k+1} &= w_{11}S(v_k) + w_{12}S(\lambda_{\alpha,k}) + w_{13}S(\lambda_{\beta,k}) - w_{14}(S(\lambda_{\alpha,k})\rho_1 \\
&\quad +S(\lambda_{\beta,k})\rho_2)i_{\alpha,k} + w_{15}(S(\lambda_{\alpha,k})\rho_2 - S(\lambda_{\beta,k})\rho_1)i_{\beta,k} \\
x_{2,k+1} &= w_{21}S(\lambda_{\alpha,k})^2 + w_{22}S(\lambda_{\beta,k})^2 + w_{23}w_f S(v_k)^2 + 2w_f(w_{21}S(\lambda_{\alpha,k})\rho_2 \\
&\quad -w_{22}S(\lambda_{\beta,k})\rho_1)i_{\alpha,k} + 2w_f(w_{21}S(\lambda_{\alpha,k})\rho_1 + w_{22}S(\lambda_{\beta,k})\rho_2)i_{\beta,k} \\
x_{3,k+1} &= w_{31}S(v_k) + w_{32}S(\lambda_{\alpha,k}) + w_{33}S(\lambda_{\beta,k}) + w_{34}S(i_{\alpha,k}) + w_{35}u_{\alpha,k} \\
x_{4,k+1} &= w_{41}S(v_k) + w_{42}S(\lambda_{\alpha,k}) + w_{43}S(\lambda_{\beta,k}) + w_{44}S(i_{\beta,k}) + w_{45}u_{\beta,k} \\
x_{5,k+1} &= w_{51}S(q_k) + w_{52}v_k
\end{aligned} \tag{1}$$

where w_{ij} are the online adjustable network weights, except $w_f = 0.001$, $w_{14} = 0.001$, $w_{15} = 0.001$, $w_{35} = 0.02178$, $w_{45} = 0.02178$ and $w_{52} = 0.001$ which are fixed weights. q_k is the motor position, v_k is the velocity, $\lambda_{\alpha,k}$ and $\lambda\beta, k$ are the magnetic fluxes and $i_{\alpha,k}$ and $i_{\beta,k}$ are the motor currents.

The model (1) is represented in the Nonlinear Block Controllable (NBC) form [14], with three different blocks:

$$x_k^1 = x_{5,k}, \quad x_k^2 = \begin{bmatrix} x_{1,k} \\ x_{2,k} \end{bmatrix}, \quad x_k^3 = \begin{bmatrix} x_{3,k} \\ x_{4,k} \end{bmatrix} \tag{2}$$

The system is transformed into the error variables as

$$z_{k+1}^1 = K_1 z_k^1 + W_k'^1 z_k^2$$

$$z_{k+1}^2 = K_2 z_k^2 + W_k'^2(\chi_k) z_k^3 \tag{3}$$

$$z_{k+1}^3 = f^3(\chi_k) - \chi_{\delta,k+1}^3 + W'^{3,k} u(\chi_k)$$

where $z_k^1 = \chi_k^1 - \chi_{\delta,k}^1$, $z_k^2 = \chi_k^2 - \chi_{\delta,k}^2$, $\chi_{\delta,k}^1$ $\chi_{\delta,k}^2$ are the desired dynamics of the first and the second block respectively, K_1 and K_2 are constant diagonal matrices, and

$$\begin{aligned}
W_k'^1 &= w_{52}, \quad W_k'^3 = \begin{bmatrix} w_{35} & 0 \\ 0 & w_{45} \end{bmatrix} \\
W_k'^2 &= \begin{bmatrix} -w_{14}S(\lambda_{\alpha,k})\rho_1 - w_{14}S(\lambda_{\beta,k})\rho_2 & w_{15}S(\lambda_{\alpha,k})\rho_2 - w_{15}S(\lambda_{\beta,k})\rho_1 \\ 2w_f(w_{21}S(\lambda_{\alpha,k})\rho_2 - w_{22}S(\lambda_{\beta,k})\rho_1) & 2w_f(w_{21}S(\lambda_{\alpha,k})\rho_1 + w_{22}S(\lambda_{\beta,k})\rho_2 \end{bmatrix} \\
f^3 &= \begin{bmatrix} w_{31}S(v_k) + w_{32}S(\lambda_{\alpha,k}) + w_{33}S(\lambda_{\beta,k}) + w_{34}S(i_{\alpha,k}) \\ w_{41}S(v_k) + w_{42}S(\lambda_{\alpha,k}) + w_{43}S(\lambda_{\beta,k}) + w_{44}S(i_{\beta,k}) \end{bmatrix}
\end{aligned} \tag{4}$$

The procedure to make this change of variables is fully explained in [14]. Trajectory tracking can now be achieved for the position reference $\chi_{\delta,k}^1$.

3 Inverse Optimal Control for the LIM

In (3) z_k^1 and z_k^2 will have stable dynamics if the third block state z_k^3 tends to zero when $k \to \infty$. Then, a control law has to be synthesized such that it stabilizes z_{k+1}^3 in (3). The inverse optimal control law [4], [14] can now be applied. It takes the following form:

$$u_k = -\frac{1}{2}(R + \frac{1}{2}W'^{3T}PW'^3)^{-1}W'^{3T}P(f^3(\chi_k) - \chi_{\delta,k+1}^3) \tag{5}$$

where $R = I_2$, I_2 is the 2×2 identity matrix, W'^3 and $f^3(\chi_k)$ are defined in (4) and T denotes the transpose matrix.

Fig. 1 represents the proposed identification and control scheme. The RHONN model block is parallel to the unknown real LIM model and calculates the corresponding weights for identification. The neural inverse optimal control uses these weigths to compute the inptut signal for the LIM model which stabilizes the error with respect to the reference signal and minimizes the cost functional. This control law is neural in the sense that it employes the neural weights to syntesize the input vector.

3.1 Magnetic Fluxes Observer

In order to observe the magnetic fluxes of the LIM and use them in the control law synthesis, the following Super Twisting observer is employed

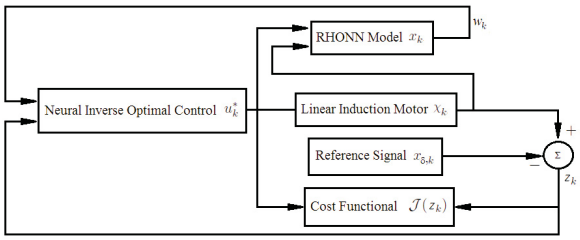


Fig. 1. Identification and control scheme

$$
\begin{aligned}
\hat{\lambda}_{\alpha,k+1} &= (1-k_6T)\hat{\lambda}_{\alpha,k} + k_4Tv_k\rho_1 i_{\alpha,k} - k_4T\rho_1 i_{\alpha,k} + k_5T\rho_2 i_{\alpha,k} + k_4T\rho_2 i_{\beta,k} \\
&\quad -k_4Tv_k\rho_2 i_{\beta,k} + k_5T\rho_1 i_{\beta,k} + a_5Tsign(\tilde{i}_{\alpha,k}) + a_6T\tilde{i}_{\alpha,k} \\
\hat{\lambda}_{\beta,k+1} &= (1-k_6T)\hat{\lambda}_{\beta,k} + k_4Tv_k\rho_2 i_{\alpha,k} - k_4T\rho_2 i_{\alpha,k} - k_5T\rho_1 i_{\alpha,k} - k_4T\rho_1 i_{\beta,k} \\
&\quad +k_4Tv_k\rho_1 i_{\beta,k} + k_5T\rho_2 i_{\beta,k} + a_7Tsign(\tilde{i}_{\beta,k}) + a_8T\tilde{i}_{\beta,k} \\
\hat{i}_{\alpha,k+1} &= (1+k_9T)\hat{i}_{\alpha,k} - k_7T\hat{\lambda}_{\alpha,k}\rho_2 - k_8T\hat{\lambda}_{\alpha,k}v_k\rho_1 + k_7T\hat{\lambda}_{\beta,k}\rho_1 \\
&\quad -k_8T\hat{\lambda}_{\beta,k}v_k\rho_2 - k_{10}Tu_{\alpha,k} + a_1T\left|\tilde{i}_{\alpha,k}\right|^{1/2} sign(\tilde{i}_{\alpha,k}) + a_2T\tilde{i}_{\alpha,k} \\
\hat{i}_{\beta,k+1} &= (1+k_9T)\hat{i}_{\beta,k} + k_8T\hat{\lambda}_{\alpha,k}v_k\rho_2 - k_7T\hat{\lambda}_{\alpha,k}\rho_1 - k_7T\hat{\lambda}_{\beta,k}\rho_2 \\
&\quad -k_8T\hat{\lambda}_{\beta,k}v_k\rho_1 - k_{10}Tu_{\beta,k} + a_3T\left|\tilde{i}_{\beta,k}\right|^{1/2} sign(\tilde{i}_{\beta,k}) + a_4T\tilde{i}_{\beta,k}
\end{aligned}
\tag{6}
$$

where $\hat{\lambda}_{\alpha,k}$, $\hat{\lambda}_{\beta,k}$, $\hat{i}_{\alpha,k}$ e $\hat{i}_{\beta,k}$ are the estimations of $\lambda_{\alpha,k}$, $\lambda_{\beta,k}$, $i_{\alpha,k}$ and $i_{\beta,k}$ respectively, and $\tilde{i}_{\alpha,k} = \hat{i}_{\alpha,k} - i_{\alpha,k}$ and $\tilde{i}_{\beta,k} = \hat{i}_{\beta,k} - i_{\beta,k}$ are the available estimation errors.

4 Real Time Implementation

For the real-time application of the control law described above, the following equipment was employed:

DS1104 Board. DS1104 is a data acquisition and control board (trademark of dSPACE GmbH). It has its own processor and memory where the control algorithm is saved. In figure 2 (a) the top view of the DS1104 is shown.

This board has 6 PWMs drivers, a slave DSP, analog-to-digital converter, digitalo-to-analog converter, incremental encoder and 20 digital input/outputs. It allows to download applications directly from Simulink.

The DS1104 contains a processing unit MPC8240, which is conformed with: microprocessor PowerPC603e, where the control model is implemented, interruption controller, DRAM synchronized controller, 4 general purpose timers, 32 bits, PCI interface (5v, 33Mhz, 32 bit).

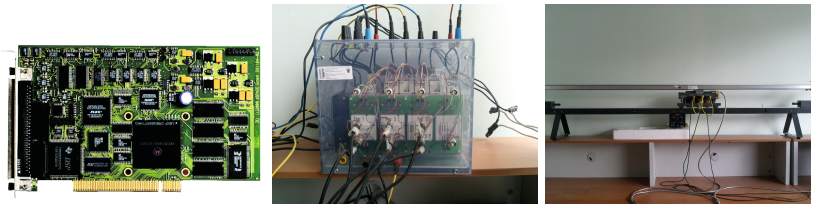
PMW Driver. The PMW driver is employed for the power stage. It is especially designed for motor applications. It has six inputs which are exited by PWMSV for sinusoidal signal reconstruction. These signales must have TTL levels and they correspond to three signals, one for each phase, and their respective inverse. The PMW driver presents short circuit protection, which is activated when the IGBT rasie their temperature. In figure 2 (b) the PMW driver employed is shown.

The main features of the PMW driver are: triphasic signal rectifier, high speed IGBT (50GB123D), polarity-change speed of $5\mu s$, SKHI22 drivers with high speed CMOS excitation and isolation between the logic electronic stage and the power electronic stage.

State Measuring. For the control algorithm, we have available the motor currents and position measuring. In order to obtain the currents measuring, we employed the LEM HX-10P transducers. This sensor presents high measuring precision, small size and space saving, and high immunity to external interference. The sensor output voltage is directly sent to the analog-to-digital converter of the DS1104.

The position of the motor is measured by a linear encoder KA-800M, which send its output signals directly to the board. The motor velocity can be calculated by means of the position change in the sample time.

Linear Induction Motor. The plant where the control law is applied is a linear induction motor LabVolt model 8228. This motor consists in a moveable vehicle and a stationary rail. The moveable vehicle, which is mounted on four bearing rollers, contains what is usually named as the stator of a conventional induction motor. The stationary rail is referred to as the rotor in a conventional induction motor. Two thumb screws on the moveable vehicle previde adjustment of the air gap between the pole faces and the stationary rail surface. Figure 2 (c) shows the moveable vehicle in the rail, with the linear encoder above the motor.



a) b) c)

Fig. 2. DS1104 board (a), PMW driver (b) and linear induction motor (c)

Software. Matlab R2007b and Simulink 7 are the programming software in which the control algorithm is implemented, while the *Real-Time Workshop* transforms these programs into the programming language C and then links it to the DSpace software. Matlab and Simulink are trademarks of the MathWorks Inc.

DSpace R4 is the software which works as the link tool between the control programm in Simulink and the DS1104 board. The ControlDesk tool allows to display the information recieved from the board in order to clarify the visualization of the experiment. DSpace is a trademark of dSPACE GmbH.

Figure 3 shows the connection scheme between the devices employed for the control system. The computer has the necessary software to program the control algorithm and to transmit the information to the data acquisition board. The corresponding signals travel through the PMW driver which transform them into the control inputs for the linear induction motor. The position and the currents measuring are also transmited to the board for the closed loop system.

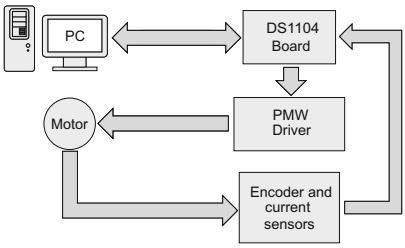


Fig. 3. Control system scheme

4.1 Implementation Results

The neural inverse optimal control algorithm is implemented in real-time employing the board data and the observer outputs as the system states. The voltage vector computed by the control programm are transmitted to the board for its application to the linear induction motor.

In this algorithm, the sample time is $T = 0.0003s$ and the matrices K_i of the equation (3) are defined as $K_1 = 0.9$ and $K_2 = 0.9$. For the Kalman filter algorithm, the matrices are defined as

$$Q = \begin{bmatrix} 0.63759 & 0 \\ 0 & 0.63759 \end{bmatrix}, \quad R = 0.006729 \tag{7}$$

and the Super Twisting observer parameters are defined as $a_1 = a_3 = 50$, $a_2 = a_3 = 500$, $a_5 = a_7 = 1$ y $a_6 = a_8 = 0.1$. Finally, the matrix P for inverse optimal control is

$$P = \begin{bmatrix} 15 & 0.1 \\ 0.1 & 15 \end{bmatrix} \tag{8}$$

The position reference is taken as a sinusoidal signal with an amplitude of 20 cm. The control law applied to the motor is shown in the Figure 4 and the trajectory traching result is presented in the Figure 5.

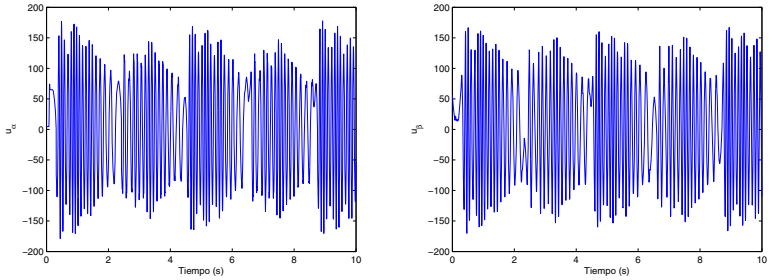


Fig. 4. Input signals for trajectory tracking in real-time application

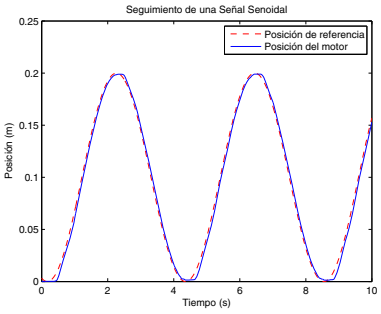


Fig. 5. Real-time motor position with a sinusoidal reference signal

5 Conclusions

A neural inverse optimal controller for nonlinear discrete-time systems is applied in real-time to a linear induction motor. The neural identifier, trained with the extended Kalman filter algorithm, adjusts its parameters in order to reproduce the system dynamics and provides the corresponding neural weights, which are used for the control law. A Super Twisting-based observer was designed to dispose of the flux magnitude of the motor. The proposed scheme achieves trajectory tracking for a sinusoidal position reference with an amplitude of 20 cm.

References

1. Alanis, A.Y., Sanchez, E.N., Loukianov, A.G., Chen, G.: Discrete-time output trajectory tracking by recurrent high-order neural network control. In: Proceedings of the Conference on Decision and Control 2006, San Diego, California (December 2006)
2. Hopfield, J.: Neurons with graded responses have collective computational properties like those of two state neurons. Proc. Nat. Acad. Sci. 81, 3088–3092 (1984)

3. Haykin, S.: Kalman filtering and neural networks. John Wiley and Sons, Inc., New York (2001)
4. Ornelas, F., Loukianov, A.G., Sanchez, E.N.: Discrete-time nonlinear systems inverse optimal control: A control Lyapunov approach. In: IEEE Multiconference on Systems and Control (MSC 2011), Denver, CO, USA, September 28-30 (2011)
5. Freeman, R.A., Kokotovic, P.V.: Robust nonlinear control design. State space and Lyapunov Techniques. Birkhauser, Boston (1996)
6. Salgado, I., Fridman, L., Camacho, O., Chairez, I.: Discrete Time Super-Twisting Observer for 2n dimensional systems. In: 8th International Conference on Electrical Engineering Computing Science and Automatic Control (CCE), Merida, Yucatan, Mexico (2011)
7. Moreno, J.A., Osorio, M.: A Lyapunov approach to second-order sliding mode controllers and observers. In: Proceddings of the 47th IEEE Conference on Decision and Control (CDC), Cancun, Q. Roo, Mexico (2008)
8. Wildi, T.: Electrical machines, drives and power systems, 5th edn. Prentice Hall, Upper Saddle River (2002)
9. Toliyat, H., Kliman, G.B.: Handbook of electric motors, 2nd edn. CRC Press, Boca Raton (2004)
10. Abbasian, M., Soltani, J., Salarvand, A.: Control of high speed Linear Induction Motor using Artificial Neural Networks. In: 2008 Conference on Human System Interactions, Krakow, Poland (May 2002)
11. Hassan, A.A., Mohamed, Y.S., Elbaset, A.A., Hiyama, T., Mohamed, T.H.: A neural network based speed control of a linear induction motor drive. In: 2010 IEEE Region 10 Conference (November 2010)
12. Lin, F.J., Wai, R.J., Chou, W.D., Hsu, S.P.: Adaptive backstepping control using recurrent neural network for linear induction motor drive. IEEE Transactions on Industrial Electronics 49, 134–146 (2002)
13. Lin, F.J., Huang, P.K., Chou, W.D.: Recurrent-Fuzzy-Neural-Network-Controlled Linear Induction Motor Servo Drive Using Genetic Algorithms. IEEE Transactions on Industrial Electronics 54, 1449–1461 (2007)
14. Lopez, V.G., Sanchez, E.N., Alanis, A.Y.: PSO Neural inverse optimal control for a linear induction motor. In: IEEE Congress on Evolutionary Computation, Cancun, Q. Roo, Mexico (2013)

Preliminary Results on a New Fuzzy Cognitive Map Structure

John T. Rickard[1], Janet Aisbett[2], Ronald R. Yager[3], and Greg Gibbon[2]

[1] Distributed Infinity, Inc. Larkspur, CO, USA
terry.rickard@reagan.com
[2] The University of Newcastle, Callaghan, NSW, Australia
{janet.aisbett,greg.gibbon}@newcastle.edu.au
[3] Machine Intelligence Institute, Iona College, New Rochelle, NY, USA
yager@panix.com

Abstract. We introduce a new structure for fuzzy cognitive maps (FCM) where the traditional fan-in structure involving an inner product followed by a squashing function to describe the causal influences of antecedent nodes to a particular consequent node is replaced with a weighted mean type operator. In this paper, we employ the weighted power mean (WPM). Through appropriate selection of the weights and exponents in the WPM operators, we can both account for the relative importance of different antecedent nodes in the dynamics of a particular node, as well as take a perspective ranging continuously from the most pessimistic (minimum) to the most optimistic (maximum) on the normalized aggregation of antecedents for each node. We consider this FCM structure to be more intuitive than the traditional one, as the nonlinearity involved in the WPM is more scrutable with regard to the aggregation of its inputs. We provide examples of this new FCM structure to illustrate its behavior, including convergence.

1 Introduction

Fuzzy cognitive maps (FCM) [1-4] are fuzzy signed di-graphs whose nodes correspond to high-level descriptive concepts and whose links have weights corresponding to the causal relationships (positive or negative) between these concepts. Associated with each node is a fuzzy value indicating the degree to which the corresponding concept is activated as a function of the activations of the other nodes that link into it. FCMs are implemented as dynamical systems that enable the modeling of first-order feedback relationships in complex networks. Typically, certain nodes are initialized and held to fixed activation strengths, and then the network is iterated to determine the evolution of activations of the remaining nodes. The asymptotic behavior of these activations reflects the coupling of causal relationships among the nodes. FCMs have been the subject of a great deal of research interest in recent decades and have proven useful in modeling numerous applications, as surveyed in [4].

The dynamic structure of traditional FCMs is based upon a neural network model, where at each iteration the activation level of a given node is computed as a weighted

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_17,

sum of the activations of its antecedent nodes at the previous iteration, which is then normalized by a sigmoid-type "squashing function" that maps this sum into either the interval $[-1,1]$ or the interval $[0,1]$ Starting from an initial state, repeated iteration of the node activations of FCMs employing this structure are known to follow trajectories resulting in either a fixed point, a periodic limit cycle, an aperiodic attractor or a chaotic attractor [2,5].

Aside from this variable convergence (or in some instances, non-convergence) behavior associated with the traditional FCM architecture, there is a more fundamental issue that detracts from its use in models of real-world conceptual relationships. The simple, biologically-inspired neuronal model employed in this architecture is a serial combination of two functions: 1) a linear weighted arithmetic average to aggregate antecedent node activations, and 2) a nonlinear mapping of this aggregate output back into the interval $[-1,1]$ or $[0,1]$. The serial combination of these two functions results in a somewhat inflexible and inscrutable mathematical transformation, as it imposes the limitation of a linear combination of the input activations, followed by a nonlinearity that is chosen primarily for its normalization properties rather than for its logical significance. The entanglement between these two operations complicates the cognitive interpretation of the overall transformation.

While this neuronal model has proven useful in many neural network applications involving the processing of relatively low-level features such as those derived from time series or pixel values, where a cognitive interpretation of the operation is perhaps of less concern, we question its efficacy in modeling the relationships between node activations involving the higher-level conceptual features typically encountered in FCM models. A more intuitive and scrutable aggregation operator is desirable in these applications.

This has led us to investigate alternative FCM architectures. We are especially interested in the class of mean operators [6,7] for use as the aggregation operator for the antecedent activations in the nodes of the FCM, and in this paper we consider in particular the weighted power mean (WPM) operator [8-11] acting separately on the positively and negatively causal antecedents to a given node, followed by taking the difference between the positive and negative aggregates, which is then simply shifted and scaled to produce a resulting node activation in $[0,1]$.

The WPM operator provides a scrutable aggregation of antecedent activations, incorporating both importance weighting of the antecedents and the ability to take a continuously variable perspective on the input contributions to the aggregation, ranging from the most pessimistic (corresponding to the minimum activation amongst the input antecedents) to the most optimistic (corresponding to the maximum activation). This perspective is determined by the selection of the power exponent p used in the WPM. Various choices for p correspond to well-known aggregation operators, e.g., min ($p = -\infty$), harmonic mean ($p = -1$), geometric mean ($p = 0$), arithmetic mean ($p = 1$), root-mean-square ($p = 2$), and max ($p = +\infty$).

The normalization of the output of the WPM operator is implicit in its structure, resulting in values lying in the unit interval. Thus the use of the WPM as an

aggregation operator for the positively (negatively) causal antecedents to a given node in an FCM enables us more intuitively to specify how the activations of these antecedent nodes exert a corresponding positive (negative) influence on the activation of the subject node.

Another advantage of the WPM is that, using our results in [10], one can feasibly compute type-2 fuzzy WPM aggregations of type-2 fuzzy inputs. This enables the FCM architecture employing the WPM to be generalized to the perceptual computing paradigm of [12], in similar fashion to that described in [13] for social networks. Indeed, our interest in the application of the WPM aggregator to FCM structure was originally motivated by the insights gained from its application to social network analysis. In this case, rather than using scalar values for the WPM weights, exponents and activation values of the nodes in the FCM, we can use interval type-2 (IT2) fuzzy membership functions corresponding to a set of vocabulary words as in [12,13], which enables us to account for imprecise knowledge of these parameters.

In Section II of this paper, we first describe the FCM architecture constructed from mean operators in general terms. We then detail this structure in the case of the WPM operator and illustrate convergence behavior. Section III provides examples of this new FCM architecture, and Section IV concludes. We stress that our research in this area is ongoing, and thus the results in this paper are preliminary.

2 FCMs Constructed from Mean Operators

2.1 General Structure

For a traditional FCM, at time k, the state of the i^{th} node attribute is given by

$$A_i(k) = f\left(\sum_{j=1}^{n} W_{ij} A_j(k-1)\right), \tag{1}$$

where $W_{ii} \neq 0$ admits the case of self-feedback and $f(x)$ is a transfer or 'squashing' function that maps the inner product back into the interval $[-1,1]$, e.g., $f(x) = \tanh(cx) = \left(e^{2cx}-1\right)/\left(e^{2cx}+1\right)$ or into the interval $[0,1]$, e.g., $f(x) = \left(1+e^{-cx}\right)^{-1}$.

In this paper we consider updating node states using the shifted and scaled aggregations of positively and negatively causal antecedent node states obtained through a weighted mean-type aggregation operator $L(\mathbf{x})$, which for all points $\mathbf{x} = (x_1,...,x_n)$ in the state n-cube $[0,1]^n$ satisfies

$$\min(x_1,...,x_n) \leq L(\mathbf{x}) \leq \max(x_1,...,x_n). \tag{2}$$

Such mean aggregation operators include the familiar weighted power means (WPM), the exponential means and the ordered weighted averages. They also include new classes of thresholding mean type aggregation operators introduced in [14,15].

Specifically, we consider dynamic systems for which the state at node i at the k^{th} time interval is given by

$$x_i(k) = L_i\left(\mathbf{x}\left(k-1\right)\right) = \ldots = L_i^k\left(\mathbf{x}\left(0\right)\right) \tag{3}$$

for some mean aggregation operator $L_i : [0,1]^n \to [0,1]$ and for $\mathbf{x}(k)$ the vector of node states $x_i(k)$ at time k.

2.2 FCM Using the Weighted Power Mean

Consider a FCM in which the state of the i^{th} node at time k is given by the following expression using the WPM:

$$x_i(k+1) = 0.5\left[\begin{array}{l}\left(\sum_{j=1}^{n} W_{ij}^{+} x_j(k)^{p_i^{+}}\right)^{\frac{1}{p_i^{+}}} \\ \\ -\left(\sum_{j=1}^{n} W_{ij}^{-} x_j(k)^{p_i^{-}}\right)^{\frac{1}{p_i^{-}}} + \delta_i + \left(1-\delta_i\right) x_i(k)\end{array}\right], \tag{4}$$

where $W_{ij}^{+} \geq 0$, $W_{ij}^{-} \geq 0$, $W_{ij}^{+} W_{ij}^{-} = 0$, and where $\sum_{i=1}^{n} W_{ij}^{+} = 1$ or 0 and $\sum_{i=1}^{n} W_{ij}^{-} = 1$ or 0, and

$$\delta_i = \begin{cases} 0, & W_{ii}^{+} = 1, \sum_{i=1}^{n} W_{ij}^{-} = 0 \\ 1, & \text{otherwise} \end{cases}.$$

The conditions insure that a given antecedent node may be positively or negatively causal (or neither) for a consequent node, but not both, while setting $\delta_i = 0$ covers cases where a node's activation is fixed at its initial value, since then

$$x_i(k+1) = 0.5\left[\left(x_i(k)^{p_i^{+}}\right)^{\frac{1}{p_i^{+}}} + x_i(k)\right] = x_i(k).$$

The two WPM operators in (4) admit separate sets of importance weights on their respective antecedents and also admit separate exponents in the WPMs, which yields separate perspectives on the aggregations of positively and negatively causal antecedents. Thus (4) provides a very general and logically intuitive inferencing structure for specifying the FCM node dynamics.

Note further from (4) that if a node's activation is not fixed and if all of its positively causal antecedent nodes have unity activations and all of its negatively causal antecedent nodes have zero activations, then the two WPM terms within the brackets take the values 1 and 0, respectively, and $x_i(k)$ takes the value 1. On the other hand,

if all positively causal antecedent nodes have zero activation and all negatively causal antecedent nodes have unity activation, then $x_i(k)$ takes the value zero. Finally, if the two WPM terms produce equal values then $x_i(k)$ takes the neutral value of 0.5. Thus our FCM structure exhibits intuitively desired behaviors at both the contra extremes and the equal-valued activations of the positively and negatively causal aggregations.

We observe that the matrices W_{ij}^{+} and W_{ij}^{-} in some instances are right stochastic matrices, i.e., when they have at least one positive entry in each row, since then all of their row sums are equal to unity [16-18]. This type of matrix is also termed a probability matrix, transition matrix or Markov matrix, and is ubiquitous in the analysis of Markov chains. However, this is not always the case, particularly as the FCM node interconnections become more sparsely populated. In the latter instances, there may be one or more rows in either W_{ij}^{+} or W_{ij}^{-} having only zero entries.

Another feature to note from (4) is that, when all rows of *both* W_{ij}^{+} and W_{ij}^{-} have at least one positive entry and none of the node values is held fixed (i.e., there is an off-diagonal positive entry in at least one of the matrices for each row), then the stationary value of this equation is $\forall_i \lim_{k\to\infty} x_i(k) = 0.5$. Under these assumptions, since all row sums of W_{ij}^{+} and W_{ij}^{-} then equal unity, the first two terms in the brackets in (4) cancel each other when $\forall_j x_j(k) = 0.5$. This leaves only $\delta_i = 1$ within the brackets, and thus results in the identity $\forall_i x_i(k) = 0.5$. Since in most applications of FCMs we are interested in their dynamics when one or more node activations are fixed, this case is of little practical interest.

2.3 Convergence Properties

The convergence properties for nonlinear iterations of the form in (4) can be notoriously difficult to prove in the absence of being able to demonstrate that the nonlinearity represents a contraction mapping. There are certain cases where the WPM FCM cycles repeatedly between values, so (4) clearly is not a contraction mapping. A simple example of this is seen by choosing the following matrices for W_{ij}^{+} and W_{ij}^{-}:

$$W^{+} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad W^{-} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{5}$$

in which case each $x_i(k), i = 1,2$ in (4) cycles between two values. One can artificially construct other FCMs in higher dimensions that also exhibit this cyclical behavior.

However, in hundreds of thousands of simulations of WPM FCMs having a more realistic structure, we have observed overall exponential convergence with only a tiny

fraction, i.e. $O\left(10^{-4}\right)$ exceptions. These simulations were conducted by generating uniform random entries lying in $[0,1]$ for the initial vector $\mathbf{x}(0)$ and the matrices W_{ij}^{+} and W_{ij}^{-}, along with uniform random entries lying in $[-10,10]$ for the individual values of p_j^{+} and p_j^{-} in (4). We then randomly zeroed out entries in W_{ij}^{+} and W_{ij}^{-} with varying probabilities, ranging up to 0.5, which produced sparser non-zero entries in these matrices, including instances where one or more entire rows of either matrix had all zero entries. We then iterated (4) for a maximum of 1500 iterations or until the squared norm of the successive differences $\left\|\mathbf{x}(k+1)-\mathbf{x}(k)\right\|_2$ was less than 10^{-12}. Figure 1 is a histogram of the number of iterations taken, drawn from 100,000 WPM FCMs using such randomly generated weight matrices, WPM exponent vectors and initial activations. The highest count to achieve convergence in this particular simulation was 667 iterations, but this obviously was an outlier. In other simulations, we have observed, in the above-noted tiny fraction of cases, periodic cycling between two values for $\mathbf{x}(k+1)$ and $\mathbf{x}(k)$.

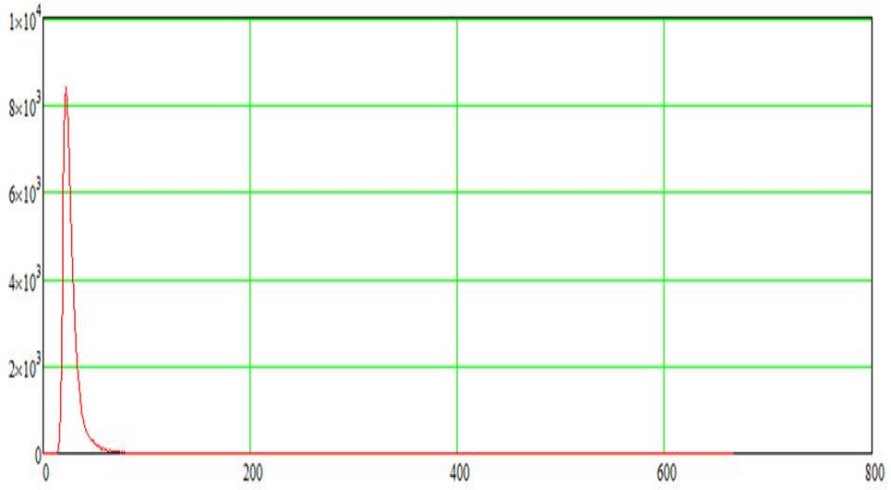


Fig. 1. Histogram of number of iterations required to reach the convergence criterion of a squared norm difference between successive activations of less than 10^{-12}. Horizontal axis is iteration count. Data from 100,000 simulations.

For an 8-node FCM, convergence from arbitrary initial node activations to "interesting" final values that are dependent only upon the system parameters and the values of any fixed node activations generally occurs in $O(10)$ iterations (excepting the previously mentioned case where no node activations are fixed and all rows of W_{ij}^{+} and W_{ij}^{-} have at least one non-zero entry, whereupon all node activations converge to 0.5).

We also tested the extreme cases where all exponents in the WPMs take values of positive or negative infinity, and observed similar results. Thus the WPM FCM system appears to be quite stable for a wide range of realistic parameter values, unlike the traditional FCM structure, which exhibits variable and generally unpredictable limiting behaviors.

We have further work to do on the WPM FCM, both in determining possibly stricter analytical conditions on convergence and in the analysis of the logical implications of the converged values. However, we believe that the initial results obtained recommend themselves to exploitation of this more scrutable structure for modeling the causal relationships between higher-level concepts.

3 Examples

We present some examples in this section that illustrate the behaviors of the WPM FCM for various system parameters and initial states. The examples are chosen to illustrate the effects of successive constraints on the initial node activations, beginning with the unconstrained case. For these examples, we could have selected a particular FCM from the numerous ones that have been studied in the literature (e.g., see 4). However, our purpose in this series of examples is to illustrate the impact of incremental changes in the WPM FCM structure in order to demonstrate the intuitive logical consistency of this structure, which is one of its primary benefits relative to the traditional structure. In future work, we shall perform comparisons between these two structural alternatives on previously studied applications.

3.1 Example 1

Let the transposes of the WPM exponent vectors $\mathbf{p}^{\pm}$ be given by

$$\begin{aligned} \mathbf{p}^{+} &= \begin{bmatrix} 9.651 & 0.271 & 7.961 & -3.471 & 1.416 & -3.037 & -4.197 & 8.456 \end{bmatrix}^{T} \\ \mathbf{p}^{-} &= \begin{bmatrix} 6.14 & 3.734 & 5.337 & 4.435 & -3.169 & -0.314 & 6.764 & 0.812 \end{bmatrix}^{T} \end{aligned} \tag{6}$$

and consider the matrices W_{ij}^{+} and W_{ij}^{-} given by

$$W^{+} = \begin{bmatrix} 0.393 & 0 & 0 & 0 & 0 & 0.455 & 0.152 & 0 \\ 0.219 & 0 & 0.7 & 0 & 0 & 0 & 0.08 & 0 \\ 0.37 & 0 & 0.184 & 0 & 0 & 0 & 0 & 0.446 \\ 0 & 0.055 & 0 & 0 & 0 & 0 & 0 & 0.945 \\ 0 & 0.983 & 0 & 0.017 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \tag{7}$$

$$W^- = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0.891 & 0 & 0.109 & 0 & 0 \\ 0 & 0 & 0 & 0.244 & 0.756 & 0 & 0 & 0 \\ 0 & 0 & 0.342 & 0.131 & 0.172 & 0 & 0.355 & 0 \\ 0 & 0 & 0 & 0 & 0.108 & 0.265 & 0.387 & 0.24 \\ 0.402 & 0.221 & 0 & 0 & 0 & 0.377 & 0 & 0 \\ 0.407 & 0.055 & 0.159 & 0.057 & 0 & 0.323 & 0 & 0 \\ 0.537 & 0.454 & 0 & 0 & 0 & 0 & 0 & 0.009 \end{bmatrix} \tag{8}$$

Note that row 7 of W^+ in (7) has all zeroes, indicating no positively causal inputs to node 7, whereas all other nodes have both positively and negatively causal inputs.

From an initial activation state $\mathbf{x}(0)$, the WPM FCM converges in 56 iterations to the final state shown below:

$$\mathbf{x}(0) = \begin{bmatrix} 0.99 & 0.037 & 0.761 & 0.054 & 0.813 & 0.344 & 0.648 & 0.294 \end{bmatrix}^T \tag{9}$$

$$\mathbf{x}(56) = \begin{bmatrix} 0.619 & 0.529 & 0.476 & 0.426 & 0.626 & 0.368 & 0.223 & 0.324 \end{bmatrix}^T \tag{10}$$

Note from (6) that p_7^- for the negatively causal WPM is relatively large and positive, so this WPM tends toward the maximum of its inputs. Since there is zero contribution from the positively causal WPM for this node, this causes the converged value of node 7 to be relatively small (0.223).

This value also happens to be the sole positively causal input for node 8, whereas its primary negatively causal inputs from nodes 1 and 2 have activations above the neutral value of 0.5, and their corresponding WPM has a positive exponent (0.812). This causes the converged activation for node 8 also to be relatively small. Examining the other converged activations, we conclude that they appear to be consistent with the system parameters.

3.2 Example 2

Suppose that we now replace the first rows of W_{ij}^+ and W_{ij}^- in Example 1 with all zeroes, i.e., so that the activation of node 1 is held fixed, with the remaining rows of these matrices, the WPM exponents and the initial activations unchanged. The FCM now converges in 19 iterations to:

$$\mathbf{x}(19) = \begin{bmatrix} 0.99 & 0.647 & 0.568 & 0.275 & 0.778 & 0.271 & 0.066 & 0.122 \end{bmatrix} \tag{11}$$

The activation of node 1 remains constant as expected, and its high value (virtually the maximum of 1) coupled with its significant weight in $W_{7,1}^-$ and the large positive WPM exponent p_7^- for the negatively causal inputs to node 7 (with no positively causal input) results in a very low converged activation for node 7, exactly as would be anticipated.

3.3 Example 3

Suppose now that both the first and fourth nodes' activations are held fixed in Example 1, so that the corresponding rows in W_{ij}^{+} and W_{ij}^{-} are all zero, with the WPM exponents and initial activations unchanged. Then the FCM activations converge in 25 iterations to:

$$\mathbf{x}(25) = \begin{bmatrix} 0.99 & 0.705 & 0.555 & 0.054 & 0.806 & 0.267 & 0.065 & 0.108 \end{bmatrix} \quad (12)$$

Comparing this vector with the converged activations (11) of the previous example, we see that fixing the activation of node 4 at its low initial value of 0.054 has caused only minor rebalancing of the free nodes' activations. On examining the fourth column of W^{+} in (7), we see that node 4's activation contributes nothing in the way of positively causal influence to any node except node 5, and to this one only to a very small degree since $W_{5,4}^{+} = 0.017$. Thus virtually all of the effect of fixing node 4's activation is accounted for in the negatively causal inputs, which is most prominent for node 2. Again, the results are consistent with what would be expected of the logic.

4 Conclusion

We introduced a new FCM structure in this paper that has a more scrutable interpretation of the aggregations that go into the activations of its nodes, by employing the WPM as the aggregation operator in place of the traditional approach using a linear weighted average followed by a non-linear squashing function. While purely periodic cycling between values of the activations can occur, individual components converged in simulations of realistic structures.

We illustrated this new FCM structure using examples both where the node activations are unconstrained and where one or two of the initial activations are held fixed. We demonstrated that the converged activations obtained from the iterations are consistent with the implied logic of the WPM aggregations of the positively and negatively causal inputs to the nodes, which lends empirical evidence to the utility of this new structure.

In addition to its scrutability, this new structure can be extended to IT2 representations of both the linkage weights and the node activations using the results in [10] and [13], which enables us to compute the successive IT2 membership functions of the node activations as the iterations proceed. Thus we can employ the "perceptual computing" paradigm of [12] to account for imprecise word-based descriptions of the causal relationship strengths, WPM exponents and initial activation levels in the FCM. This represents a major extension to the modeling capability of FCMs.

There obviously remains much work to do on both the analytical and practical aspects of this new FCM structure. Given the preliminary and ongoing nature of our research, we have not yet applied it to specific modeling problems, nor have we yet compared our results to traditional FCM structures. However, we intend to do so in future work.

References

1. Kosko, B.: Fuzzy cognitive maps. Int. J. Man-Mach. Stud. 24(1), 65–75 (1986)
2. Kosko, B.: Fuzzy Engineering. Prentice-Hall, Englewood Cliffs (1997)
3. Glykas, M. (ed.): Fuzzy Cognitive Maps. STUDFUZZ, vol. 247. Springer, Heidelberg (2010)
4. Papageorgiou, E.I., Salmeron, J.L.: A review of fuzzy cognitive maps research during the last decade. IEEE Trans. Fuzzy Syst. 21(1), 66–79 (2013)
5. Boutalis, Y., Kottas, T.L., Christodoulou, M.: Adaptive estimation of fuzzy cognitive maps with proven stability and parameter convergence. IEEE Trans. Fuzzy Syst. 17(4), 874–889 (2009)
6. Yager, R.R.: A general approach to criteria aggregation using fuzzy measures. International J. Man-Machine Studies 38, 187–213 (1993)
7. Yager, R.R.: On mean type aggregation. IEEE Trans. Systems, Man, Cybernetics—Part B: Cybernetics 26, 209–221 (1996)
8. Dujmović, J., Larsen, H.L.: Generalized conjunction/disjunction. J. Approximate Reasoning 46, 423–446 (2007)
9. Dujmović, J.: Continuous preference logic for system evaluation. IEEE Trans. Fuzzy Syst. 15(6), 1082–1099 (2007)
10. Rickard, J.T., Aisbett, J., Yager, R.R., Gibbon, G.: Fuzzy weighted power means in evaluation decisions. In: Proc. 1st World Symposium on Soft Computing, Paper #100, San Francisco, CA (2010)
11. Rickard, J.T., Aisbett, J., Yager, R.R., Gibbon, G.: Linguistic weighted power means: comparison with the linguistic weighted average. In: Proc. FUZZ-IEEE 2011, 2011 World Congress on Computational Intelligence, Taipei, Taiwan, pp. 2185–2192 (2011)
12. Mendel, J.M., Wu, D.: Perceptual Computing. John Wiley & Sons, Hoboken (2010)
13. Rickard, J.T., Yager, R.R.: Perceptual computing in social networks. In: Proc. 2013 International Fuzzy Systems Association World Congress/North American Fuzzy Information Processing Society Annual Mtg. Paper #9, Edmonton, Alberta, Canada (2013)
14. Rickard, J.T., Aisbett, J.: New classes of threshold aggregation functions based upon the Tsallis q-exponential. In: Rickard, J.T., Aisbett, J. (eds.) 2013 International Fuzzy Systems Association World Congress/North American Fuzzy Information Processing Society Annual Mtg, paper #8, Edmonton, AB, Canada (2013)
15. Rickard, J.T., Aisbett, J.: New classes of threshold aggregation functions based upon the Tsallis q-exponential with applications to perceptual computing. IEEE Trans. on Fuzzy Syst. (accepted for publication, 2013)
16. Meyer, C.: Matrix Analysis and Applied Linear Algebra. SIAM, Philadelphia (2001)
17. Gantmacher, F.R.: Theory of Matrices, Vol. 2. AMS Chelsea Publishing, Providence, RI (1989), `http://bookos.org/g/F.%20R.%20Gantmacher`
18. Encyclopedia of Mathematics entry under "Stochastic matrix", `http://www.encyclopediaofmath.org/index.php/Stochastic_matrix`

Time Series Image Data Analysis for Sport Skill

Toshiyuki Maeda[1], Masanori Fujii[1], and Isao Hayashi[2]

[1] Faculty of Management Information, Hannan University, Japan
maechan@hannan-u.ac.jp
[2] Faculty of Informatics, Kansai University, Japan

Abstract. We present a sport skill data analysis with time series image data retrieved from motion pictures, focused on table tennis. We do not use body nor skeleton model, but use only hi-speed motion pictures, from which time series data are obtained and analyzed using data mining methods such as C4.5 and so on. We identify internal models for technical skills as evaluation skillfulness for forehand stroke of table tennis, and discuss mono and meta-functional skills for improving skills.

1 Introduction

As for human action and skill, internal structure of technical skill is layered with mono-functional skill which is generated by human intention, and meta-functional skill which is adjusted with environmental variation [13,8].

Matsumoto et al. discuss that highly skilled workers in companies have internal models of the layered skill structures and they select an action process from internal models in compliance with situations [8].

It is even difficult, however, for skilled workers to understand internal models completely by himself. They usually observe objectively their own represented actions, and achieve highly technical skills with internal models. High level skill is emerged with the refinement of internal models, where some processes are smoothly collaborated such as a bottom-up process from mono-functional skill into meta-functional skill, and as a top-down process of arrangement from representing actions into mono and meta-functional skills [2]. On the contrary, in the field of sport skill analysis, many researches are based on the body structure model and/or skeleton structure model introduced from activity measurement or biomechatronical measurement [10,3,9].

In our research, we assume that forehand strokes [3,9] of table tennis play exemplify sport action, and then identify internal models using data mining methods without body structure model nor skeleton structure model. We focus on technical skill of table tennis [6], and analyze forehand strokes from motion pictures. We evaluate those into three play levels as high/middle/low, and identify internal models using data mining methods [7].

2 Related Works

In [4], on the basis of laboratory research on self-regulation, it was hypothesized that positive self-monitoring, more than negative self-monitoring or comparison

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_18,

and control procedures, would improve the bowling averages of unskilled league bowlers (60 subjects). Conversely, negative self-monitoring was expected to produce the best outcome for relatively skillful league bowlers (67 subjects). In partial support of these hypotheses, positive self-monitors significantly improved their bowling averages from the 90-game baseline to the 9- to 15-game post-intervention assessment (Ximprovement = 11 pins) more than all other groups of low-skilled bowlers; higher skilled bowlers' groups did not change differently. In conjunction with other findings in cognitive behavior therapy and sports psychology, the implications of these results for delineating the circumstances under which positive self-monitoring facilitates self-regulation is discussed.

In [1], comparison of initial and terminal temporal accuracy of 5 male top table tennis players performing attacking forehand drives led to the conclusion that because of a higher temporal accuracy at the moment of ball/bat contact than that at initiation the players did not fully rely on a consistent movement production strategy. Functional trial-to-trial variation was evidenced by negative correlations between the perceptually specified time-to-contact at the moment of initiation and the mean acceleration during the drive; within-trial adaptation was also evident in two of the subjects. It is argued that task constraints provide the organizing principles of perception and action at the same time, thereby establishing a mutual dependency between the two. Allowing for changes in these parameters over time, a unified explanation is suggested that does not take recourse to large amounts of (tacit) knowledge.

In [15], in the present studies, the Leuven Tennis Performance Test (LTPT), a newly developed test procedure to measure stroke performance in match-like conditions in elite tennis players, was evaluated as to its value for research purposes. The LTPT is enacted on a regular tennis court. It consists of first and second services, and of returning balls projected by a machine to target zones indicated by a lighted sign. Neutral, defensive, and offensive tactical situations are elicited by appropriately programming the machine. Stroke quality is determined from simultaneous measurements of error rate, ball velocity, and precision of ball placement. A velocity/precision (VP) and a velocity/precision/error (VPE) indices are also calculated. The validity and sensitivity of the LTPT were determined by verifying whether LTPT scores reflect minor differences in tennis ranking on the one hand and the effects of fatigue on the other hand. Compared with lower ranked players, higher ones made fewer errors ($P < 0.05$). In addition, stroke velocity was higher ($P < 0.05$), and lateral stroke precision, VP, and VPE scores were better ($P < 0.05$) in the latter. Furthermore, fatigue induced by a prolonged tennis load increased ($P < 0.05$) error rate and decreased ($P < 0.05$) stroke velocity and the VP and VPE indices. It is concluded that the LTPT is an accurate, reliable, and valid instrument for the evaluation of stroke quality of high-level tennis players.

[16] describes a method for the measurement of sports form. The data obtained can be used for quantitative sports-skill evaluation. Here, they focus on the golf-driver-swing form, which is difficult to measure and also difficult to improve. The measurement method presented was derived by kinematical human-body model

analysis. The system was developed using three-dimensional (3-D) rate gyro sensors set of positions on the body that express the 3-D rotations and translations during the golf swing. The system accurately measures the golf-driver-swing form of golfers. Data obtained by this system can be related quantitatively to skill criteria as expressed in respected golf lesson textbooks. Quantitative data for criteria geared toward a novice golfer and a midlevel player are equally useful.

In [14], the ability to recognize patterns of play is fundamental to performance in team sports. While typically assumed to be domain-specific, pattern recognition skills may transfer from one sport to another if similarities exist in the perceptual features and their relations and/or the strategies used to encode and retrieve relevant information. A transfer paradigm was employed to compare skilled and less skilled soccer, field hockey and volleyball players' pattern recognition skills. Participants viewed structured and unstructured action sequences from each sport, half of which were randomly represented with clips not previously seen. The task was to identify previously viewed action sequences quickly and accurately. Transfer of pattern recognition skill was dependent on the participant's skill, a sport practised, the nature of the task and degree of structure. The skilled soccer and hockey players were quicker than the skilled volleyball players at recognizing structured soccer and hockey action sequences. Performance differences were not observed on the structured volleyball trials between the skilled soccer, field hockey and volleyball players. The skilled field hockey and soccer players were able to transfer perceptual information or strategies between their respective sports. The less skilled participants' results were less clear. Implications for domain-specific expertise, transfer and diversity across domains are discussed.

3 Analysis for Table Tennis Forehand Strokes

In researches of sports motion analysis, [12] records excited active voltage of muscle fiber using on-body needle electromyography, and [11] uses a marking observation method with on-body multiple marking points, where their objects are to clarify body structure and skeleton structure.

In our research, we assume that technical skills consist of internal models of layered structure as;

- Mono-functional skills corresponding to each body part, and
- Meta-functional skills as upper layer.

We thus identify internal models from observed motion picture data and skill evaluation with represented actions, without discussing the body structure or the skeleton structure. Figure 1 shows our system structure.

In this paper, we focus on table tennis among various sports, and analyze table tennis skills of forehand strokes from observed motion picture data and skill evaluation with represented actions.

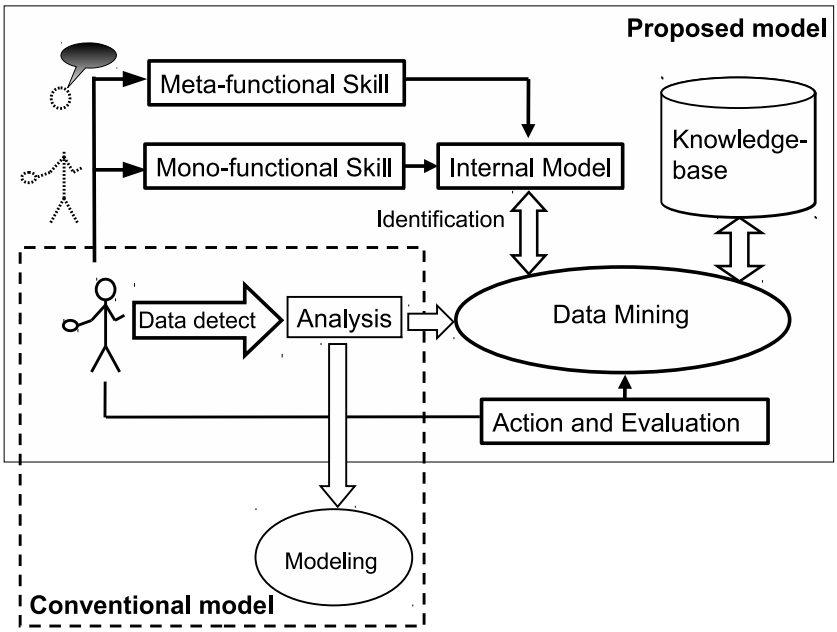


Fig. 1. Conceptional model system

3.1 Experiments

In our experiments, there are 15 subjects who are university male students. At first, we have recorded moving pictures of 15 subjects who are 7 high / 3 middle / 5 low-level university students. As skill evaluation of representing action, We classify as;

- Expert class: members of table tennis club at university,
- Intermediate class: student who used to be members of table tennis club at junior high or high school, and
- Novice class: inexperienced students.

Each player is marked 9 points on the right arm as;

1. Acromioclaviclar joint point,
2. Acromiale point,
3. Radiale,
4. Ulna point,
5. Stylion,
6. Stylion ulnae,
7. Inner side of racket,

8. Outer side of racket, and
9. Top of racket.

Figure 2 shows positions of marking setting.

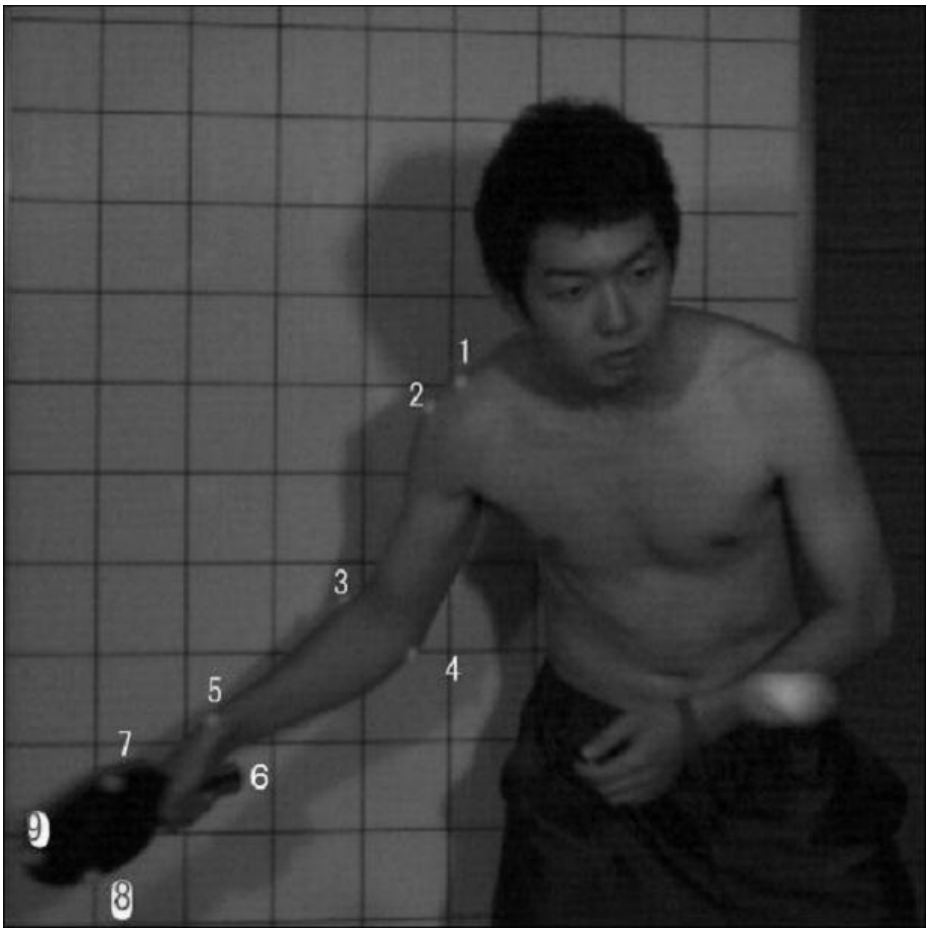


Fig. 2. Measurement markings

The ball delivery machine (TSP52050, YAMATO Table Tennis Inc.) is installed around 30 cm from the end line of the table on the extension of the diagonal line. Balls are delivered on 20 degree elevation angle, 25 of speed levels, and 30 of pitch level at that machine. A subject player returns the delivered ball in a fore-cross way, where the ball is bounded 75 cm inside from the end line. We have recorded the moving traces of forehand strokes using a high-speed camcorder (VCC-H300 by Digimo Inc., resolution: 512×512 pixel and frame-rate: 90 fps) installed 130 cm tall and 360 cm ahead of the player.

While returning the player in 10 minutes, several forehand strokes are recorded for each player (See Figure 3).

Fig. 3. Pictures of subject

4 Skill Class Identification Using Data Mining Techniques

4.1 Three-Class Identification

In our experiments, technical skills of table tennis depend on trajectories rather than axes of observed making points. The skill evaluation of representing action consists of three classes such as Expert, Intermediate, and Novice. Each marking position is represented two dimensional and so the observed data are reconstructed in 90-input / 3-class output. As for expert players, data on two players, which have a high correlation coefficient, are used as learning data, and the rest (one player) for the evaluation.

For applying observed data of forehand strokes of 9 subject players, we reconstruct time series data from the original data. One datum is a set of 90-tuple numbers (9 markings $\times$ 2 axis $(x, y)\times$ 5 frames), and each datum is overlapped with 3 frames data (from third to fifth frame) of the next datum for presenting linkage of each datum (See Fig 4).

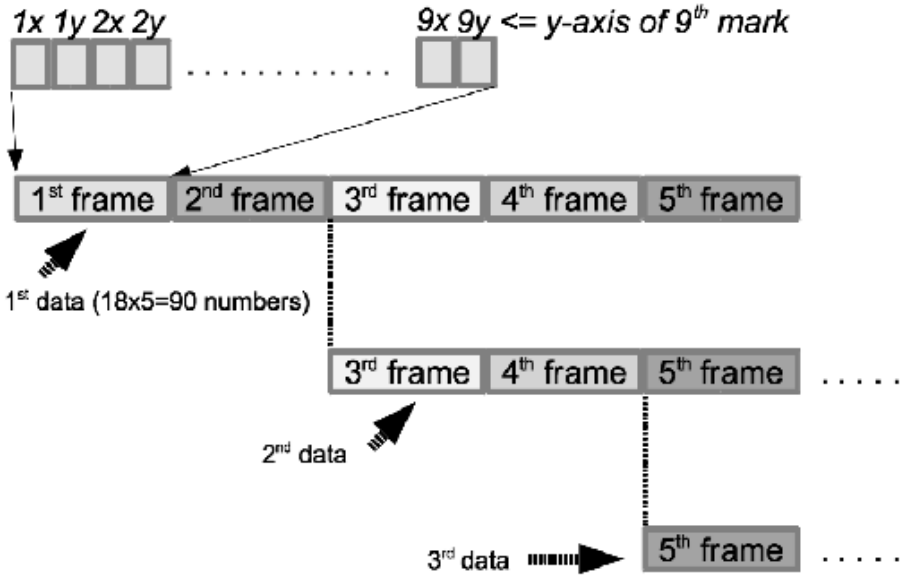


Fig. 4. Data structure from isolated pictures

We use an integrated data-mining environment "weka" [17] and analyze the data by C4.5, Native Bayes Tree (NBT), Random Forest (RF). Table 1 shows the recognition rate of modified data sets. Table 2 also shows the discrimination of classes for each analyzing method for evaluation data.

Table 1. Recognition rate of modified data sets for three classes

	Recognition Rate(%)	
	Learning data	Evaluation data
C4.5	98.1	43.3
NBT	100.0	32.8
RF	100.0	25.4

In those results, recognition rates of NBT and RF for learning data are 100%, which may be over-learned. The rates for evaluation data are not so good, though C4.5 makes good results for both learning and evaluation data. On the contrary, the result of the number of class recognition for each method in Table 2 implies that NBT and RF tend to recognize Expert as Intermediate as well as Novice as Intermediate, and furthermore, fail to evaluate Intermediate for Expert and Novice evaluation data. C4.5 recognizes Expert as Novice, and Novice as Intermediate. All recognition methods tend to select Intermediate in general.

We thus make new data sets which consist of differences of marking data for each frame of the modified data, and apply C4.5 into the new data. Figure

Table 2. Discrimination of classes

		Number of classes for learning data		
	Output class	Expert	Intermediate	Novice
C4.5	Expert	14	0	2
	Intermediate	2	0	23
	Novice	11	0	15
NBT	Expert	1	0	2
	Intermediate	14	0	17
	Novice	12	0	21
RF	Expert	6	0	4
	Intermediate	13	0	25
	Novice	8	0	11

Table 3. Recognition rate of differential data sets

	Recognition Rate(%)	
	Learning data	Evaluation data
C4.5(Difference)	97.1	48.9
C4.5(Original)	98.1	43.3

3 shows the result. This difference data can be regarded as acceleration rate approximately. This result shows a little improvement for recognition rate, which may suggest that the acceleration value is more important to recognize than the time series data.

4.2 Two-Class Identification

As mentioned above, one reason for decreasing the classification rate may be the existence of Middle class, as the features are not specific rather than the other two classes. The skill evaluation of representing action consists of two classes (expert / novice). Each marking position is represented two dimensional and so the observed data are reconstructed in 90-input / 2-class output. As for expert players, data on two players, which have a high correlation coefficient, are used as learning data, and the rest (one player) for the evaluation.

In those results, recognition rates of NBT for cross validation and learning data are not so good. The recognition rate for evaluation data on C4.5 is quite good, though NBT makes poor results as for all data.

Table 4. Recognition rate of modified data sets for two classes

	Recognition Rate(%)		
	Cross Validation	Learning data	Evaluation data
C4.5	95.6	98.1	81.2
NBT	58.9	58.9	52.8

Table 5. Discrimination of classes on C4.5

		Number of classes for learning data	
	Output class	Expert	Novice
C4.5	Expert	40	0
	Novice	26	72

We investigate further for C4.5 analysis so that novice player classification is perfect, though some of expert players are classified into novice, which might be because of some subtle differences of swings, though they should be investigated more.

5 Conclusion

This paper addresses analysis and identification for internal models for technical skills as evaluation skillfulness for forehand stroke motion pictures of table tennis, and discuss mono and meta-functional skills for improving skills. We had some experiments and some results imply that expert or intermediate players can make some categorical groups for technical skills, but there seems not to be a category for novice players because of various individual technical skills. Futhremore, for applying observed data of forehand strokes of players, we reconstruct time series data from the original data and analyze the new data by data mining techniques such as C4.5, NBT, RF, where the recognition rate for evaluation data is not so good, though C4.5 makes good results for both of learning and evaluation data.

As future plans, we have to progress further evaluation, and measure more precise data and then analyze if needed.

Acknowledgment. Part of this research was supported by the Ministry of Education, Science, Sports and Culture, Japan; Grant-in-Aid for Scientific Research (C), 24520173 (2012-2014). This research was collaborated with members of the table tennis club at Hannan University, students at Hannan University and Kansai University. The authors greatly appreciate those.

References

1. Bootsma, R.J., Van Wieringen, P.C.: Timing an attacking forehand drive in table tennis. Journal of Experimental Psychology: Human Perception and Performance 16(1), 21–29 (1990)
2. Hayashi, I., Maeda, T., Fujii, M., Wang, S., Tasaka, T.: Acquisition of Embodied Knowledge on Sport Skill Using TAM Network. In: Proceedings of 2009 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2009), Jeju, Korea, pp. 1038–1043 (2009)

3. Kasai, J., Mori, T., Yoshimura, T., Ohta, A.: Studies of Forehand Strokes in Table Tennis by 3 Dimensional Analysis Using the DLT Method. Waseda Journal of Human Sciences 7(1), 119–127 (1994) (in Japanese)
4. Kirschenbaum, D.S., Ordman, A.M., Tomarken, A.J., Holtzbauer, R.: Effects of differential self-monitoring and level of mastery on sports performance: Brain power bowling. Cognitive Therapy and Research 6(3), 335–341 (1982)
5. Lees, A.: Technique analysis in sports: a critical review. Journal of Sports Sciences 20(10), 813–828 (2002)
6. Maeda, T., Hayashi, I., Tasaka, T., Wang, S., Fujii, M.: Analysis of Embodied Knowledge Using Data-Mining Methods from Image Data. In: Proceedings of the 23th Annual Conference of the Japanese Society for Artificial Intelligence, 1K1-OS8-2, Takamatsu, Japan (2009) (in Japanese)
7. Maeda, T., Hayashi, I., Fujii, M.: Time Series Data Analysis for Sport Skill. In: Proceedings of the 2012 12th International Conference on Intelligent Systems Design and Applications, Kochi, India, pp. 392–397 (2012)
8. Matsumoto, Y.: Organization and Skill – Organization Theory of Preservation of Technique, Hakuto Shobo (2003) (in Japanese)
9. Miyaki, M., Ashida, N., Takashima, T., Azuma, T., Tsuruta, K.: Motion Analysis of Forehand Stroke in Table Tennis Play. In: Proceedings on 42nd Conference of Japanese Society of Physical Education, p. 681 (1991) (in Japanese)
10. Mochizuki, Y., Himeno, R., Omura, K.: Artificial Skill and a New Principle in Sports(Special Issue on Digital Human: Measurement and Modeling of Human Functions. System, Control, and Information 46(8), 498–505 (2002) (in Japanese)
11. Moribe, A., Ae, M., Fujii, N., Hougen, K., Yuda, A.: Study of Forehand Attack in Table Tennis Play – Focus on reaction to variation of rallies –. In: Proceedings on 54th Conference of Japanese Society of Physical Education, p. 377 (2003)
12. Oka, H., Ikuta, A., Nishira, A.: Electromyographic Study of Forehand Skill in Table Tennis. Hyogo University of Teacher Education Journal 20, 19–27 (2000)
13. Shiose, T., Sawaragi, T., Kawakami, K., Katai, O.: Technological Scheme for the Preservation of Technique from Ecological Psychological Approach. Ecological Psychology Research 1(1), 11–18 (2004) (in Japanese)
14. Smeetona, N.J., Wardb, P., Williamsa, A.M.: Do pattern recognition skills transfer across sports? A preliminary analysis. Journal of Sports Sciences 22(2), 205–213 (2004)
15. Vergauwen, L., Spaepen, A.J., Lefevre, J., Hespel, P.: Evaluation of stroke performance in tennis. Medicine and Science in Sports and Exercise 30(8), 1281–1288 (1998)
16. Watanabe, K., Hokari, M.: Kinematical analysis and measurement of sports form. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 36(3), 549–557 (2006)
17. `http://www.cs.waikato.ac.nz/ml/weka/` (2012)
18. Williams, A.M., Vickers, J., Rodrigues, S.: The effects of anxiety on visual search, movement kinematics, and performance in table tennis: A test of Eysenck and Calvo's processing efficiency theory. Journal of Sport & Exercise Psychology 24(4), 438–455 (2002)

Towards Incremental A-r-Star

Daniel Opoku[1], Abdollah Homaifar[1], and Edward W. Tunstel[2]

[1] Electrical and Computer Engineering Department,
North Carolina A & T State University
dopoku@aggies.ncat.edu,
Homaifar@ncat.edu
[2] Johns Hopkins University Applied Physics Laboratory Laurel, MD 20723, USA
Edward.Tunstel@jhuapl.edu

Abstract. Graph search based path planning is popular in mobile robot applications and video game programming. Previously, we developed the A-r-Star pathfinder, a suboptimal variant of the A-Star pathfinder with performance that scales linearly with increasing the resolution (size) and hence sparseness of the grid map of a given continuous world. This paper presents the study of the direct acyclic graph (tree structure) formed by the A-r-Star and outlines steps to developing an incremental version of the A-r-Star. The incremental version of A-r-Star is able to replan faster using information from previous searches to speed up subsequent searches.

Keywords: Graph and tree search strategies, Heuristic methods.

1 Introduction

Graph search based path planning is popular in mobile robot applications and video game programming. Search-based planning techniques usually operate on occupancy grids [1]. The configuration space is represented as a tessellation of regularly sized grid cells with the start location of the robot and the goal location within the grid. A search is then performed on the grid to solve the point-to-point problem by finding a chain of free cells (grid cells that are free of obstacles) linking them. Usually these cells form the shortest possible path. The A^* algorithm, one of the pioneer pathfinders (graph search algorithms), is most popular for path planning in mobile robot and video games [2]. Among the challenges facing A^* and its counterparts is the fact that they are offline search algorithms and that limits their applicability to static environments. Incremental search algorithms are more powerful in handling path planning in dynamic environments; since they reuse information from previous searches to speed up subsequent path planning tasks. Such incremental pathfinders include D^* [3], $D^* - Lite$ [4] and Lifelong Planning $A^*(LPA^*)$ [5] algorithms have been developed and presented in literature. Previously we developed A_r^* (pronounced "A-r-Star") pathfinder [6], a suboptimal variant of the A-Star pathfinder with performance that scales linearly with increasing the resolution (size) and hence sparseness of the grid map of a given continuous world. Informal proofs and simulation experiments have

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_19,

demonstrated that, the A_r^* algorithm is faster than the A^* algorithm running in a uniform and sparse grid. Also, the performance of A_r^* can scales linearly with increasing the gridding resolution of a given world. This paper presents the study of the direct acyclic graph (tree structure) formed by the A-r-Star and outlines steps to developing an incremental version of the A-r-Star. The incremental version of A-r-Star is able to replan faster using information from previous searches to speed up subsequent searches. The rest of the paper is organized as follows, Section 2 gives a background of the directed acyclic graph (DAG) formed by search algorithms, Section 3 describes the A_r^* and steps to developing the incremental A_r^*, Section 4 gives some results and discussion and Section 5 gives the conclusion and some possible research extensions.

2 Background

2.1 The Direct Acyclic Graph

The planning algorithms maintain two main lists namely the OPEN list and the CLOSED list [1]. The CLOSED list comprises of all the nodes that have been explored already, and the OPEN list comprises of all the nodes that have at least one of their neighbours explored and are therefore potential candidates for the next exploration. The path planning algorithms essentially take the regular grid and build a DAG; specifically a tree rooted at the node from which the search starts (NB: this node may be different from the start node). A DAG is a graph with directed edges which has no directed cycles. Table 1 lists some of the planning algorithms and their root nodes. Fig. 1 is an example DAG built by A^* searching from node (1,1) to node (6,6) on an obstacle free 6x6 grid world.

Table 1. Root Nodes for the DAG Built by the Various Pathfinders

Path Finder	Root Node
$\boldsymbol{A^*}$	Start Node
$\boldsymbol{D^*/Focussed-D^*}$,	Goal Node
Incremental $\boldsymbol{A^*}$ ($\boldsymbol{LPA^*}$)	Start Node
$\boldsymbol{D^*-Lite}$	Goal Node

It can be shown that, after each complete search, all the nodes on the CLOSED list (explored nodes) are roots of sub-graphs in the DAG and the nodes on the OPEN list are the leaves of DAG. Thus, blocking of a single node on the CLOSED list means that node has been discontinued from the main DAG. Hence, the system has to find a systematic way of reconnecting all the sub-graphs (and leaves) rooted at that blocked node to the graph where possible. In some circumstance, the blocked node will put all the leaf nodes beyond a lower bound of the path cost to the goal (i.e. the new optimal goal distance $g^{**}(s_{goal})$), and that means that all the leaves on the sub-graph cannot become part of the main graph anymore. Under such circumstances, the system will take the goal and place it as part of another sub-graph.

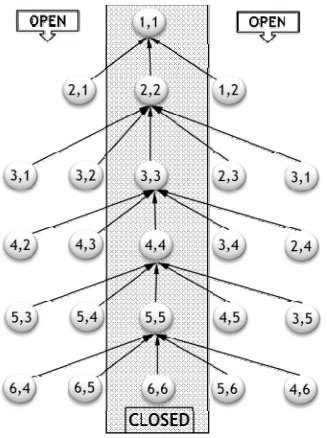


Fig. 1. An example of the DAG built by A* on a **6x6** grid while searching from $(\mathbf{1},\mathbf{1})$ to $(\mathbf{6},\mathbf{6})$

This idea of discontinuous sub-branches and reconnection of sub-branches has been exploited in the developing of incremental search algorithms such as LPA^*, D^*, $Focussed - D^*$ and $D^* - Lite$, and etc. D^*, uses the concept of RAISED (nodes whose edge costs have increased due to the change in edge cost) and LOWERED (nodes whose edge costs have decreased by the change in edge cost) to propagate the cost changes to all the sequences that contain the edges whose costs have changed [1].

LPA^* exploits this knowledge by using consistency evaluation (over-consistent and under-consistent nodes) to propagate the edge cost changes to the affected sequences of back-pointers (sub-graphs) [2]. $D^* - Lite$ [3]operates in a way similar to LPA^* except that it starts searching from the goal and so implements a routine to prevent cycles in the graph when propagating the cost changes. The subsequent sections illustrate how this knowledge has been used to develop an incremental version of A_r^* .

3 Methodology

3.1 The A-r-Star Pathfinder

The A_r^* algorithm is a modified version of the A^* algorithm that interweaves node decimation with path-finding in a uniform grid/mesh. For a given node, A^* counts only immediate nodes (4- or 8- connected nodes) as its neighbors. Thu, even when all the nodes in a particular area are similar, it will still do some computation for all of them. Consequently, if an obstacle blocks the direct line of sight close to the goal, the number of nodes that need to be explored increases by a factor of two or more. This translates into increasing the search time. Fig. 2 illustrates this phenomenon and shows how this contributes to 'kinks' in the path returned by A^*. The circles indicate

the nodes that were explored (placed on the CLOSED list) before the path was found. A_r^* uses the non-uniform mesh building idea that, a cluster of nodes with similar characteristics can be represented with a bigger node with minimal loss of information (see Fig. 3) [4]. The A_r^* algorithm therefore incorporates collapsing of cluster of free nodes up to a given radius, r into a single node with properties that commensurate with the union of those nodes.

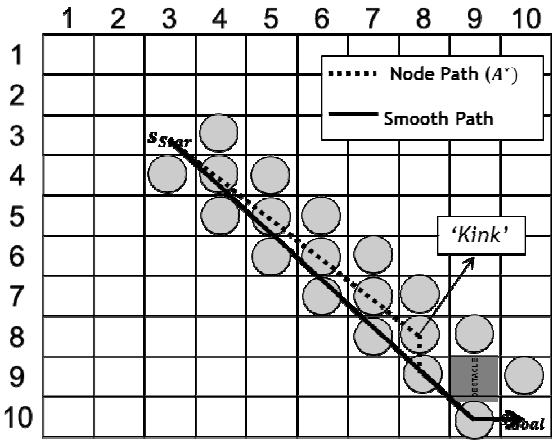


Fig. 2. This figure shows how A-Star responds to an obstacle

For instance, in Fig. 3, the nodes at Level-1 to Level-3 of $s_{Start} = (5,5)$ have been decimated to form one big node with the original Level-4-Neighbors now forming the neighbors of New Node. For multi-level terrain, one will use a distance transform [1, 2] to identify changes in the nodes; but for the binary occupancy grid such as the one used in this paper, the task reduces to identifying the nearest obstacle to the current node.

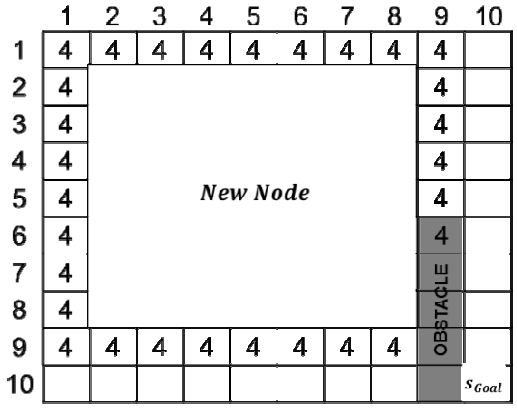


Fig. 3. This figure shows decimated Level-1 through Level-3 nodes to form a single node

The overall effect is a reduction in the number of nodes needed to be explored, computation cost and increased search speed in a sparse uniform gridded world. The 'star' in the name does not suggest that it always finds an optimal path, it is just intended to retain its resemblance to its namesake, A^*. The r stands for radius (range) defined as the maximum allowable radius (in node distance) of a 'ball' of nodes that can be counted as the neighbors of a given node (see Fig. 3) Thus, only Level-R-Neighbors are considered during the search where $1 \leq R \leq r$.

Let the node distance from the closest obstacle node to a given node s be $r_0(s)$, then at the end of the search: $r_0(s) \leq r \; \forall \; s \in S_{Open}$. Implementation-wise, this is achieved by searching for the minimum R, such that, at least one of the Level-R-Neighbors of a node being expanded belongs to the set of blocked nodes. Then all nodes in the neighborhood of s such that $R < r_0(s)$ are tagged as skip nodes ($s'.tag = Skip$) and nodes such that $R = r_0(s)$ are returned as the Level-R-neighbors. The pseudo code for the algorithm is similar to that of A^* with two modifications. The first modification is by replacing the Expand subroutine in the A^* algorithm [1, 2] with a modified version whose the pseudo-code shown in Algorithm 1.

This modification will not allow a node to be placed on the OPEN list once it is tagged as skip (Algorithm 1 line L.10) However, the nodes that get placed on the $OPEN$ list before being tagged as skip will be explored. Thus, after the first modification, tag open dominates/overwrites tag skip. The second modification is to switch the tag dominance so that skip dominates open. This suppresses nodes that make it to the $OPEN$ list from one node before being tagged as skip from another node from being expanded (i.e., they will stay on the $OPEN$ list till the algorithm terminates).

```
L.01: Expand(s, S_Closed, S_f , s_goal, r)
L.02:   R:=1;
L.03:   skipflag:=true
L.04: while R≤r
L.05:     levelrneighbor(s):=LRNG(R, s)
L.06:     foreach  s' ∈ levelrneighbor(s)
L.07:       if s' ∉ S_f
L.08:         skipflag:=false
L.09:         continue
L.10:       Elseif  s'.tag≠skip and  s'∉ S_Closed
L.11:         lrneighbors:=lrneighbors ∪ s'
L.12:     if not(skipflag)
L.13:       return lrneighbors
L.14:     Foreach s' ∈ lrneighbors
L.15:       s'.tag:=skip;
L.16:       r++
L.17:   return lrneighbors
L.18: end
```

Algorithm 1. First modification resulting in the Basic A-r-Star

The resulting algorithm after the last modification is called the A_r^* algorithm. The A_r^* pseudo code can be derived after the first modification by including in the main code a 'tag inspector' to ensure that nodes that are tagged skip are not expanded, the price to be paid is in the form of storage memory requirement. An alternative way, which trades off memory for computational power, is to immediately delete a node from the OPEN list once it is tagged skip.

3.2 Multiple Goal Path Planning

The A_r^* algorithm, like the other pathfinders, builds a DAG during a path search. The only major difference is that the DAG build by A_r^* has many 'silent' nodes. These are the nodes tagged as skip and therefore doesn't appear on the graph. Consequently, very less information is available about these nodes. The DAG developed by A_r^* can be harnessed in subsequent path searches. This is the case if the environment remains unchanged and the given subsequent path search task starts from the same root node. Multiple-destination path planning is applicable to multiple agent based applications wherein a single planner plans paths to send agents from a single point (base station) to multiple destinations. Conversely, it can be applied to scenarios wherein a planner dispatches agents from multiple destinations to a single point, to accomplish a single goal that may be beyond the capability of a single agent.

To derive the validity usability of previous DAG for subsequent searches, consider the general cost function used in most the heuristic graph search algorithms shown in Equation (1).

$$f(s) = g(s) + h(s) \tag{1}$$

Here, $g(s)$ is the start distance-defined as the best discovered distance from the root node to the node s; and $h(s)$ is the user defined heuristic estimate of the distance between the node s and the destination/goal node. Therefore, the cost function $f(s)$ is the estimated cost from the root node to the goal node passing through node s. Assuming that the environment is static, it is expected that after the search, the cost function will become Equation (2) for all the nodes which are on the CLOSED list. The 'star' in $g^*(s)$ indicates an optimal distance.

$$f(s) = g^*(s) + h(s) \tag{2}$$

This implies that, the same structure can be used to plan the path to any point in the environment from the same root node. Since the nodes on the CLOSED node have optimal g(s), they do not need further expansion. To prove this, consider the expansion of Equation (1) for the static environment as shown in (3).

$$f(s) = g(parent(s)) + c(s, parent(s)) + h(s) \tag{3}$$

Since $c(s, parent(s))$ remains constant for a static environment; for every node on the OPEN list, there exists a parent on the CLOSED list and thus $g(parent(s)) = g^*(parent(s))$. Note however that, this parent might not be the optimal parent for that particular node on the OPEN list. The task of planning a suboptimal path from

the root node to another node with parent on the CLOSED list is trivial cause $g(parent(s)) = g^*(parent(s))$. Thus, it reduces to path planning from the parent to the child (NB: there is a direct line of sight between every child and the parent) and the path planning between the root node and the parent which exists already in the previous structure. This approach is similar to the ROADMAP [1] approach to path planning. The task of planning from the root node to another node beyond the reach of the previous search can be done by changing the heuristic for all the nodes in the *OPEN* list to conform to the current goal node. That is, $h(s, s_{Goal})$ becomes $h(s, s_{NewGoal})$; here $s_{NewGoal}$ is the new destination. This will naturally extend the previous graph towards that new goal.

3.3 Incremental A-r-Star

The incremental A_r^* algorithm consists of essentially two main procedures namely *PROCESS-STATE* and *PRUNE-BRANCH*. The *PROCESS-STATE* procedure handles the computation of the DAG either from scratch or by extending an existing DAG from previous search to cover the current goal node as described in subsection 3.2. Therefore, the *PROCESS-STATE* procedure is the same as the A_r^* pathfinder presented earlier except that it starts its search from the goal node. This implies that, the DAG that will be built will be rooted at the goal node. Secondly, we introduce an array that keeps track of the parent-child relationships. The *PRUNE-BRANCH* procedure effects edge cost changes and prunes the sub-graphs and/or leaves from the main DAG by dissolving all the sub-graphs centered at the parent of the blocked node. Note: the dissolution starts from the parent and not the current node because the current node is blocked and need to be removed from the tree along with all its siblings. The pseudo code for the algorithm for the *PRUNE-BRANCH* procedure is shown in Algorithm 2.

```
L01:  PRUNE-BRANCH(s)
L02:    s_p=parent(s)
L03:    PNodes=∅
L04:    PNodes.Insert(s_c)
L05:    while PNodes≠∅
L06:      s_c=PNodes.pop()
L07:      if TAG(s_c)=closed
L08:        Branches=RootBranches(s_c)
L09:        PNodes.Insert(Branches)
L10:        CLOSED.Remove(s_c)
L11:      esleif TAG(s_c) =open
L12:        OPEN.Remove(s_c)
L13:      TAG(s_c)=NEW
L14:      g(s_c)=∞
L15:      OPEN.Insert(s_p)
L16: end
```

Algorithm 2. The pseudo code for *PRUNE-BRANCH*

The algorithm first looks forward for the parent of the node affected, and saves that in s_p. This node is placed on the $PNodes$ array. The system then enters into the looping mode until $PNodes$ becomes empty. At every loop, the system pops a node from $PNodes$ and stores it in s_c which is the current node under consideration.

Next, the system accrues all its children using $RootBranches$ and puts them on PNodes and set $TAG(s_c) = NEW$ to free the node for subsequent re-planning. NB: If s_c belongs to the CLOSED or OPEN list, it must be removed from it. For example, Fig. 4 shows a DAG built by A_r^* pathfinder planning from node (1,1) to the node (3,10) in an obstacle free grid. After planning the path, if the robot start navigating from the starts node (3,10) towards the root node (1,1) and it discovers that node (1,5) is blocked, the first 10 iterations will proceed as shown in Table 2. Iterations 11 to the end only involve the freeing of the leaf nodes. In effect, the other sub-graphs rooted at (1,3) and those which are not rooted at (1,3) are left. The information stored in these sub-graphs can therefore be reused for the replanning.

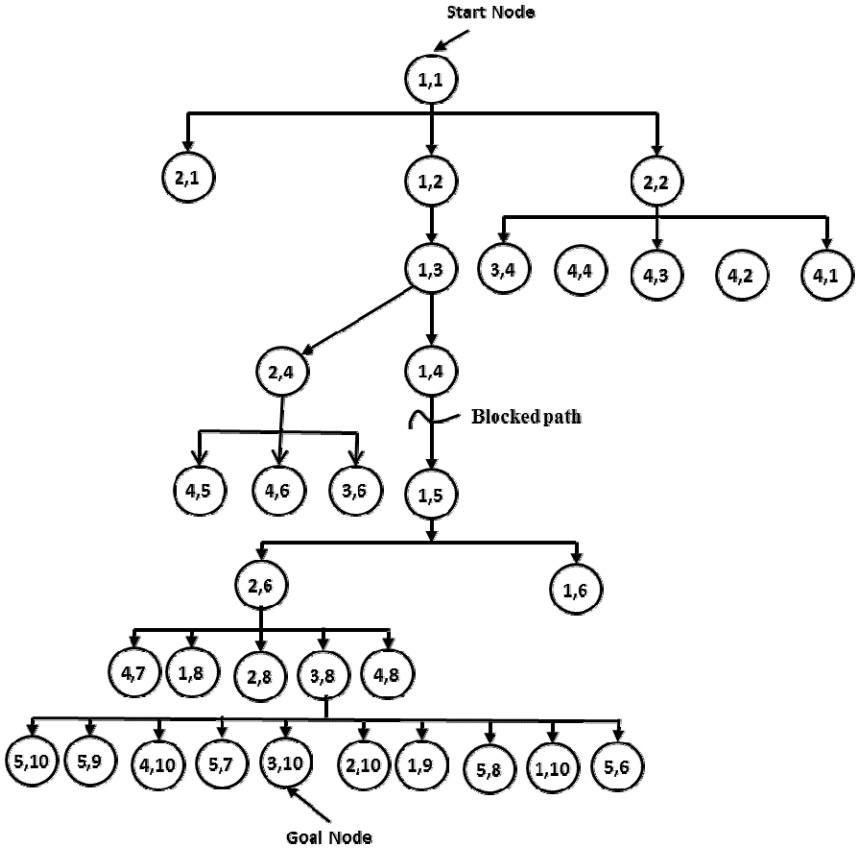


Fig. 4. A Sample DAG built by the A-r-Star for a 10 x10 grid world without obstacle, when searching from node (1,1) to (3,10)

Table 2. First Ten Iterations of PRUNE-BRANCH Acting on the Sample DAG in Fig. 4 When Blocked at the Node (1,5)

Iteration	$PNodes$	S_c	$RootBranches$
1	(1,4)	(1,4)	(1,5)
2	(1,5)	(1,5)	(1,6),(2,6)
3	(1,6), (2,6)	(1,6)	
4	(2,6)	(2,6)	(4,7), (1,8), (2,8), (3,8), (4,8)
5	(4,7), (1,8), (2,8), (3,8), (4,8)	(4,7)	
6	(1,8), (2,8), (3,8), (4,8)	(1,8)	
7	(2,8)	(2,8)	
8	(3,8), (4,8)	(3,8)	(1,9), (1,10), (2,10),(3,10), (4,10), (5,6), (5,7), (5,8), (5,9), (5,10)
9	(1,9), (1,10), (2,10),(3,10), (4,10), (5,6), (5,7), (5,8), (5,9), (5,10)	(1,9)	
10	(1,10), (2,10),(3,10), (4,10), (5,6), (5,7), (5,8), (5,9), (5,10)	(1,10)	

3.4 Challenge for the incremental A-r-Star

One major challenge of using the *PRUNE-BRANCH* algorithm is that, when the blocked node is very close to the root node; it tends to be time consuming. This is because, there will be a lot of branches to dissolve and this will take a longer time to complete. During real time implementation on a robot, this can be overcome by dissolving nodes and their corresponding branches as soon as they are traversed, thus the dissolution time will spread over the run time.

4 Results and Discussion

4.1 Comparison of the A-r-Star Planning from Scratch with That of A-r-Star Reusing the Previous DAG

Table 3 shows a time comparison for the multiple-goal search on the maze in Fig. 5. Different destinations were searched in turn as shown in the goal column of the Table 3. The environment is kept constant throughout the search. Also, s_{Start} was kept constant at $(280,90)$. In the figure, blue = previous path; magenta = plan from scratch; Green = replan using previous DAG; black = obstacle nodes; gray = the multi-resolution grid built by A_r^*; white = free spaces. The first scenario plans from scratch anytime it is queried with a new goal but the second one reuses the information from DAG built by previous searches. It can be seen that, reusing the information from previous DAG amounts to a substantial increase in speed depending on how close the new goal is to a leaf in the previous graph.

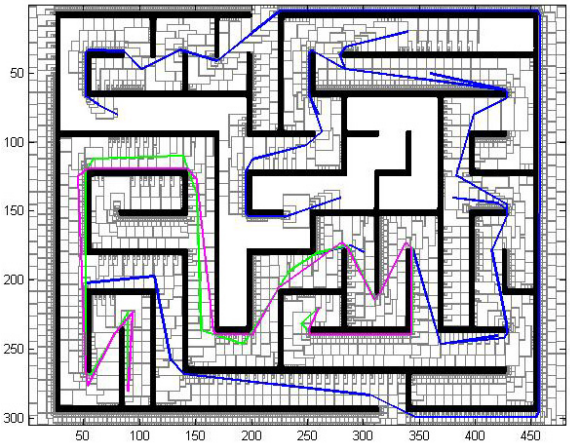


Fig. 5. A simple maze showing the different paths planned and replanned by the A-r-Star

4.2 Comparison of the Incremental-A-r-Star and the D-Star-Lite

Fig. 7 illustrates the re-plan time comparison between the incremental A_r^* and $D^* - Lite$ for the world map shown in Fig. 6. The cells that were blocked have been indicated on the horizontal axis. Also note that the node blocking was accumulative, meaning when a node is blocked, it remains blocked in the next iteration. The incremental A_r^* Algorithm outperforms $D^* - Lite$ when the blocked node is farther from the root node but $D^* - Lite$ outperforms the incremental A_r^* when the blocked node is closer to the root node.

Table 3. The run time comparison for the A-r-Star algorithm searching multiple number of times in a static environment when it reuses the previous information and when it plans from scratch

Goal	Plan From Scratch	Use Previous DAG
(80, 80)	4.97	4.96
(80, 260)	4.85	0.07
(20, 200)	2.73	0.05
(20, 340)	5.70	1.04
(50, 360)	4.84	0.02
(140, 280)	5.40	0.15
140, 380)	4.13	0.02
(180, 300)	2.16	0.01
(240, 420)	2.88	0.01
(220, 260)	3.67	0.02

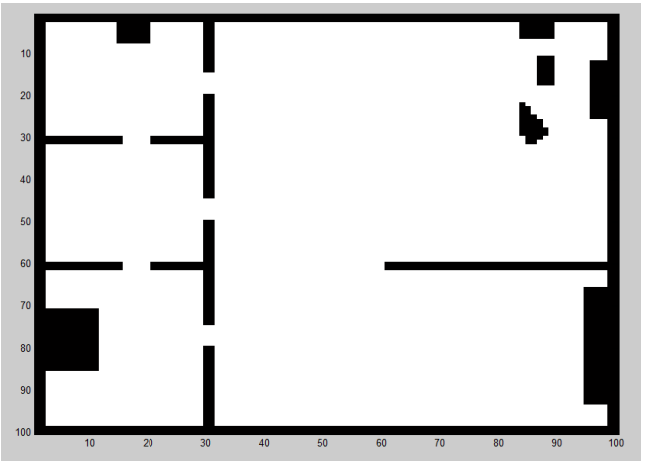


Fig. 6. The grid map used for the simulation

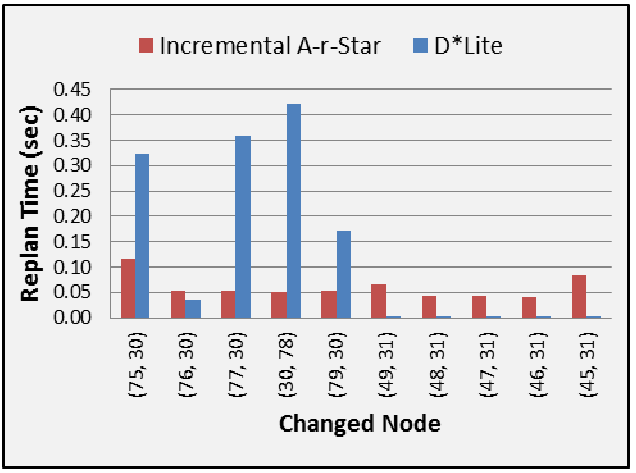


Fig. 7. Results for the incremental A-r-Star search compared to D*-lite

5 Conclusion and Future Work

This research has presented A_r^*pathfinder along with the systematic way of developing an incremental version of A_r^* to handle dynamic environments. The result of this simulation has been compared to that of $D^* - Lite$. This shows that, the average performance of the incremental A_r^* pathfinder can be up to 3 times faster when a previously searched path is blocked far from the root node but can be worse than that of $D^* - Lite$ when the blocked node is near to the start node. This is due to the challenge stated earlier in subsection 3.4 about the incremental A_r^*. An extended research can look into better ways of implementing the PRUNE-BRANCH when a path becomes blocked near the root node.

Acknowledgment. This material is based in part upon work supported by the National Science Foundation (NSF) under Cooperative Agreement No. DBI-0939454. Any opinions, findings and conclusions are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. Noykov, S., Roumenin, C.: Occupancy grids building by sonar and mobile robot. Robot. Auton. Syst. 55(2), 162–175 (2007)
2. Hart, P.E., Nilsson, N.J., Raphael, B.: A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics 4(2), 100–107 (1968)
3. Stentz, A.: Optimal and efficient path planning for partially-known environments. In: Proceedings of the 1994 IEEE International Conference on Robotics and Automation (1994)
4. Koenig, S., Likhachev, M.: D*lite. In: Eighteenth National Conference on Artificial Intelligence 2002, pp. 476–483. American Association for Artificial Intelligence, Edmonton (2002)
5. Koenig, S., Likhachev, M., Furcy, D.: Lifelong planning A*. Artif. Intell. 155(1-2), 93–146 (2004)
6. Opoku, D., Homaifar, A., Tunstel, E.: The A-r-Star (Ar*) Pathfinder. International Journal of Computer Applications, 32–44 (2013)
7. Stentz, A.: Optimal and Efficient Path Planning for Unknown and Dynamic Environments. International Journal of Robotics & Automation 10(3), 89–100 (1995)
8. Schroeder, W.J., Zarge, J.A., Lorensen, W.E.: Decimation of triangle meshes. SIGGRAPH Computer Graph. 26(2), 65–70 (1992)
9. Huang, C.T., Mitchell, O.R.: A Euclidean distance transform using grayscale morphology decomposition. IEEE Transactions on Pattern Analysis and Machine Intelligence 16(4), 443–448 (1994)
10. Matsumoto, Y., Ino, T., Ogasawara, T.: Development of intelligent wheelchair system with face and gaze based interface. In: Proceedings of the Robot and Human Communication, pp. 262–267 (2001)
11. SungHwan, A., et al.: On-board odometry estimation for 3D vision-based SLAM of humanoid robot. In: International Conference on Intelligent Robots and Systems (IROS) (2012)

Comparative Analysis of Evaluation Algorithms for Decision-Making in Transport Logistics

Yuriy P. Kondratenko[1], Leonid P. Klymenko[2], and Ievgen V. Sidenko[1]

[1] Department of Intelligent Information Systems
Petro Mohyla Black Sea State University, 68-th Desantnykiv str. 10,
54003 Mykolaiv, Ukraine
y_kondratenko@rambler.ru, emoty@mail.ru
[2] Department of Ecology
Petro Mohyla Black Sea State University, 68-th Desantnykiv str. 10,
54003 Mykolaiv, Ukraine
rector@chdu.edu.ua

Abstract. The analysis of existing methods and approaches for solving transport logistics problems was performed in this paper, particularly, for optimal choice of transport company. In the working process the complex of decision making criteria was formed and the hierarchical structure of decision support system (DSS) for corresponding tasks was made. Thereby the list of different-type methods (classical and fuzzy) for synthesis of developed DSS was defined. A comparative analysis of the application of fuzzy analytic hierarchy process and the method based on fuzzy inference was held for synthesis DSS for the optimal choice of transport company. The final results prove the effectiveness and reasonability of using fuzzy modeling in problems of transport logistics.

Keywords: fuzzy AHP, fuzzy inference system, decision support system, transport logistics.

1 Introduction

Decision support systems (DSS) for solving different-type problems help for decision makers (DM) to form and use corresponding databases of a priory and current data, models, algorithms and criteria of making effective decisions [1] in automatic and interactive modes.

Different methods, models, theories and algorithms are used for analysis and creation of alternative decisions in DSS [1], among them are: intelligent analysis of data, simulated and fuzzy modeling, genetic algorithms, neural networks, decision making theory, fuzzy-sets theory and fuzzy logics, etc.

Any intelligent system can be presented as generalized model, the structure of which is described with the help of corresponding approaches and mathematical relations upon availability of clearly defined (formalized) information, which can be presented by quantity characteristics. Thereby the necessity in processing fuzzy, in other words quality information, which is hard or impossible to formalize, becomes more actual [2].

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_20,

Information systems and program complexes play important role in transport logistics as they serve for analysis, planning and supporting of decision making processes, and also for providing the necessary level of services quality and increasing of transport cargo traffic effectiveness [3].

2 The Analysis of Researches and Publications

Today there exists a lot of publications on the research of DSS based on the fuzzy logics [4-6], which examine methods of the theory of fuzzy sets for modeling, analysis and synthesis of intelligent hierarchically-organized systems. Researches, which are conducted in different countries, have proved that for many subjects of management, parameters of which change in the process of operation, it is appropriate to use fuzzy computerized automatic control systems [7, 8].

The research of fuzzy logics was associated with the necessity of intelligent systems development, which is able to interact with a human taking from him verbal (fuzzy) information. For this a new mathematical tool is needed, which translates ambiguous statements to the language of clear and formal mathematical formulas.

Fuzzy systems comparing to others have a list of advantages [9]:

– possibility of processing and analysis of fuzzy input data;
– fuzzy criteria formalization of estimation and comparison;
– qualitative evaluations of input information as well as output results;
– quick simulation of complex dynamic systems and their comparative analysis with a given degree of accuracy.

In the process of DSS development based on fuzzy logical inference there is a possibility of sharp increase in the number of fuzzy rules, which leads to difficulty of their formalizing and increasing of simulation time. This is due to the fact that when there are a great number of input system parameters, it is hard for expert to describe cause-and-effect by means of fuzzy rules as human memory can simultaneously store no more that 7 ± 2 states of investigated system [4].

One of the unsolved problems when using hierarchical approach for developing DSS on basis of fuzzy logical conclusion is complexity of structuring and consideration of a large number of input parameters of such systems [7].

Using the theory of fuzzy sets and fuzzy logics when building DSS allows solving problems on intellectual level with the help of fuzzy databases of rules and also provides an opportunity to estimate alternative decisions and choose the best among them [10].

To measure customer's expectations different assessing methods are used, including questionnaires, expert analysis, statistical methods, etc. The difficulty is that most of system parameters cannot be measured quantitatively, so it is difficult to get explicit evaluations. The customer's expectation usually are based on his subjective opinion, experience of his work and more often they are expressed in such statements as "it is desirable that the cargo has been delivered at 12 o'clock", "it is possible to pay in range of 2000 to 3500" and so on. In the relevant statements there are elements of

ambiguity. A tool to formalize fuzzy customer's expectations is a mathematical tool, which is based on the theory of fuzzy sets [3].

Improvement of cargo delivery quality by western carriers is done by means of gathering consolidated cargo from quantity of senders, development of optimal routes of cargo transportation, and also connecting more effective type of transport on the certain stage of transportation [11]. Irrational transportations lead to increase of logistical and first of all transport expenses, and also to additional workload of transport routes [12].

The results of researches [13, 14] showed low effectiveness of firm activity when selecting company-carrier. This is due to the lack of a priory information about the level of tariffs for transportation, the cost and types of services provided by different carriers, etc. Choosing carrier in conditions of planned economy mostly is accompanied by abstract calculations of transport expenses, excluding the impact of quality parameters of forwarding services.

To guarantee that the fuzzy rule based modeling methodology will work in all possible situations we must make sure that every possible system can be obtained by applying the fuzzy rule based modeling methodology to appropriate rules [15].

Analysis of existing methods for optimal choice of transport company allows to determine a list of disadvantages, among them are: difficulties in formalizing procedures of choosing company-carrier; inability to take into account additional factors; evaluation of cargo physical parameters only and way of transportation; use of transitivity for quality indicators [14].

The purpose of this research is a development of the intelligent DSS for multicriteria decision making in problems of transport logistics, analysis of synthesis distinctive features and modeling hierarchically-organized DSS on the basis of fuzzy logics, and also conduction of comparative analysis of results of DSS work using fuzzy analytic hierarchy process (FuzzyAHP) and the method based on fuzzy inference (FIS) for optimal choice of transport company.

3 The Synthesis of DSS for Optimal Choice of Transport Company Based on FuzzyAHP

The analysis of literature sources [3, 11, 14] allows marking main factors that influence of the process of selection of company-carrier. Fig. 1 shows hierarchical structure of decisions tree, which includes the main goal – choice of transport company, in the second hierarchical level are criteria: price, time of delivery, reliability, in the third hierarchical level are subcriteria: proposals for discounts, cost of services, delivery time, timeliness of delivery, image of the company, possibility of cargo insurance, integrity of cargo and in the last (fourth) hierarchical level are alternative variants: companies A, B, C, D with the help of which the consumer of transport services chooses the best transport company for cargo transporting. The problem of optimal selection of company-carrier is a multicriteria decision making task, that's why there is a possibility of using alternative methods: AHP, FuzzyAHP and others [16].

In classical formalization of AHP human judgments are represented as accurate numbers in the selected assessments scale. In the paper [17] it was proposed to use fuzzy numbers with tent belonging functions for assessments.

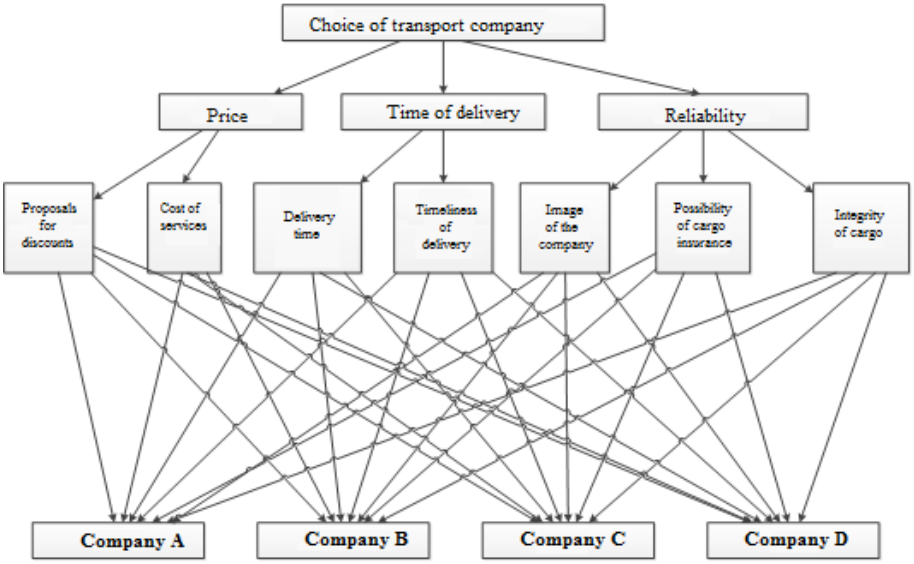


Fig. 1. Hierarchical structure of decisions tree for the optimal choice of transport company

In the study [18] there was researched the problem of fuzzy judgments influence on changing advantages rating. Thereby there was made a conclusion about the fact that increase of the degree of fuzziness (the distance between the left and middle boundaries of fuzzy triangular number) almost did not influence on the vector of priorities. The method proposed in the study of Chang [19] allows getting weights from fuzzy pairwise comparison matrixes (PCM). The result of using Chang method is a vector of priorities, which has point value.

Let's consider fuzzy comparison matrix:

$$\tilde{A} = \left(\tilde{a}_{ij}\right)_{n\times n} = \begin{bmatrix} (1,1,1) & (l_{12},m_{12},u_{12}) & \dots & (l_{1n},m_{1n},u_{1n}) \\ (l_{21},m_{21},u_{21}) & (1,1,1) & \dots & (l_{2n},m_{2n},u_{2n}) \\ \vdots & \vdots & \vdots & \vdots \\ (l_{n1},m_{n1},u_{n1}) & (l_{n2},m_{n2},u_{n2}) & \dots & (1,1,1) \end{bmatrix} \quad (1)$$

where $\tilde{a}_{ij} = \left(l_{ij}, m_{ij}, u_{ij}\right)$, $\tilde{a}_{ij}^{-1} = \left(1/l_{ij}, 1/m_{ij}, 1/u_{ij}\right)$ for $i, j = 1,...,n$ and $i \neq j$; $\tilde{a}_{ij}$ – element of PCM; l_{ij} – lower bound of fuzzy number $\tilde{a}_{ij}$; m_{ij} – the most expected value of fuzzy number $\tilde{a}_{ij}$; u_{ij} – upper bound of fuzzy number $\tilde{a}_{ij}$.

The method consists of several steps [18-20].

Step 1. The sums on the matrix rows are calculated according to formula 2 by means of fuzzy arithmetic:

$$RS_i = \sum_{j=1}^{n} \tilde{a}_{ij} = \left(\sum_{j=1}^{n} l_{ij}, \sum_{j=1}^{n} m_{ij}, \sum_{j=1}^{n} u_{ij} \right), i = 1,...,n, \quad (2)$$

where RS_i – sum of i row of matrix; i – order number of PCM row; j – order number of PCM column; n – total number of comparison elements.

Step 2. The corresponding amounts are normalized according to formula 3:

$$\tilde{S}_i = \frac{RS_i}{\sum_{j=1}^{n} RS_j} = \left(\frac{\sum_{j=1}^{n} l_{ij}}{\sum_{k=1}^{n}\sum_{j=1}^{n} u_{kj}}, \frac{\sum_{j=1}^{n} m_{ij}}{\sum_{k=1}^{n}\sum_{j=1}^{n} m_{kj}}, \frac{\sum_{j=1}^{n} u_{ij}}{\sum_{k=1}^{n}\sum_{j=1}^{n} l_{kj}} \right), i = 1,...,n, \quad (3)$$

where $\tilde{S}_i$ – a normalized amount value of matrix i row; RS_i – sum of i row of matrix

Step 3. The degree of hypothesis conformity is calculated $S_i \geq S_j$ according to formula 4:

$$V(\tilde{S}_i \geq \tilde{S}_j) = \begin{cases} 1, & \text{if } m_i \geq m_j \\ 0, & \text{if } l_j \leq u_j \\ \dfrac{u_i - l_j}{(u_i - m_i) + (m_j - l)}, & other \end{cases} \quad (4)$$

where $V(\tilde{S}_i \geq \tilde{S}_j)$ – degree of hypothesis conformity $S_i \geq S_j$; $\tilde{S}_i$ – normalized amount value of matrix i row, $\tilde{S}_i = (l_i, m_i, u_i)$; $\tilde{S}_j$ – normalized amount value of matrix j row $\tilde{S}_j = (l_j, m_j, u_j)$.

Step 4. The degree of admissibility of the fact that S_i is better than the rest $(n-1)$ fuzzy numbers is calculated according to formula 5:

$$V\left(\tilde{S}_i \geq \tilde{S}_j \mid j = 1,..,n; j \neq i\right) = \min_{j \in \{1,..,n\}, j \neq i} V\left(\tilde{S}_i \geq \tilde{S}_j\right), i = 1,...,n, \quad (5)$$

where $V\left(\tilde{S}_i \geq \tilde{S}_j \mid j=1,..,n; j \neq i\right)$ – the degree of admissibility of the fact that S_i is better than the rest $(n-1)$ fuzzy numbers.

The vector of priorities $W=\left(w_1,...,w_n\right)^T$ of fuzzy matrix is calculated according to formula 6.

$$w_i = \frac{V\left(\tilde{S}_i \geq \tilde{S}_j \mid j=1,..,n; j \neq i\right)}{\sum_{k=1}^{n} V\left(\tilde{S}_k \geq \tilde{S}_j \mid j=1,..,n; j \neq k\right)}, i=1,...,n, \quad (6)$$

where $V\left(\tilde{S}_i \geq \tilde{S}_j \mid j=1,..,n; j \neq i\right)$ – the degree of admissibility of that fact that S_i is better than the rest $(n-1)$ fuzzy numbers; $V\left(\tilde{S}_k \geq \tilde{S}_j \mid j=1,..,n; j \neq i\right)$ – the degree of admissibility of the fact that S_k is better than the rest $(n-1)$ fuzzy numbers.

Let's consider in more details the features of four-level DSS functioning, which are synthesized by authors on the basis of FuzzyAHP method. Hierarchical structure of system for optimal selection of transport company is designed in Java Enterprise. The main window of intelligent DSS is shown on Fig. 2.

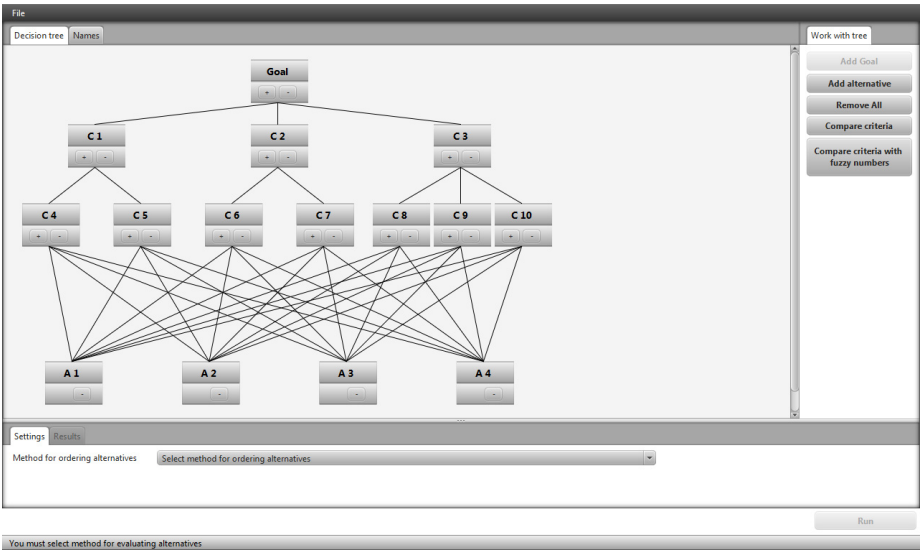


Fig. 2. DSS for optimal choice of transport company based on FuzzyAHP method

For estimate the relative importance of DM judgments [17] proposed to use the following fuzzy triangular numbers: (1,1,1), (1,2,3), (2,3,4), (3,4,5), (4,5,6), (5,6,7), (6,7,8), (8,9,9).

In Tables 1, 2 there is shown the evaluation of importance of criteria of second and third hierarchy levels using the scale of relative importance with fuzzy numbers.

Table 1. The comparison of importance of criteria of second hierarchy level

	Price (C1)	**Time of delivery (C2)**	**Reliability (C3)**
Price (C1)	(1,1,1)	(1/4,1/3,1/2)	(1/3,1/2,1)
Time of delivery (C2)	(2,3,4)	(1,1,1)	(1/3,1/2,1)
Reliability (C3)	(1,2,3)	(1,2,3)	(1,1,1)

Table 2. The comparison of importance of subcriteria of Price criteria (C1)

Price (C1)	**Proposals for discounts (C4)**	**Cost of services (C5)**
Proposals for discounts (C4)	(1,1,1)	(1/4,1/3,1/2)
Cost of services (C5)	(2,3,4)	(1,1,1)

In Table 3 there is shown the comparison of alternatives by criteria “Proposals for discounts (C4)” using fuzzy numbers.

Table 3. The comparison of alternatives by criteria “Proposals for discounts (C4)” using fuzzy numbers

Proposals for discounts (C4)	**Company A (A1)**	**Company B (A2)**	**Company C (A3)**	**Company D (A4)**
Company A (A1)	(1,1,1)	(6,7,8)	(1/4,1/3,1/2)	(8,9,9)
Company B (A2)	(1/8,1/7,1/6)	(1,1,1)	(1/9,1/9,1/8)	(1,1,1)
Company C (A3)	(2,3,4)	(8,9,9)	(1,1,1)	(8,9,9)
Company D (A4)	(1/9,1/9,1/8)	(1,1,1)	(1/9,1/9,1/8)	(1,1,1)

Pairwise comparisons of all items carried at each hierarchical level of DSS for optimal choice of transport company.

4 The Synthesis of DSS for Optimal Choice of Transport Company Based on Fuzzy Inference

FuzzyAHP for synthesis DSS has several limitations and problems. Appropriate method allows you to assign a zero weight of criteria solution or alternative. This leads to excluding criterion or alternative from consideration in the decision making process [20]. Also FuzzyAHP based on Extent Analysis Method allows incompleteness estimates in a PCM. In PCM are imposed limits on the number of elements of comparison. This is due to the fact that the expert must simultaneously consider all factors, since assigning specific numerical value of a particular factor; the expert must also match it with the other. Appropriate method cannot evaluate the consistency of fuzzy PCM, that there is a problem of checking the adequacy of expert judgments of logical thinking.

The authors proposed an approach based on fuzzy inference for synthesis DSS, which takes into account a greater number of input parameters than FuzzyAHP. The expert must only generate fuzzy rules. The choice of transport company based on the overall assessment of the quality of transport services.

The analysis of literature sources [3, 12] shows that among input parameters, which influence on assessment of the quality of transport services, for structural organization of DSS, can be defined following, there are: x_1 – customs costs; x_2 – expenses connected with possible situations on the way; x_3 – expenses on transportation; x_4 – authenticity of information about cargo movement; x_5 – efficiency of reporting information; x_6 – a risk during transportation; x_7 – saving according to amount of cargo; x_8 – saving according to quality of cargo; x_9 – delivery performance; x_{10} – possibility of cargo delivering to any part of territory; x_{11} – readiness for delivery. Output signal of DSS is the assessment of transport services quality (y).

Before the formation process of fuzzy data bases it is important to determine the number and type of linguistic term (LT) for assessment of input and output parameters. For assessment of input coordinates ($x_i, i = 1,...,N$) 3 LT were chosen (L – "low", M – "medium", H – "high"), for output variable – 5 (L – "low", LM – "lower then medium", M – "medium", MH – "higher then medium", H – "high") with triangular shape of belonging function.

The structure of DSS (Fig. 3) is designed in such way that the number of inputs of each subsystem does not exceed five. This allows decreasing the number of database rules, thus increasing the sensitivity of system to influence of input coordinates. Thereby it is important to conduct structuring of input variables only by common properties, which are the main (important) within a single subsystem [4]. It is advisable to combine input coordinates in the following groups: $y_1 = f_1(x_1, x_2, x_3)$, $y_2 = f_2(x_4, x_5)$, $y_3 = f_3(x_7, x_8)$, $y_4 = f_4(x_9, x_{10}, x_{11})$, $y_5 = f_5(x_6, y_2)$, $y_6 = f_6(y_2, y_5)$,

$y=f_7(y_1,y_4,y_6)$. Where $y_i, i=1,...,6$ are intermediate variables of DSS module, including: y_1 – cost of transportation; y_2 – the level of informativeness of cargo delivery; y_3 – safety of cargo; y_4 – image of subjects-participants of cargo transportation; y_5 – reliability of delivery system; y_6 – level of cargo transportation.

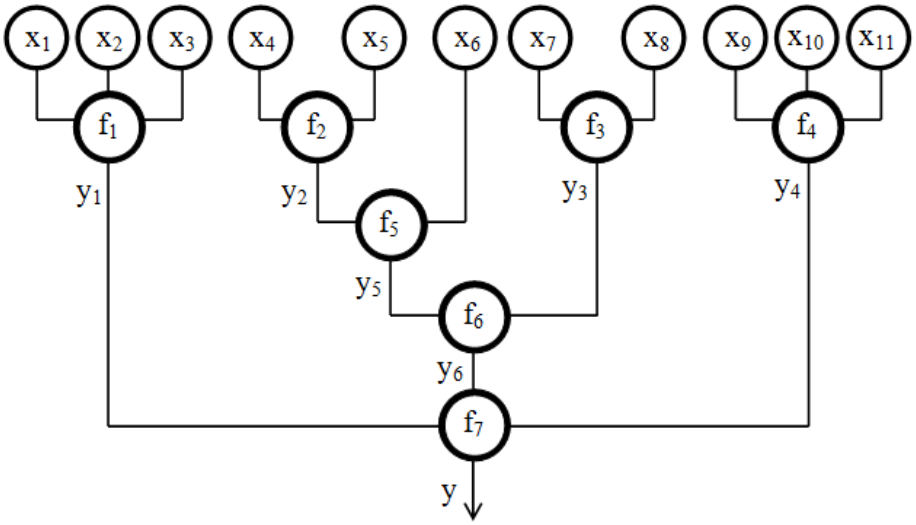


Fig. 3. The structure of DSS for assessment of transport services quality

In constructing fuzzy knowledge bases for DSS structure (Fig. 3) there are used 3 LT with triangular shape of membership function that are presented for variables $\{x_1, x_2, ..., x_{11}, y_2, y_3, y_5, y_6\}$ in Fig. 7, for variables $\{y_1, y_4, y\}$ – 5 LT in Fig. 7.

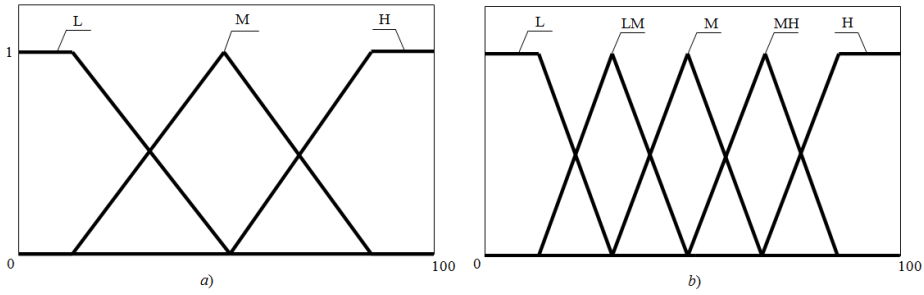


Fig. 4. Linguistic terms for variables $\{x_1, x_2, ..., x_{11}, y_2, y_3, y_5, y_6\}$ (*a*) and for variables $\{y_1, y_4, y\}$ (*b*) Selective set of rules for the first subsystem $y_1 = f_1(x_1, x_2, x_3)$ can be presented in Table 4

After describing fuzzy system and developing databases of fuzzy rules, synthesis of fuzzy DSS module is done in FuzzyTECH. Any system of fuzzy derivation in program environment FuzzyTECH is shown as separate project [6]. There are several software products that allow developing DSS based on fuzzy logical. Widely famous is the software package MatLab, which includes tools "fuzzy" for development of such class of systems [6].

Table 4. Rules for the first subsystem $y_1 = f_1(x_1, x_2, x_3)$

№ of rule	**1**	**3**	**6**	**10**	**13**	**14**	**15**	**17**	**22**	**25**	**27**
x_1	L	L	L	M	M	M	M	M	H	H	H
x_2	L	L	M	L	M	M	M	H	M	H	H
x_3	L	H	H	L	L	M	H	M	L	L	H
y_1	L	LM	M	L	LM	M	MH	M	M	MH	H

The characteristic surface for the first rules database of the first fuzzy subsystem $y_1 = f_1(x_1, x_2, x_3)$ is presented in Fig. 5.

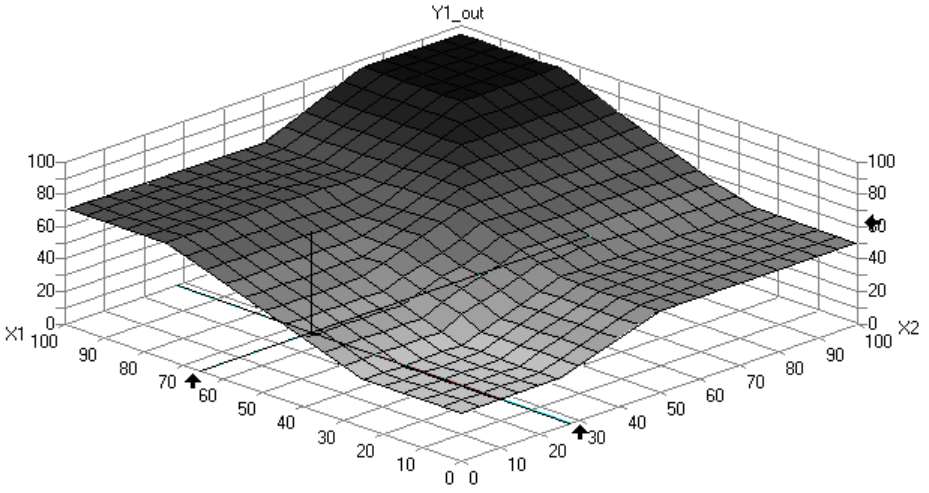


Fig. 5. The characteristic surface of the first subsystem for the components (x_1, x_2)

In the process of fuzzy DSS work with a fixed structure of the rule bases and at a variable structure of the vector of input data $N_r < N$, the results of making decisions y undergo deformation. This is due to the fact that the values of the input parameters (signals) that do not take part in modeling of fuzzy DSS $(x_i = 0, i \in \{1, 2, ..., N\})$, carry

out negative impact on the result y through the appropriate fuzzy rules. To solve this problem the authors have developed an approach (based on two algorithms of editing of rules antecedents and consequents), which consists in correction of the rules of fuzzy rule bases at variation of input parameters that allows not to take into account the values of input signals $x_i = NI, i \in \{1, 2, ..., N\}$, which are not important for DM in the process of making decisions [21].

Method of correction (editing) the rules of fuzzy knowledge bases at different number of input coordinates of the system [21], which developed by the authors, can be used in developed DSS for optimal choice of transport company.

5 The Analysis of DSS Work Results

According to results of DSS work for the optimal choice of transport company, using FuzzyAHP, the vector of priorities of alternative variants corresponds to the following values:

- transport company A – 0,223;
- transport company B – 0,158;
- transport company C – 0,262;
- transport company D – 0,357.

The best variant of cargo transporting is company-carrier D with the priority 0,357.

The results of modeling DSS for optimal choice of transport company based on the overall assessment of the quality of transport services, using fuzzy inference, for different companies (A, B, C, D) are shown in Table 5 and Table 6.

Table 5. The value of the input data of transport companies

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}
A	75	90	35	80	60	45	70	90	30	95	50
B	25	15	30	90	95	5	35	30	20	35	50
C	10	30	87	90	95	30	60	75	55	37	75
D	***33***	***90***	***85***	***50***	***30***	***7***	***95***	***80***	***80***	***65***	***95***

Table 6. The results of modeling DSS for optimal choice of transport company

	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y
A	58	94	85	33	61	83	57
B	4	93	17	29	94	61	34
C	33	92	68	66	85	82	54
D	***55***	***15***	***94***	***96***	***59***	***94***	***74***

From the results of DSS work for assessment of transport services quality (Fig. 3, Table 5, 6) we received the following values:

– transport company A – 57 balls;
– transport company B – 34 balls;
– transport company C – 54 balls;
– transport company D – 74 balls.

According to the results of modeling it is clear that among all transport companies with assessment of transport services quality (74 balls) the best one is the fours (D) transport company, and the worst mark has the third (C) transport company (34 balls).

As a result of the algorithm based on fuzzy inference, the optimal is the score that corresponds to the maximum number of balls. According to this algorithm at the input of DSS for the optimal choice of transport company the expert sets clear values of input parameters (X_1, X_2, ..., X_{11}) of the specific transport company (Eg., company A in Table 5). Thus by using fuzzy rules (Table 4) at the output of the DSS the assessment of the corresponding transport company is formed (for example, company A - 57 balls from Table 6). Table 5 shows that the result of the algorithm (Y = 74) is optimal in terms of the DM for the optimal choice of transport company.

Checking the adequacy of the results of the DSS for the optimal choice of transport company (Table 5, 6) is carried by comparative analysis of the proposed method, particular, method based on fuzzy logic inference (FIS), and well-known FuzzyAHP for different sets of criteria (input parameters). Moreover, the five experts were involved. Let's consider corresponding sets of input parameters:

– I set consists of 2 parameters: delivery price, timeliness of delivery;
– II set consists of 3 parameters: I set + risk during transportation;
– III set consists of 4 parameters: II set + saving of cargo;
– IV set consists of 5 parameters: III set + readiness for delivery.

The simulation results of DSS for the optimal choice of transport company using FuzzyAHP method (Table 7) and method based on FIS (Table 8) math by the optimal decision, that proves the adequacy of the proposed method based on FIS.

Table 7. The simulation results of DSS using FuzzyAHP for different sets of input parameters

Company / Set of parameters	Company A (A1)	Company B (A2)	Company C (A3)	Company D (A4)
I	**0,286**	0,182	0,269	0,263
II	0,214	0,196	**0,315**	0,275
III	0,307	0,114	0,233	**0,346**
IV	0,256	0,204	0,228	**0,312**

Table 8. The simulation results of DSS using FIS for different sets of input parameters

Company / Set of parameters	Company A (A1)	Company B (A2)	Company C (A3)	Company D (A4)
I	**62**	36	54	51
II	53	47	**60**	57
III	66	50	65	**72**
IV	60	42	51	**68**

The diagram of distribution of the vector of priorities for the optimal choice of transport company using appropriate methods for IV set of input parameters (Table 7, 8) is presented in Fig. 5. Also carried out normalizing the vector of priorities for the method based on FIS.

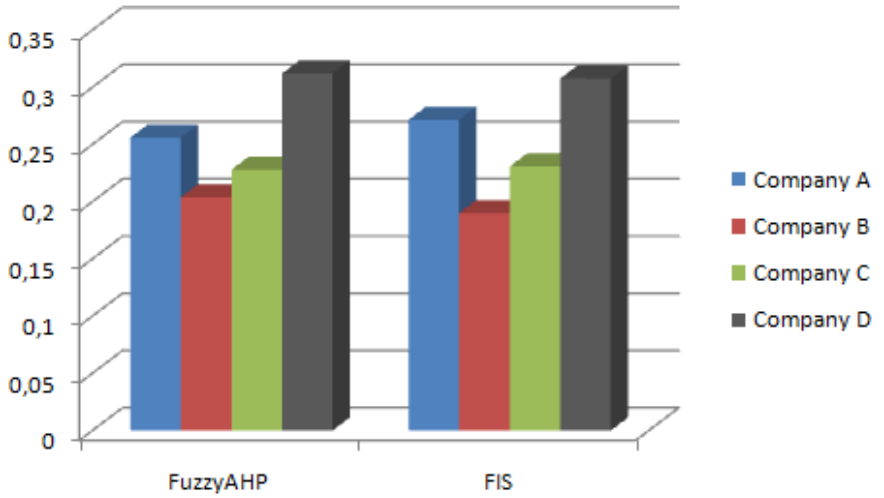


Fig. 6. The diagram of distribution of the vector of priorities for IV set of input parameters

Checking the work of proposed method based on the fuzzy inference was carried out to solve the problem of assessing the quality of transport services [12, 21]. In the process of developing the appropriate DSS using the method of analytic hierarchy process and FuzzyAHP method to assess the quality of transport service encountered a number of problems associated with the restriction of input DSS coordinates. As their number were 19, so it made sense to use a method based on fuzzy inference, which had no relevant restrictions.

6 Conclusions

DSS simulation results proved that the use of fuzzy logics is appropriate in cases when the criteria for decision-making, influencing factors and parameters of the real system are too complex for analysis using quantitative methods.

In the case when necessary to consider a large number of input parameters appropriate to use fuzzy inference method for synthesis DSS.

Synthesized DSS for the optimal choice of transport company is a universal software complex for solving hierarchically-organized multicriteria problems using diverse methods of decision making.

The analysis of fuzzy multicriteria decision-making methods allows to formulate requirements for future developments in this area, including the development of theoretical approaches to describe the complex relationships between the criteria.

References

1. Ross, T.J.: Fuzzy Logic with Engineering Applications. Wiley, California (2004)
2. Kondratenko, V.Y., Yatsenko, V.S.: Object-Oriented Models for the Synthesis of Intelligent Systems with Fuzzy Logic. In: Proceedings of the Odessa National Polytechnic University, vol. 26(4), pp. 44–52 (2006) (In Ukrainian)
3. Mirotin, L.B.: Transport Logistics. Publishing House Examen, Moscow (2005) (in Russian)
4. Yeh, C.-H., Deng, H., Chang, Y.-H.: Fuzzy Multicriteria Analysis for Performance Evaluation of Bus Companies. European Journal of Operational Research 126(3), 459–473 (2000)
5. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
6. Messarovich, M.D., Macko, D., Takahara, Y.: Theory of Hierarchical Multilevel Systems. Academic Press, New York (1970)
7. Yager, R., Filev, D.: Essential of Fuzzy Modeling and Control. Wiley, New York (1994)
8. Zimmerman, H.J.: Fuzzy Set Theory. Kluwer, Boston (1991)
9. Piegat, A. (ed.): Fuzzy Modeling and Control. STUDFUZZ, vol. 69. Springer, Heidelberg (2001)
10. Babuska, R., Verbruggen, H.B.: A New Identification Method for Linguistic Fuzzy Models. In: International Conference FUZZ-IEEE/IFES, Yokohama, pp. 905–912 (1995)
11. Stroh, M.B.: A Practical Guide to Transportation and Logistics. Logistics Network Inc., Boston (2006)
12. Kondratenko, Y.P., Encheva, S.B., Sidenko, I.V.: Synthesis of Intelligent Decision Support Systems for Transport Logistic. In: IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, Prague, pp. 642–646 (2011)
13. Davidsson, P., Henesey, L., Ramstedt, L., Tornquist, J., Wernstedt, F.: An Analysis of Agent-based Approaches to Transport Logistics. Transportation Research Part C: Emerging Technologies 13(4), 255–271 (2006)
14. Meade, L., Sarkis, J.: Strategic Analysis of Logistics and Supply Chain Management Systems Using the Analytical Network Process. Transportation Research Part E: Logistics and Transportation Review 34(3), 201–215 (1998)

15. Kreinovich, V., Mouzouris, G.S., Nguyen, H.T.: Fuzzy Rule Based Modeling as a Universal Approximation Tool. Fuzzy Sets 2, 135–195 (1998)
16. Saaty, T.: Decision Making Hierarchy Analysis Method. Radio and Communication, Moscow (1993) (in Russian)
17. Laarhoven, V., Pedrych, W.: Fuzzy Extension for Saaty's Priority Theory. Fuzzy Sets and Systems 11, 229–241 (1983)
18. Narasimha, B., Chen, N.: Effect of Imprecision in Specification of Pair-wise Comparisons on Ranking of Alternatives using Fuzzy AHP. In: AMCIS, Boston, pp. 238–243 (2001)
19. Chang, D.Y.: Applications of the Extent Analysis Method on Fuzzy AHP. European Journal of Operational Research 95, 649–655 (1996)
20. Wang, Y.M., Luo, Y., Zhongsheng, H.: On the Extent Analysis Method for Fuzzy AHP and Its Applications. European Journal of Operational Research 186, 735–747 (2008)
21. Kondratenko, Y.P., Sidenko, I.V.: Decision-Making and Fuzzy Estimation of Quality Level for Cargo Delivery. In: Proceeding of the 2nd World Conference on Soft Computing, Baku, pp. 418–423 (2012)

Handling Big Data with Fuzzy Based Classification Approach

Neha Bharill and Aruna Tiwari

Department of Computer Science and Engineering
Indian Institute of Technology
Indore, India
{phd12120103,artiwari}@iiti.ac.in

Abstract. Big data is a collection of very large and complex data that is difficult to load into the computer memory. The major challenges include searching, categorization and analysis of big data. In this paper, a fuzzy based supervised classifier is proposed to handle the searching, storage and categorization of big data. In this classifier, we proposed a Random Sampling Iterative Optimization Fuzzy c-Means (RSIO-FCM) clustering algorithm which partitions the big data into various subsets. These subsets adequately cover all the instances (object space) of big data. Then, clustering is performed on these subsets by feeding forward the centers of clustered subset to group remaining subsets. Further, the designed classifier based on Bayesian theory is used to assign the labels to these clusters and also used to predict labels of unknown instances. Thus, the proposed approach results in effective clusters formation which also eliminates the problem of overlapping cluster centers faced by algorithm discussed in [1] named as Simple Random Sampling plus Extension FCM (rseFCM). The effectiveness of proposed clustering algorithm over rseFCM clustering is evaluated on two very large benchmark datasets in terms of fuzzification parameter m, objective function, computational time and accuracy. Experimental results demonstrate that, the RSIO-FCM algorithm generates more appropriate cluster centers location due to which it achieves better classification accuracy as compared to the rseFCM algorithm. Thus, it observed that, cluster centers location will have significant impact over classification results.

1 Introduction

Big data is a collection of vast amount of data that is difficult to handle with existing computer memory [1]. The abundant amount of information generated from the various social networking sites, especially Facebook [1] alone logs produced 25 terabytes (TB) of data per day. Handling of such big data with existing resources is the major challenge. The challenges include categorization, searching and sharing and visualization of big data with limited resources. Clustering and classification are primary tasks used in pattern recognition to effectively handle the storage, searching and categorization of data from very large (VL) databases. Clustering [2] is a process of grouping data into manageable parts such that, the

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_21,

samples in each group share some similarity with each other. Classification is used for categorization or identification of samples, which determines the set of categories, the new observation or sample belongs to. Hence, both the approaches jointly handle these challenges associated with big data.

Many algorithms have been proposed to perform clustering of big data but, only few of them address the fuzzy clustering problems. The literal FCM, clusters the entire dataset and works well only for small range datasets. In contrast to the literal schemes, simple random sampling plus extension FCM (rseFCM)[1] approach is designed to perform clustering on VL data. It works on representative samples taken from very large data to perform clustering. However, the rseFCM suffers from overlapping cluster centers because the representative sample does not cover all the objects present in VL data, thus the big dataset cannot be grouped appropriately.

In this paper, the problem of overlapping cluster centers and challenges associated with big data are overcome with Random Sampling Iterative Optimization Fuzzy c-Means (RSIO-FCM) algorithm. The proposed algorithm generates various subsets of big data. These subsets covers all the objects present in big data. Then, it performs clustering on these subsets by feeding forward the cluster centers location of one subset to group remaining subsets. Thus, it generates non overlapping cluster centers location and works significantly well for big datasets. Finally, the designed classifier which is dependent on the cluster centers and based on the concept of Bayesian theory is used to predict the class labels of unknown samples. The designed classifier is dependent on the cluster centers therefore, cluster centers location will have significant impact over classification results.

Section 2 describes the Review of related work. Section 3, describes the proposed RSIO-FCM clustering algorithm for very large data. Then, we apply the clustering results of both the approaches on classification mechanism based on Bayesian theory. In section 4, we perform experimentation with two very large datasets for demonstrating the effectiveness of proposed approach. Finally, section 5 is presented with concluding remarks.

2 Review of Related Work

Many methods have been proposed by researchers for clustering very large data. Generally, these methods are based on various types of algorithms. Sampling methods, that compute cluster centers on sampled data which is randomly selected from huge dataset include CLARA [3], CURE [4], and the coresets algorithms [5]. These algorithms works well for crisp partitions. Methods that work well to produce fuzzy partition include the *fast* FCM (FFCM) [6], in which literal FCM as discussed in algorithm 1, is iteratively applied for larger nested samples till the change is reflected in the solution; and the multistage random FCM [7] which combines FFCM till the final run of FCM on the full dataset. These algorithm are based on extension of literal fuzzy c-means clustering as discussed in algorithm 1.

The literal fuzzy c-Means algorithm is based on optimization of objective function [1].

$$J_m = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^m \parallel x_i - v_j \parallel^2, 1 \leq m < \infty \tag{1}$$

Algorithm1: *LFCM/AO to iteratively minimize the* J_m

Step2: Compute cluster membership.

$$u_{ij} = \frac{\parallel x_i - v_j \parallel^{\frac{-2}{m-1}}}{\sum_{k=1}^{c} \parallel x_i - v_k \parallel^{\frac{-2}{m-1}}}, \forall i, j \tag{2}$$

Step3: Check the constraint.

$$\sum_{j=1}^{c} u_{ij} = 1 \tag{3}$$

Step4: Compute the cluster centers.

$$v_j = \frac{\sum_{i=1}^{n}[u_{ij}]^m x_i}{\sum_{i=1}^{n}[u_{ij}]^m}, \forall j \tag{4}$$

Step5: $if \parallel V_0 - v_j \parallel < \epsilon$ then stop, otherwise go to step2; $\epsilon = 10^{-3}, m > 1$

Where U is the n×c partition matrix, m is a fuzzification parameter which drastically affects the clustering results, V= $\{v_1, v_2, \ldots, v_c\}$ denotes the set of cluster centers and ϵ denotes the predefined constant.

2.1 Sampling and Noniterative Extension

The most obvious way to address the VL data is to sample the dataset and then use literal FCM to generate cluster centers of the sampled data. Algorithm 2 outlines the approach based on sampling and noniterative extension named as rseFCM. It produces the cluster centers of sampled data by using literal FCM as discussed in algorithm 1 and then, it uses the extension approach in Step 3 of rseFCM to produce partition of full data.

Algorithm2: *rseFCM to approximately minimize the* J_m [1]

Input : X, c, V_0, m
Output: U, V
Step1: Sample the n_s objects from X without replacement, denoted X_s.
Step2: $U_s, V = LFCM(X_s, V_0, m)$
Step3: Extend the partition (U_s, V) to X,$\forall x_i \notin X_s$ using Eq. (2), produce (U, V)

Note: Once the extension step is completed, so that partition U on the full dataset is known, then by using U a completion step yields the cluster centers V with Eq.(4).

The rseFCM algorithm as discussed in [1] is mainly designed to perform clustering of very large data. The rseFCM algorithm works on sampled data taken from entire dataset, then the cluster center locations produced on sampled data will be extended to compute partitions on full data. The rseFCM algorithm suffers from overlapping cluster centers location and significantly affects classification results because, the sampled data does not adequately covers all the objects of big data. Thus, the error between the cluster center locations produced by sampled data and location produced by clustering entire dataset will be drastically higher. Therefore, the rseFCM algorithm approximately minimizes the objective function which drastically deviates from the minimum value of objective function produced by clustering entire dataset.

To design the clustering algorithm that scale well on big data and to overcome the drawbacks of resFCM algorithm, the uniform random sampling with iterative optimization (RSIO-FCM) algorithm is proposed. The RSIO-FCM algorithm divides VL dataset into various subsets. These subsets adequately cover all the objects present in the VL dataset. Then, RSIO-FCM algorithm feed forwards the cluster centers computed by clustering one subset of data to group remaining subsets. Further, to assign the class labels to these clusters and to predict the class labels of unknown samples, a classifier is build based on Bayesian theory. Hence, the proposed algorithm covers the entire object space and results in better cluster formation by eliminating the problem of overlapping cluster centers faced by rseFCM algorithm. It is observed that, the classification accuracy achieved by applying RSIO-FCM algorithm is comparatively higher than rseFCM algorithm. Thus, it is inferred that, the accuracy of classifier greatly depends on the clustering results.

3 Proposed Work

The proposed classifier works in supervised learning scheme, there are two phases of learning: the *training phase* and the *classification phase*. In training phase, Given a training set denoted here as $X_{training} = \{(x_1, y_1), \ldots\ldots (x_l, y_l)\}$ where the first component of l^{th} tuple $x_l = (f_1, \ldots\ldots, f_d)$ denotes the attribute of i^{th} training instance in *d–dimensions*. The second component $Y_i = \{y_1, \ldots\ldots, y_n\}$ denotes the class labels associated with the training instance. The goal here is to form the grouping of similar instance and to learn a mapping from $x \rightarrow y$. In the classification phase, the labels of test instance in X_{test} are predicted using the induced classifier.

3.1 Training of Classifier

In training phase of classifier, the $X_{training}$ is clustered using the proposed RSIO-FCM algorithm as discussed in section 3.1.1.

3.1.1 Uniform Random Sampling with Iterative Optimization: Algorithm 3, outlines the RSIO-FCM clustering approach which effectively clusters very large data by appropriately minimizing the objective function. First, it partitions the VL data into various subsets. Then, it iteratively calls the literal FCM algorithm as discussed in algorithm 1, to compute cluster centers for every subsets by feeding forward the centers of clustered subset to cluster remaining subsets. When the partition on every subset is determined then, using Eq.(5) it computes the partition on full dataset. Once the partition on full dataset is known then, using Eq.(4) final cluster centers on full data are determined.

Algorithm3: *RSIO-FCM to minimize the* J_m

Input : X, c, V_0, m
Output: U, V
Step1: Load X as n_s sized randomly chosen subsets $X = \{x_1, x_2 \ldots\ldots x_s\}$
Step2: Sample x_1 from X without replacement
Step3: $U_l, V_l = LFCM(x_1, V_0, m)$
Step4: for $p = 2$ to s do
$\quad U_p, V_p = LFCM(x_p, V_p - 1, m)$
Step5: Compute the partition on full dataset

$$U = \sum_{i=1}^{s} U_l \tag{5}$$

Step6: Compute cluster centers with partition on full dataset using Eq.(4)
Step7: Compute the objective function using Eq. (1)

The RSIO-FCM algorithm adequately covers the object space by generating cluster centers for every subsets. Therefore, the cluster centers location produced by RSIO-FCM algorithm reflects the actual position as produced by clustering the entire dataset. Due to the proper cluster centers location generated by RSIO-FCM algorithm, it appropriately minimizes the objective function as compared to rseFCM algorithm. To show the effectiveness of clustering results generated by RSIO-FCM algorithm, the classifier is designed based on Bayesian theory. It is used to assign the labels to these clusters and also predict the labels of unknown samples as discussed in section 3.2.

3.2 Classification Mechanism

After grouping the VL data in various clusters, the classifier is designed based on Bayesian theory which assigns class labels to these clusters. The designed classifier is dependent only on the cluster centers. This emphasizes that, the cluster centers location will have significant impact over classification results. To determine the output class labels for unknown samples, a relational matrix **P** is

modeled which establish cluster class relationship through Bayesian theory and also assign labels to these clusters.

$$P(class_l/cluster_k) = \frac{Num(x \in class_l, x \in cluster_k)}{Num(x \in cluster_k)} \tag{6}$$

where $Num(x \in class_l, x \in cluster_k)$ denotes the posterior probability and $Num(x \in cluster_k)$ denotes the prior probability. Now, the relation matrix $\mathbf{P}$ constitutes all the $P(class_l/cluster_k)$ as $K \times L$ matrix. The relation matrix $\mathbf{P}$ determines the statistical relationship between the formed clusters and the given class labels. Next the output class labels $f(object_i)$ can be determined through the total probability theorem as

$$P(class_l/object_i) = \sum_{k=1}^{c} P(cluster_k/object_i)P(class_l/cluster_k) \tag{7}$$

where $P(cluster_k/object_i)$ represents the posterior probability, which determines the presence of corresponding sample in particular cluster and can be computed by Eq.(2) and $P(class_l/cluster_k)$ is computed Eq.(6), then the output class labels $f(object_i)$ can be determined.

$$f(object_i) = \arg_{1 \le l \le L} \max P(class_l/object_i) \tag{8}$$

As presented in Eq.(7),classifier is designed using relational matrix as computed in Eq.(6) which is dependent on the cluster centers. Therefore, the cluster centers location will have significant impact on classification results. Thus, both the approaches are complementary to each other.

4 Experimental Results

The experimentation is performed on machine having Intel (R) Core $(TM)2$ Duo processor with 3.00 GB memory. All the code was written in the MATLAB R2013a computing environment. We evaluate the clustering and classification capability of the proposed RSIO-FCM algorithm and the existing rseFCM algorithm on two benchmark datasets. For both the algorithms, we randomly choose c objects as the initial cluster centers to initialize V_0. In all of our experiment, we fix the value $\epsilon = 10^{-3}$. We have selected these datasets from the University of California at Irvine (UCI) Machine Learning Repository [8]. The basic information of these benchmark datasets is illustrated in table 1 as follows:

Table 1. Basic Information about the Datasets

Datasets	Samples	Features	Classes
Pen-Based Recognition of Handwritten Digits	10992	16	10
Page Blocks Classification	5473	10	5

Table 2. Comparison of rseFCM and RSIO-FCM on Benchmark Datasets

(a)Pen-Based Recognition of Handwritten Digits

	rseFCM			**RSIO-FCM**		
m	objective function	Total time(s)	Accuracy	objective function	Total time(s)	Accuracy
1.2	7833685633	31.79	74.88%	4622236.18	28.56	78.45 %
2.3	11917813.07	3.40	71.99%	10917812.07	2.089	81.89%
3.5	13718180.49	3.38	69.67 %	1271610.49	1.892	82.33 %
4.5	281935.08	2.42	65.96 %	211935.08	1.32	85.34 %

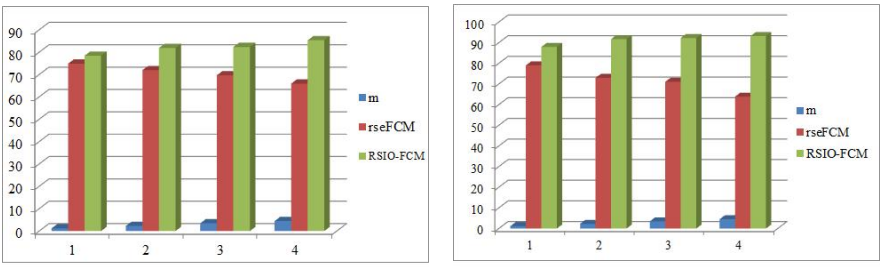
(b)Page Blocks Classification

	rseFCM			**RSIO-FCM**		
m	objective function	Total time(s)	Accuracy	objective function	Total time(s)	Accuracy
1.2	387910619.6	1.54	79.02%	58137603.98	0.560	87.96%
2.3	181503981.8	1.44	72.98%	12096408.21	0.495	91.77%
3.5	79004007.83	1.43	71.09 %	772374.015	0.484	92.34 %
4.5	39501984.9	1.36	63.90 %	70788.88	0.480	93.22 %

The performance of rseFCM and RSIO-FCM are measured on very large data over various parameters. (*a*) The fuzzification parameter m which drastically affects the cluster formation. (*b*) The objective of both the algorithms is to minimize the objective function iteratively. (*c*) The time required in seconds to group the very large data and to compute the cluster centers. (*d*) The classification accuracy which determines the searching of unknown instance from large dataset. The results are reported in table 2.

As reported in Table 2, it is observed that rseFCM algorithm works on minimizing the objective function, since the sampled data does not covers the adequate object space, it fails to appropriately minimize the objective function as compared to minimum value achieved by RSIO-FCM algorithm. The rseFCM algorithm also suffers from overlapping cluster centers. Therefore, it results in degradation of classification accuracy. In contrast to rseFCM, the RSIO-FCM algorithm generates cluster centers for every subset of dataset by covering the entire object space, due to which the achieved objective function value is comparatively more minimized. The RSIO-FCM algorithm generates proper cluster centers location due to which classification accuracy enhances with the increase of fuzzification parameter m by constantly minimizing the objective function. Thus, RSIO-FCM is preferable for big data.

As shown in Fig. 1 with increase of fuzzification parameter m the classification accuracy achieved by RSIO-FCM algorithm is constantly increasing and much higher than the rseFCM algorithm on both datasets. The same is verified with the results reported in Table 2, the RSIO-FCM algorithm shows drastic improvement in performance for very large data as compared to rseFCM algorithm on two benchmark datasets. Thus, accuracy of classifier greatly depends on the clustering.



(a) (b)

Fig. 1. Accuracy Comparison of resFCM and RSIO-FCM
(*a*) Pen-Based Recognition of Handwritten Digits (*b*) Page Blocks Classification

5 Conclusion

The major challenges associated with big data include searching ,categorization and storage of big data. There are many ways to handle these challenges, but the existing approaches does not jointly address these challenges. In this paper, we designed a fuzzy based supervised classifier which jointly addresses these challenges. In this classifier, the proposed RSIO-FCM clustering algorithm overcomes the drawbacks of existing clustering algorithm named as rseFCM algorithm. The RSIO-FCM algorithm divides the VL data into various subsets. Then, it clusters every subset by feeding forward centers of clustered data to cluster remaining subsets. Thus, RSIO-FCM algorithm covers the entire object space adequately and generates same cluster centers location as produced by clustering entire data. Further, the classifier designed based on Bayesian theory is used to assign labels to these cluster. Therefore, the proposed classifier overcomes the drawback of rseFCM algorithm which fails to cover the entire object space adequately and unable to jointly address all these challenges. It is observed that, due to proper cluster center locations generated by RSIO-FCM clustering algorithm, it generates better classification accuracy than rseFCM algorithm. It is inferred that, accuracy of classifier greatly depends on clustering efficiency.

References

1. Havens, T.C., Bezdek, J.C., Leckie, C., Hall, L.O., Palaniswami, M.: Fuzzy c-Means Algorithms for Very Large Data. IEEE Trans. Fuzzy System 20(6), 1130–1146 (2012)
2. Cai, W., Chen, S., Zhang, D.: A Multiobjective Simultaneous Learning Framework for Clustering and Classification. IEEE Trans. on Neural Networks 21(2), 185–200 (2010)
3. Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley-Blackwell, New Work (2005)
4. Guha, L.S., Rastogi, R., Shim, K.: CURE: An efficient clustering algorithm for large databases. Inf. Syst. 26(1), 35–58 (2001)

5. Har-Peled, S., Mazumdar, S.: On coresets for k-means and k-median clustering. In: Proc. ACM Symp. Theory Comput., pp. 291–300 (2004)
6. Shankar, B.U., Pal, N.: FFCM: An efficient approach for large data sets. In: Proc. Int. Conf. Fuzzy Logic, Neural Nets, Soft Comput., Fukuoka, Japan, p. 332 (1994)
7. Cheng, T., Goldgof, D., Hall, L.: Fast clustering with application to fuzzy rule generation. In: Proc. Int. Conf. Fuzzy Syst., Tokyo, Japan, pp. 2289–2295 (1995)
8. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine learning Databases. Dept. Inf. Comput. Sci., Univ. California Irvine, Irvine (1998), `http://www.ics.uci.edu/~mlearn/MLRepository.html`

OWA Based Model for Talent Selection in Cricket

Gulfam Ahamad[1], S. Kazim Naqvi[1], and M.M. Sufyan Beg[2]

[1] FTK-Centre for Information Technology, Jamia Millia Islamia, New Delhi-110025
gulfamahamad.jmi@gmail.com, sknaqvi@jmi.ac.in
[2] Department of Computer Engineering, Jamia Millia Islamia, New Delhi-110025
mmsbeg@hotmail.com

Abstract. Talent selection in cricket is a task which is usually carried out by coaches and senior players. The method relies on instincts or natural abilities of the selectors for talent assessment and selection. However, it suffers with subjectivity, personal biasness and external influences. In country such as India where more than 1-million players play cricket daily, talent selection problem becomes significant. In this paper, we propose a model which can rank players in order of their talent. The model can potentially help reduce the implicit problems of manual talent selection system. The model assesses the cricketing talent of individual players based on the quantitative outcome of the identified parametric tests for assessing players' physical/motor, anthropometric and cognitive skills and capabilities with respect to cricket. The Ordered weighted averaging aggregation (OWA) operator with Relative Fuzzy Linguistic Quantifier (RFLQ) is used to measure the weights and aggregate players' talent values. The model is applied to the Jamia Millia Islamia's (JMI) University Cricket team and results have been summarized.

Keywords: Talent selection in Cricket, OWA, RFLQ, Model.

1 Introduction

Today sports have not remained merely limited to means for keeping healthy and fit. They have rather grown to become lucrative profession options for people with interest. However, only interest is not sufficient to qualify the requirements of becoming a professional sportsperson. A wrong initial choice of sport, probably influenced by family, friends or media may jeopardize careers of young sports enthusiasts. Thus, timely talent identification within a person for a particular sports become significant as it can save the athlete from wasting time on a sports for which (s)he is not suitable. Also a suitable talented sportsperson may quickly excel in the sport which is right for him with minimal efforts. A number of authors [26], [27], [28] have defined talent as an increasable natural endowment of a superior quality of a person. Talent identification is a process to identify the ability of superior quality. It is a complex multifaceted, multidimensional and multi-stage process [22] [23] [24] [25].

Cricket is very popular sports in India. It is formally played in ten nations viz. Australia, Bangladesh, England, India, Pakistan, New Zealand, Srilanka, South Africa,

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_22,

West Indies, and Zimbabwe with 35-associate countries in all. In India alone 55,000 cricket matches are played every day [31]. With such huge popularity a large number of young enthusiasts aspire to become professional cricketers. In India, with very limited coaches available, especially in the country interiors, talent identification and advice is rarely available. To address this problem, we had proposed an algorithm in [30] based on quantitative outcome of 28-tests which can help identify the talent level of a cricketer. These parametric tests have been proposed in [29] and are based on assessing physical/motor, anthropometric and cognitive abilities [20]. The next issue is to select talent optimally out of a large group of talented persons to play for a team. As noted earlier, more than a million cricketers play cricket daily in India. Short listing or selecting talented players out of such large number for even district and state levels thus becomes very challenging. Talent selection is a long procedure that requires careful planning in order to achieve the expected results [21]. Unavailability of scientific techniques makes the process highly susceptible to personal biasness and external influences.

Our literature review on the subject could not uncover any existing technique for the problem of talent selection especially in Cricket. In this paper, we propose a talent selection methodology based on OWA operator for cricket which can reduce the subjectivity, biasness and influences in talent selection to a large extent. Section 1.1 of the paper summarizes various techniques which have been used in the past on problem of selection amongst alternatives. The techniques summarized here were applied on problems from different domains including sports. In section 2.0, we discuss OWA operator and RFLQ. Section 3.0 explains the methodology with framework and algorithm for talent selection in cricket. The experiment and results are described in the section 4.0 with data.

1.1 Selection Techniques

Various techniques for selection of optimum amongst alternatives have been proposed in the past although very few of them have been used for the talent selection problem. None of them has been applied to cricket. In the following paragraphs, we have summarized the more frequently cited techniques and briefly describe their application domain:

S.N. Omkar and R. Verma [1] proposed a technique to select a cricket team using genetic algorithm based on the past performances of the national level players. The solution provided does not cover the talent aspect while selecting cricketers. Thus, its applicability is limited to the case where complete historical information about player is available. José M. Merigó and Anna M. Gil-Lafuente [2] have proposed a technique to shortlist football players based on the ordered weighted averaging distance (OWAD) operator, ordered weighted averaging adequacy coefficient (OWAAC), normalized adequacy coefficient (NAC), normalized hamming distance (NHD) and ordered weighted averaging index of maximum and minimum (OWAIMAM) [2]. However, the algorithm was proposed to work on conceptual/theoretical parameters for player selection. No details about the parameters characterizing the players were provided. The paper also did not describe weight measuring/adjustment technique.

Selection techniques have also been applied to non-sports domains. Jose M. Merigo and Montserrat Casanovas [3], proposed an algorithm to select an appropriate company to invest the money using the induced ordered weighted averaging distance (IOWAD) and also compared the algorithms with a number of alternatives. The fuzzy aggregation operators with Dempster-Shafer belief structure, probability in the ordered weighted averaging (POWA) operator [4] and induced Euclidean ordered weighted averaging distance (IEOWAD) [10] are also used to select the investment policy by the experts and also made the comparison to the fuzzy weighted averaging (FWA), fuzzy ordered weighted averaging (FOWA) etc. [14]. Immediate probability fuzzy ordered weighted averaging (IP-FOWA) [5], induced and uncertain heavy ordered weighted averaging operator (UIHOWA) [11] and fuzzy induced generalized aggregation operators (FIOWA) with multi person quasi-FIOWA [7] are used to select the strategies in risk emergent [13]. Grey relative degree is used to select the proper candidate for office by experts in fuzzy group decision making environment [6]. The used weights vectors are hypothetical not real data. Probabilistic ordered weighted distance (POWD) is a unification of probability and OWA. POWD is used in the environment of political management to make the decision on price policy and comparison also made to others [8]. To take the decision in product management, the induced 2- tuple linguistic generalized aggregation operators (2- TILGOWA) are used [9].The consensus selection model proposed by F.J. Cabrerizo, et al in the environment of unbalanced fuzzy linguistic hedge. [12]. Uncertain probabilistic ordered weighted averaging distance (UPOWAD) operator is used to select the robots [15]. Uncertain generalize ordered weighted averaging (UGOWA) is used to select the project in information technology [16] and uncertain induced ordered weighted averaging weighted aggregation (UIOWAWA) is used to select the holiday trip [17]. These summarized techniques are mapped by the ideal data which is imaginary. There is not any talent selection model in cricket find so far and no model has any live data to realize the application. In the absence of the live data, the credibility and validity is not presence in the given application of the operators. In this paper a model for cricket team selection with OWA operator is proposed from identified talented enthusiasts. So, the next section describes the OWA operator and its methodology.

2 Ordered Weighted Averaging Aggregation (OWA) Operator

The OWA operator was introduced by [18] to provide a means of aggregation, which unifies in one operator the conjunctive and disjunctive behavior. It provides a parameterized family of aggregation operators including many of the well-known operators like maximum, minimum, k-order statistics, median and arithmetic mean. For n different scores $x_1, x_2, ..., x_n$, the aggregation of these scores may be done using the OWA operator as follows.

$$\text{OWA}\ (x_1, x_2, ..., x_n) = \sum_{i=1}^{n} w_i y_i$$

where y_i is the i^{th} largest score from amongst $x_1, x_2 ..., x_n$. The weights are all non-negative $\forall i, w_i \geq 0$, and $\left(\sum_{i=1}^{n} w_i = 1 \right)$. We note that the arithmetic mean function may be obtained using the OWA operator, if $\forall i$, $w_i = \frac{1}{n}$. Similarly, the OWA operator would yield the maximum function with $w_i = 1$ and $w_i = 0$ for all $i \neq 1$. The minimum function may be obtained from the OWA operator when $w_n = 1$ and $w_i = 0$ for all $i \neq n$.

In fact, it has been shown [18] that the aggregation done by the OWA operator is always between the maximum and minimum. To find the values of the weights w_i, we need to make use of the linguistic quantifiers, explained as follows. The relative fuzzy linguistic quantifier is used to measure the weights.

2.1 Relative Fuzzy Linguistic Quantifier (RFLQ)

A relative quantifier, Q: [0, 1] → [0, 1], satisfies $Q(0) = 0$, $\exists r \in [0,1]$ such that $Q(r) = 1$. In addition, it is non-decreasing if it has the following property $\forall a, b \in [0,1]$, if $a > b$, then $Q(a) \geq Q(b)$. The membership function [18] of a relative quantifier can be represented as shown in [i]:

$$Q(r) = \begin{cases} 0 & if\ (r < a) \\ \dfrac{r-a}{b-a}; & if\ (a <= r <= b, \\ 1 & if r > b \end{cases} \quad \begin{array}{l} if\ \ r < a \\ if\ \ a \leq r \leq b\,, \\ if\ \ r > b \end{array} \tag{1}$$

where, $a, b, r \in [0,1]$ and $Q(r) = Q(i/m)$

In [19], computation of the weights w_i of the OWA aggregation from the function Q describing the quantifier has been demonstrated. In the case of relative quantifier, with m criteria [19],

$$w_i = Q(i/m) - Q((i-1)/m)), i = 1,2,....,m, with\ Q(0) = 0.$$

3 Cricket Talent Selection Methodology

Talent selection is a technique that helps in short listing/selecting most talented m-players out of n-available alternatives. To accomplish this work, we record the normalized outcome value of all n-players against the 28-tests [30] specific to talent identification in cricket. The normalized data is depicted in equation-

$$test\ outcome\ (ti)\ \in [a, b] \forall i\ \text{I}\ 1..28$$

To obtain the weights for aggregation of talent values of various players, we apply relative fuzzy linguistic quantifier (RFLQ) as described in algorithm proposed in [30]. The resultant weights are depicted in equation-3.

Now, the OWA operator is used to aggregate the values of all test outcomes of each player. The algorithm for carrying out this work has been described section 3.1. The aggregated value of each player's talent is shown in equation - 4. Next we sort the talent values in descending order to rank the talent levels of players. Now top m players may be easily picked out of n in order of their talent. The complete process is outlined in the figure-1.

$$S=\begin{array}{c} \\ p_1 \\ p_2 \\ . \\ . \\ . \\ p_n \end{array}\begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & . & . & . & t_{28} \end{array} \\ \begin{bmatrix} v_{p_1t_1} & v_{p_1t_2} & . & . & . & v_{p_1t_{28}} \\ v_{p_2t_1} & v_{p_2t_2} & . & . & . & v_{p_2t_{28}} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ v_{p_nt_1} & v_{p_nt_2} & . & . & . & v_{p_nt_{28}} \end{bmatrix} \end{array} \tag{2}$$

$$W=\begin{array}{c} \\ p_1 \\ p_2 \\ . \\ . \\ . \\ p_n \end{array}\begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & . & . & . & t_{28} \end{array} \\ \begin{bmatrix} w_{p_1t_1} & w_{p_1t_2} & . & . & . & w_{p_1t_{28}} \\ w_{p_2t_1} & w_{p_2t_2} & . & . & . & w_{p_2t_{28}} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ w_{p_nt_1} & w_{p_nt_2} & . & . & . & w_{p_nt_{28}} \end{bmatrix} \end{array} \tag{3}$$

Aggregation of the all tests' values with the help of OWA operator.

$$\text{OWA}\,(t_1,t_2,\cdots\cdots,t_{28})=\sum_{i=1}^{n} w_i t_i\ .$$

$$V=P_1(t_1,t_2,\cdots\cdots,t_{28})=$$

$$[w_1,w_2,....,w_n].[Desending\_Sort/Ascending\_Sort(t_1,t_2,.....,t_i)]$$

$$V=\begin{array}{c} \\ p_1 \\ p_2 \\ . \\ . \\ . \\ p_n \end{array}\begin{array}{c} \\ \begin{bmatrix} p_1v_1 \\ p_2v_2 \\ . \\ . \\ . \\ p_nv_n \end{bmatrix} \end{array} \tag{4}$$

Sort the V and give rank to the players for selecting the talented player

Algorithm:

```
begin
  for each player p_i, i ∈ [1,n] where n = total no. of players
    for each test T(k); k ∈ [0,28]
      r(k) = k/28;
        for each test T(k); k ∈ [0,28]
          begin
      if r(k) < a
                QR(k) = 0
         else if {a ≤ r(k) ≤ b}
                              QR(k) = (r(k) − a)/(b − a)
       else
                                   r(k) > b}
                                        QR(k) = 1
          end;
                  if (k <> 0)
                   w[k,i] = QR(k) − QR(k − 1);
    End Loop k;
End Loop i;

for each player p_i, i ∈ [1,n] where n = total no. of players
          for each test T(k); k ∈ [0,28]
      if a < b then
         y(i) = arrange_and_assign(Π_{t_k}(T[k]), ascending);
      else
         y(i) = arrange_and_assign(Π_{t_k}(T[k]), descending);
    end for k;
    sum: = 0;
    for each player p_i, i ∈ [1,n] where n = total no. of play-
ers
      sum += w(i,28)*y(i);
    OWA( k,i) = sum;
End for i;
End;
Sort descending on ascending value vector (V) = OWA(k,i);
End for i;
Cricket team= Pick up the m-players from the OWA(k,i).
```

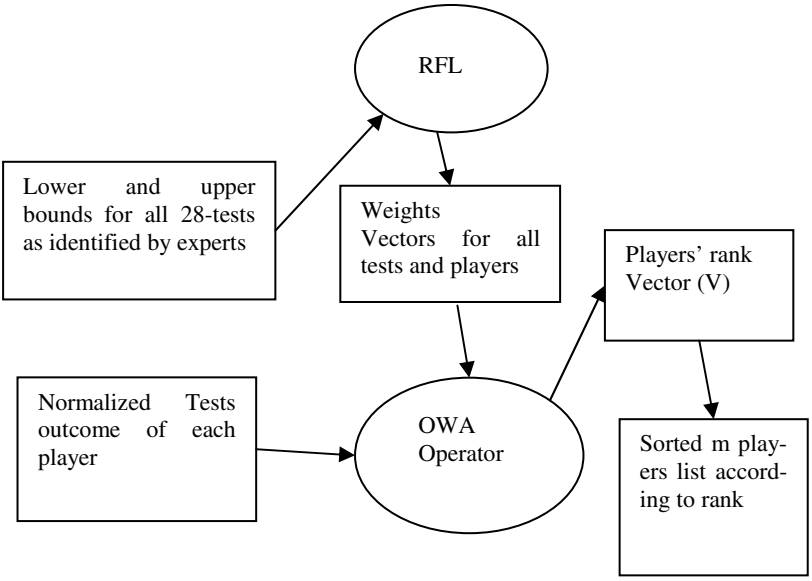


Fig. 1. Talent selection process framework

4 Experiment and Results

To experiment with the algorithm's applicability, we collected the data of the present 15-players of the JMI University Cricket Team against the tests (T_1, T_2, … T_{28}) and normalized the results on the scale of 0 to 1. The names of the players were randomly

Table 1. Player's Normalized Data Base

Tests	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28
Player1	0.48	0.54	0.28	0.32	0.54	0.72	0.16	0.09	0.6	0.32	0.66	0.74	0.65	0.56	0.69	0.72	0.42	0.65	0.58	0.68	0.6	0.8	0.7	0.75	0.35	0.6	0.8	0.85
Player2	0.72	0.54	0.71	0.16	0.81	0.16	0.23	0.57	0.43	0.37	0.49	0.28	0.09	0.17	0.61	0.8	0.76	0.68	0.59	0.27	0.23	0.19	0.34	0.42	0.48	0.69	0.54	0.39
Player3	0.28	0.76	0.69	0.53	0.44	0.36	0.24	0.77	0.61	0.52	0.48	0.26	0.11	0.62	0.77	0.91	0.89	0.17	0.27	0.38	0.44	0.79	0.83	0.73	0.65	0.31	0.29	0.84
Player4	0.52	0.69	0.88	0.23	0.47	0.57	0.92	0.88	0.29	0.25	0.45	0.38	0.97	0.86	0.1	0.92	0.16	0.43	0.69	0.74	0.44	0.38	0.69	0.88	0.42	0.73	0.75	0.25
Player5	0.12	0.39	0.69	0.26	0.66	0.49	0.37	0.64	0.88	0.83	0.77	0.68	0.44	0.27	0.36	0.39	0.49	0.56	0.16	0.55	0.61	0.79	0.41	0.5	0.73	0.61	0.63	0.78
Player6	0.18	0.49	0.15	0.38	0.64	0.65	0.71	0.9	0.48	0.86	0.76	0.63	0.29	0.31	0.55	0.56	0.61	0.52	0.36	0.68	0.66	0.49	0.52	0.69	0.3	0.45	0.68	0.66
Player7	0.5	0.95	0.85	0.87	0.69	0.29	0.36	0.66	0.79	0.77	0.71	0.63	0.66	0.41	0.58	0.41	0.35	0.25	0.23	0.21	0.44	0.68	0.44	0.49	0.56	0.61	0.63	0.8
Player8	0.15	0.29	0.38	0.91	0.83	0.86	0.54	0.55	0.68	0.74	0.44	0.38	0.49	0.26	0.19	0.23	0.22	0.38	0.44	0.61	0.59	0.58	0.41	0.36	0.29	0.68	0.71	0.81
Player9	0.48	0.53	0.21	0.14	0.89	0.9	0.68	0.41	0.28	0.65	0.71	0.88	0.81	0.69	0.44	0.35	0.25	0.23	0.19	0.56	0.58	0.68	0.71	0.45	0.49	0.63	0.59	0.55
Player10	0.61	0.59	0.43	0.77	0.71	0.7	0.68	0.51	0.43	0.44	0.62	0.52	0.32	0.22	0.12	0.89	0.18	0.28	0.63	0.73	0.85	0.81	0.79	0.68	0.59	0.22	0.35	0.43
Player11	0.88	0.18	0.21	0.29	0.81	0.72	0.59	0.63	0.33	0.38	0.61	0.5	0.53	0.44	0.38	0.74	0.64	0.81	0.63	0.49	0.38	0.35	0.65	0.88	0.71	0.74	0.25	0.29
Player12	0.33	0.69	0.95	0.13	0.19	0.69	0.75	0.37	0.45	0.53	0.43	0.55	0.65	0.75	0.29	0.39	0.48	0.29	0.61	0.4	0.5	0.81	0.84	0.73	0.71	0.77	0.69	0.57
Player13	0.11	0.64	0.18	0.19	0.81	0.71	0.38	0.63	0.51	0.83	0.87	0.81	0.29	0.39	0.61	0.59	0.71	0.49	0.43	0.58	0.63	0.78	0.77	0.81	0.63	0.66	0.28	0.15
Player14	0.35	0.45	0.75	0.55	0.85	0.65	0.25	0.46	0.67	0.76	0.95	0.12	0.61	0.44	0.65	0.51	0.39	0.43	0.63	0.88	0.61	0.25	0.65	0.59	0.44	0.39	0.53	0.61
Player15	0.18	0.98	0.7	0.8	0.6	0.65	0.5	0.61	0.49	0.51	0.53	0.59	0.65	0.38	0.71	0.73	0.71	0.65	0.44	0.59	0.53	0.63	0.69	0.71	0.88	0.71	0.95	0.91

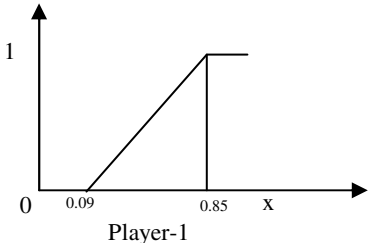


Fig. 2. Triangular fuzzy value with range for player1 that is p_1

Table 2. Weights database of players

Weights	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25	T26	T27	T28
Player1	0	0	0.026	0.084	0.007	0.046	0.047	0.048	0.046	0.047	0.048	0.047	0.046	0.047	0.048	0.046	0.047	0.048	0.047	0.045	0.048	0.048	0.046	0.038	0	0	0	0
Player2	0	0	0.028	0.046	0.05	0.048	0.05	0.05	0.049	0.05	0.05	0.05	0.048	0.05	0.05	0.05	0.05	0.05	0.049	0.05	0.05	0.033	0	0	0	0	0	0
Player3	0	0	0	0.04	0.082	0.044	0.045	0.045	0.044	0.045	0.045	0.044	0.045	0.045	0.043	0.045	0.052	0.038	0.044	0.045	0.045	0.044	0.045	0.045	0.021	0	0	0
Player4	0	0	0.012	0.04	0.044	0.043	0.044	0.044	0.043	0.043	0.044	0.043	0.043	0.044	0.044	0.042	0.044	0.044	0.044	0.043	0.044	0.044	0.042	0.044	0.044	0.033	0	0
Player5	0	0	0	0.03	0.048	0.046	0.047	0.047	0.046	0.048	0.047	0.048	0.044	0.047	0.047	0.044	0.048	0.047	0.048	0.046	0.047	0.047	0.046	0.048	0.03	0	0	0
Player6	0	0	0	0	0.039	0.046	0.048	0.048	0.047	0.048	0.048	0.048	0.047	0.048	0.048	0.046	0.048	0.048	0.048	0.047	0.048	0.048	0.047	0.048	0.048	0	0	0
Player7	0	0	0	0	0	0.005	0.049	0.049	0.047	0.049	0.048	0.049	0.047	0.049	0.049	0.047	0.048	0.049	0.049	0.049	0.049	0.048	0.048	0.048	0.049	0.049	0.028	0
Player8	0	0	0	0	0.038	0.046	0.048	0.047	0.046	0.047	0.048	0.047	0.046	0.047	0.047	0.046	0.047	0.048	0.047	0.046	0.048	0.047	0.046	0.047	0.047	0.023	0	0
Player9	0	0	0	0.004	0.047	0.046	0.048	0.047	0.046	0.048	0.047	0.047	0.046	0.048	0.047	0.046	0.047	0.048	0.047	0.046	0.046	0.047	0.046	0.047	0.048	0.009	0	0
Player10	0	0	0	0.03	0.047	0.045	0.047	0.047	0.045	0.047	0.047	0.046	0.046	0.046	0.047	0.046	0.046	0.047	0.047	0.045	0.047	0.047	0.045	0.047	0.043	0	0	0
Player11	0	0	0	0	0	0.049	0.051	0.051	0.05	0.052	0.049	0.052	0.05	0.051	0.052	0.05	0.051	0.051	0.052	0.05	0.051	0.051	0.051	0.051	0.033	0	0	0
Player12	0	0	0	0.016	0.044	0.042	0.044	0.044	0.043	0.043	0.044	0.044	0.042	0.044	0.044	0.043	0.044	0.044	0.044	0.042	0.044	0.044	0.043	0.043	0.044	0.044	0.026	0
Player13	0	0	0	0.043	0.048	0.046	0.047	0.047	0.047	0.047	0.047	0.048	0.046	0.046	0.048	0.046	0.047	0.047	0.048	0.046	0.047	0.047	0.047	0.047	0.017	0	0	0
Player14	0	0	0	0.03	0.047	0.047	0.047	0.047	0.046	0.046	0.047	0.048	0.046	0.047	0.047	0.046	0.048	0.047	0.048	0.046	0.047	0.047	0.046	0.048	0.03	0	0	0
Player15	0	0	0	0	0	0.045	0.043	0.045	0.043	0.045	0.045	0.045	0.044	0.045	0.045	0.044	0.045	0.045	0.045	0.044	0.045	0.045	0.043	0.045	0.045	0.045	0.044	0.02

Table 3. Weights Calculation for Player1 with Lower limit (0.09) and Upper limit (0.85)

i= 0,1,2,.....28	0	1	2	*3*	*4*	...	*28*
i/28 = r	0/28= 0	*1/28* =0.086	2/28 =0.071	*3/28* *=0.11*	*4/28=0.143*	...	*28/28* =1
$Q(i/n) = Q(r)$ $Q(r) = \begin{cases} 0 & if\ r < a \\ \frac{r-a}{b-a} & if\ a \le r \le b \\ 1 & if\ r > b \end{cases}$	0	0	0	*0.026*	*0.11*	...	1
$w_i = Q(\frac{i}{n}) - Q(\frac{i-1}{n})$	Not Required	0-0= **0**	0-0=**0**	*0.026-0* *=0.026*	*0.11-0.026* *=0.084*	...	1-1 =**0.000**

mapped/coded to P_1, P_2, …P_{15}. The coded names were used in further process of execution of algorithm. The normalized players' data is depicted in table-1. The weights based on the data in table-1 are calculated and are depicted in table-2. The application of the proposed algorithm for player-1 is shown in table-3 and the fuzzy triangular norm for player-1 is shown in figure-2.

Now, the OWA operator is used to aggregate the data for player1.

P1V1= (0.048, 0.54, 0.28, 0.32, . . . 0.85)=

[0,0,0.026,0.084....,0] *[0.85,0.80,0.80,0.75,.....,0.09)]

= 0.60525

Similarly, the aggregated values of other players can be aggregated.

The aggregated value vector V = [0.60525, 0.42274, 0.54348, 0.55289,0.54490, 0.53216, 0.51198, 0.45045, 0.51829, 0.54393, 0.51636, 0.53247, 0.57898, 0.54948, 0.59644].

Applying the sorting algorithm on aggregated value vector V renders the players to fall in descending sequencing of their talent ranks. This is shown below:

P1(0.60525)>P15(0.59644)>P13(0.57898)>P4(0.55289)>P14(0.54948)>P5(0.54490)
>P10(0.54393)>
P3(0.54348)>P12(0.53247)>P6(0.53216)>P9(0.51829)>P11(0.51636)>P7(0.51198)>
P8(0.45045)>P2(0.42274).

Thus, the top m-players may now be easily selected/ shortlisted.

4.1 Validation

To validate the algorithm we first decoded the dummy viz. P_1, P_2, … P_{15} given earlier with the real names. We also recorded the unbiased opinion of the Jamia's Cricket coach about his preferred team selection out of 15-available players. The coach's opinion was compared with the selection rank of the top 11-players as determined by our algorithm. The comparison shows that the 9 out 11 players selected by the algorithm were also shortlisted by the coach in his team. Thus, the algorithm's result matches with the coach opinion with a high degree of 81.8% accuracy.

5 Conclusion

The literature review shows that no talent selection model has been proposed for cricket. In this paper we presented a summary of various selection techniques, which have been proposed by different authors for different domains. Afterwards, a brief discussion on OWA operator is presented. Subsequently, we have proposed an algorithm for selecting top m-talented players out of n-available alternatives using OWA operator and RFLQ technique. The algorithm's application has been demonstrated for short listing of players for the University's cricket team. The result was validated by comparing the players' ranks generated by the algorithm with the coach's opinion.

References

1. Omkar, S.N., Verma, R.: Cricket Team selection using Genetic Algorithm. In: International Congress on Sports Dynamics (ICSD 2003), Melbourne, Australia, September 1-3 (2003)
2. Merigó, J.M., Gil-Lafuente, A.M.: Decision making in sport management based on OWA operator. International Journal of Expert System with Application 38, 10408–10413 (2011)
3. Merigo, J.M., Casanovas, M.: Decision-making with distance measures and induced aggregation operators. International Journal of Computers & Industrial Engineering 60, 66–76 (2011)
4. Casanovas, M., Merigo, J.M.: Fuzzy aggregation operators in decision making with Dempster-Shafer. International Journal of Expert System with Application 39, 7138–7149 (2012)
5. Merigo, J.M.: Fuzzy decision making with immediate probabilities. International Journal of Computers & Industrial Engineering 58, 651–657 (2010)
6. Rao, C., Peng, J.: Fuzzy Group Decision Making Model Based On Credibility Theory and Gray Relative Degree. International Journal of Information Technology & Decision Making 8(3), 515–527 (2009)
7. Merigo, J.M., Gil-Lafuente, A.M.: Fuzzy induced generalized aggregation operators and its application in multi-person decision making. International Journal of Expert System with Application 38, 9761–9772 (2011)
8. Merigo, J.M., et al.: Group decision making with distance measures and probabilistic information. International Journal of Knowledge-Based system 40, 81–87 (2013)
9. Merigo, J.M., Gil-Lafuente, A.M.: Induced 2-tuple linguistic generalized aggregation operators and their application in decision-making. International Journal of Information Science (2013), http://dx.doi.org/10.1016/j.ins.2013.02.039
10. Merigo, J.M., Casanovas, M.: Induced aggregation operators in the Euclidean distance and its application in financial decision making. International Journal of Expert System with Application 38, 7603–7608 (2011)
11. Merigo, J.M., Casanovas, M.: Induced and uncertain heavy OWA operators. International Journal of Computers & Industrial Engineering 60, 106–116 (2011)
12. Cabrerizo, F.J., et al.: A Consensus Model for Group Decision Making Problems with Unbalanced Fuzzy Linguistic Information. International Journal of Information Technology & Decision Making 8(1), 109–131 (2009)
13. Merigo, J.M., Gil-Lafuente, A.M.: New decision-making techniques and their application in the selection of financial product. International Journal of Information Science 180, 2085–2094 (2010)
14. Merigo, J.M.: Probabilities in the OWA operator. International Journal of Expert System with Application 39, 11456–11467 (2012)
15. Zeng, S., et al.: The uncertain probabilistic OWA distance operator and its application in group decision making. International Journal Of Applied Mathematical Modeling 37, 6266–6275 (2013)
16. Zhou, L.-G., et al.: Uncertain generalized aggregation operators. International Journal of Expert System with Application 39, 1105–1117 (2012)
17. Merigo, J.M., et al.: Uncertain induced aggregation operators and its application in tourism management. International Journal of Expert System with Application 39, 869–880 (2012)
18. Lager, R.R.: On ordered weighted averaging aggregation operators in multimedia decision making. IEEE Trans. Systems Man Cyber Net. 18, 183–190 (1988)

19. Chang, S.-L., et al.: Applying fuzzy linguistic quantifier to select supply chain partners at different phases of product life cycle. International Journal of Production Economics 100(2), 348–359 (2006)
20. Ahamad, G., Naqvi, S.K., Sufyan Beg, M.M.: A Studyof Talent Identification Models in Sportsand Parameters for Talent Identification in Cricket. In: International Conference on Physical Education and Sports Science, Department of Physical Health Sports Education, Aligarh, India, Ref.No.ICPESS/OP/160, November 16-18 (2012)
21. Loko, J.: Talent Selection Procedure, http://www.hurdlecentral.com/Docs/Coachig/LokoTalentselectionProcedure.pdf
22. Elferink-Gemser, M.T., Visscher, C., Lemmink, K., et al.: Relation between multidimensional performance characteristics and level of performance in talented youth field hockey players. Journal of Sports Sciences 22(11-12), 1053–1063 (2004)
23. Vrljic, K., Mallett Clifford, J.: Coaching knowledge in identifying football talent. International Journal of Coaching Science (2008)
24. Karasilshchikov, O.: Talent Recognition and Development- Elaborating on a Principle Model. International Journal of Developmental Sport Management 1(1), 16150 (2011)
25. Vaeyens, R., Malina, R.M., Janssens, M.: A Multidisciplinary Selection Model for Youth Soccer: The Ghent Youth Soccer Project. British Journal of Sports Medicine 40, 928–934 (2006)
26. Peltola, 1992. Williams & Reilly, 2000, Talent Identification in British Judo, p. 216, http://www.bath.ac.uk/sports/judoresearch/Full%20texts/Talent%20Identification%20in%20British%20%20Judo.pdf
27. Pearson, D.T., Naughton, G.A., Torode, M.: Predictability of physiological testing and the role of maturation in talent identification for adolescent team sports. Journal of Science and Medical in Sport 9(4), 277–287 (2006)
28. Mallillin, T.R., Josephine Joy Reyes, B., et al.: Sports Talent Identification in 1st and 2nd year UST High School Students. Philippine Journal of Allied Health Sciences 2(1), 41–42 (2007)
29. Talent Identification Report Explanatory Notes, http://static.ecb.co.uk/files/talent-id-report-explanatory-notes-1403.doc
30. Ahamad, G., Naqvi, S.K., Sufyan Beg, M.M.: A Model for Talent Identification in Cricket Based on OWA Operator. International Journal of Information Technology and Management Information System 4(2), 40–55 (2013) ISSN Print: 0976-6405, ISSN Online 0976-6413
31. India the new world provider for cricket, http://www.theage.com.au/sport/cricket/india-the-new-world-provider-for-cricket-20130222-2ex1x.html

Knowledge Representation in ISpace Based Man-Machine Communication

Annamária R. Várkonyi-Kóczy[1,2], Balázs Tusor[1], and Imre J. Rudas[3]

[1] Institute of Mechatronics and Vehicle Engineering
Óbuda University
Budapest, Hungary
varkonyi-koczy@uni-obuda.hu, balazs.tusor@gmail.com
[2] Institute of Mathematics and Informatics
Selye János University
Komarno, Slovakia
[3] Institute of Applied Mathematics
Óbuda University
Budapest, Hungary
rudas@uni-obuda.hu

Abstract. With the spreading of intelligent machines, man-machine communication has become an important research area. Today, intelligent robots co-operating with humans usually have to be able to store, retrieve, and update information about their environment, interpret and execute commands, offer existing and gain/learn new services. In these processes, the efficient knowledge representation and storage are of key importance. In this paper, a new graph based modular knowledge storage and representation form is presented which is able to handle inaccurate and ambiguous information, to store, retrieve, modify, and extend theoretical and practical knowledge, to interpret commands, and to learn new cognitions.

1 Introduction

Intelligent robots collaborating with humans usually have to be able to store, retrieve, and update information about their environment, interpret and execute commands, offer existing and gain/learn new services [1]. The man-machine co-operation usually involves some kind of communication, as well [2]. In these processes, the efficient knowledge representation and storage are of key importance. The size of the databases, the accessibility to the stored knowledge, the possibilities of building in new or refining the possessed information together with the flexibility of the information update have a direct effect on the speed and effectiveness of the cooperation asking for efficient data and knowledge structures.

There can be found different knowledge representation approaches in the literature which can advantageously be applied in man-machine co-operation, as well. We have to mention conceptual graphs (CG) (see [3]) which has primarily

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_23,

been developed for data base interfaces, to make easier for humans to understand the data and to make inquiries. This approach applies the concepts as basic primitives and the concepts themselves are types which incorporate every instance that shares that type. In [4] a concept graph based knowledge model is proposed to represent concepts, terminologies, methods, and processes in Software Architecture. The knowledge is classified into a hierarchy of 4 levels: fundamental concepts, domain knowledge, process knowledge, and task knowledge. [5] proposes a graph-based knowledge representation for Geographic Information System (GIS), to represent spatial and non-spatial data (nodes), also including spatial relationships (edges) between spatial objects and use the model for generating a dataset composed of both types of data. The authors of [6] propose the Feature Event Dependency Graph (FEDG). It focuses on representing the fact level knowledge compressively however without losing any important information. The FEDG is efficient in retrieving user concerned knowledge patterns and is especially useful in discovering latent knowledge and in effective reasoning. In FEDG, the knowledge is represented by feature events (nodes) and the weighted context links and dependency links (directed edges) between them.

In this paper, a new graph based knowledge storage and representation form is introduced. Our approach principally belongs to the wider family of conceptual graphs and can be considered as a more specialized version of CG however with differences, thus becoming a novel approach: First of all, it has specifically been developed for control systems, like the iSpace frame presented in [7] (for more information about ISpace, see also [8]); while in CG the concepts are themselves types, in our structure the instances are modeled as separate nodes in the instance domain; we do not define constrains like CG does, etc. However, possibly the main difference between our approach and the conceptual graph approach is that instead of relation nodes we apply specific edge types.

The knowledge representation graph proposed in this paper distinguishes between theoretical knowledge and linkage-possibilities among virtual tools and their real-word embodiments. This disassociation results in that a sophisticated modularity can be kept in the knowledge storage and the environmental changes have less effect on the knowledge structures. This way, the redundancy of the representation becomes also lower than that of the traditional structures.

The paper is organized as follows: In Section 2 knowledge representation is addressed. Section 3 deals with knowledge harmonized command interpretation and processing while Section 4 is devoted to knowledge based hypotheses building. Finally, Section 5 contains the conclusions of the paper.

2 Knowledge Representation

2.1 Knowledge Base

The knowledge base holds the knowledge of the system. It is realized as a graph-based structure, divided to abstract and instance domains. In the abstract domain, the nodes denote the known abstract objects and concepts and the directed edges between them describe their relationship, thus describing the knowledge

of the system about the world in general. In the instance domain the nodes represent existing objects that the intelligent machine (robot, iSpace) knows about. The nodes are homogeneous and nameless (for an example, see Fig. 1). Whenever the "world" of the robot is altered in any way, the changes should appear in the appropriate domains of the knowledge base as well. "Physical" changes (addition or removal of sensors, detectors, agents, etc.) should appear in the instance domain, while "property" changes (temperature of the room, etc.) should appear in the abstract domain.

Fig. 2 shows structural examples for the realization of the knowledge base (the examples are taken from the ISpace application published in [9]). Fig. 2 depicts that tea, coffee, and cappuccino are drinks, beverages are synonymous to drinks, drink machines can make drinks, brewing is (in this case) synonymous to making, and actions done to drinks are heuristically dependent on time and mood. It also shows an example how we can define minimum and maximum values for variables together with the used step size (granularity). The different types of edges used in the proposed knowledge base are (see Figs. 1- 3)

- Ability "edge": It is a three-way connection, where node A is connected to nodes B and C, if A can do C to B. E.g., *"drink machines can make drinks"*, where A is node *" drink machine"*, B is node *"make"* and C is node *"drink"*.
- Instance edges: An instance edge (denoted by an arrow with label I) assigns the address of an executive agent to a node, e.g., "*window opener*". The assigned address is the physical address of the connected window opener device.
- Meta edges: node A is connected to node B, if the concept A has a numerical value in B quality. E.g., "*the value of the day is 6*", where A is node *"value"*,

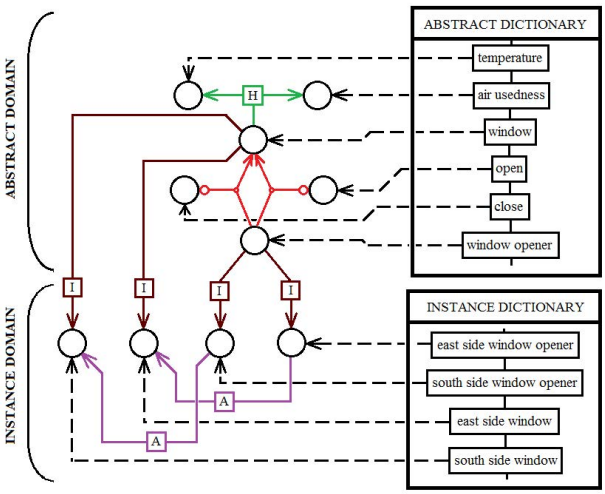


Fig. 1. Structure of the knowledge representation: the abstract and instant domains

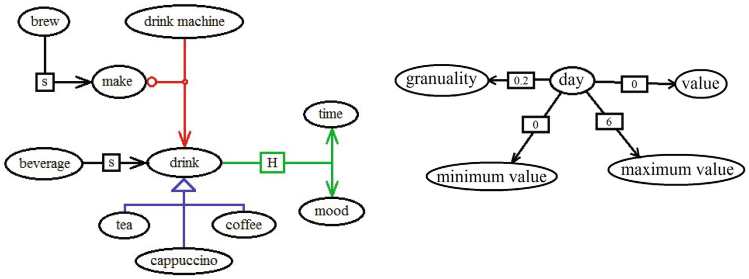


Fig. 2. Example for the abstract domain of the graph-based knowledge base (left) and structure of the meta edges (right)

B is node *"day"* (i.e., it is the 6th day of the week). With the application of meta edges *environmental* variables can be appointed, which can store the values of sensors or any other numerical values. The range of the *value* can also be described by creating and setting the values of the meta edges thus connecting nodes *"minimum value"* and *"maximum value"* to the appropriate node. In this figure the defined *environmental variable* is node *"day"*, its properties are set through the meta edges connecting it to nodes *"minimum value"* and *"maximum value"*, storing the possible minimum and maximum values. The edge between nodes *"value"* and *"day"* stores the current value of the parameter, while the edge between nodes *"granularity"* and *"day"* stores a value that remarks how sensitive the given variable is to changes.

- Inheritance edges: An inheritance edge (denoted by a triangle headed arrow) connects node A to node B, if A (e.g "*coffee*" belongs to type B (e.g. "*drink*"). I.e., "*coffee is a drink*". A inherits the properties of B.
- Heuristics edges (denoted by an arrow with label H): Node A is connected to node B, if B (usually an environmental variable) can be bound to concept A. E.g., actions done to "*coffee*" (A) are dependent of "*time*" and "*mood*" (B).
- Synonym edges: Node A is connected to node B by a synonym edge (denoted by a symmetrical connection with label S) if the two concepts are synonyms, like in our case "*drink*" and "*beverage*". By this, the robot can interpret instructions more flexibly.
- Association edges: Association edges (denoted by an arrow with label A) build a connection between two nodes (with dedicated relationship) of the instance domain. E.g., the instance of a particular *"window opener"* is associated to a certain instance of *"window"*.
- Fuzzy edges: Fuzzy edges are similar to meta edges, except that they handle fuzzy values instead of numerical values. E.g., A = "*Saturday*", B = "*day*" and the assigned value is given by the membership function that can be seen in Fig. 3, for input value x. ($x = 0...6$, assuming the days of the week start with *Sunday* (with index 0).

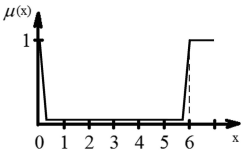


Fig. 3. Fuzzy membership function for the day "Saturday"

In Fig. 1 an example is shown how we can represent our knowledge about the existing windows and their openers. A window can be opened and closed by its opener. The actions depend heuristically on the temperature of the room and how used up is the air in the room. Let us consider that there are two windows and window opener devices in the room (denoted by the side of the room where they are located: east or south), and the openers are associated with the appropriate windows they can do actions to, thus it is trivial which device can operate on which window. The user can instruct the robot to open or close one or both windows. Each node is identified by the dictionaries.

To be able to represent instances of objects which do not exist at the moment but can be made by devices operated by the robot (e.g., *coffee*) the *void* nodes are introduced. A *void* node is the instance of the concept of the product. There is a void node for each product and they are associated with the instance (which is the drink machine, in the case of coffee) that can create the product.

2.2 Hypothesis Storage

The robot is able to build hypotheses about the environment and the habits of its user. The hypotheses are stored in the *hypothesis storage*. A hypothesis consists of (pointers to) the action and object nodes. It can also have optional numerical values, similarly to the command the hypothesis is based on. A hypothesis also has at least one *trigger*, which consists of a *justification* value that denotes how reassured the system is in the *trigger* of the hypothesis; and at least one *condition*. A *condition* is derived from an environmental variable: it consists of a condition node (the node of the environmental variable), value (the value stored in the meta edge connecting the environmental variable to node "value"), *sensitivity* (which can be derived similarly to value using node "granularity" and the appropriate meta edge), and *affirmativeness*, which is a Boolean value. If its value is false then the condition is used as if it would be negated. (Thus the trigger will be triggered only if the condition is not satisfied.)

2.3 Dictionaries

For identification, dictionaries are used (each consisting both abstract and the instance domains), which assign analogous words to the nodes. Here, as example, only one (English) dictionary is used, although the concept is designed for the usage of multiple dictionaries. Further, non-verbatim dictionaries (e.g. based

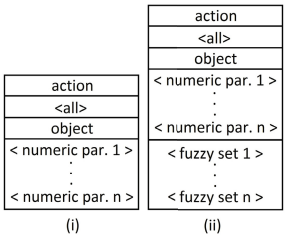


Fig. 4. Structure of the (i) instruction and (ii) prohibition type commands

on hand signs, see [10]) can also be included in the system. In general, the dictionaries are "translated" to an equivalent, non-language-dependent dictionary during the preprocessing. This way the knowledge base can be kept independent of natural languages.

The most basic form of dictionaries is an ordered list containing the words (and expressions) and the reference to their corresponding nodes. The construction of the dictionaries and the knowledge base happens simultaneously; whenever a new node is created, a new entry is also added to the dictionaries if the concept of the node is described by a word or expression that is not included in the dictionary yet.

3 Command Interpretation and Command Processing

In order to achieve advanced command processing, the system has to use grammar rules. These contain rules that determine how the words and sentences can be built with regards to the language. The modules of the *Command Processor* analyze the given command (with regards to its language and grammar) and instruct the appropriate executive agents to carry it out.

3.1 Command Parsing

The *Command Parsing Module CPM* first determines the type of the command, and then parses it. Latter step depends on the type of the command. The input of the *CPM* is the pre-processed command and its output is the parsed command. Two types of commands can be defined: instructions and prohibitions (Fig. 4).

Instructions are simple commands. They are given by the human user to achieve change in the environment or by the robot itself due to the Autonomous Action Planning, which is the result of learning. The first part of instructions describes the action that is needed to be executed, possibly followed by an optional "all" word, which means that the action is needed to be executed to all available objects (e.g., "*open all windows*"). The second part describes the object of the action which can be followed by optional numerical values.

Prohibitions are commands that are given by the human user to alter the behavior of the system by bounding one or more commands that were already

learned, thus achieving change in the hypotheses. Their structure is basically the same as the structure of instructions except that they start with "DO NOT" and have additional (at least one) text parameters, which usually represent fuzzy variables. E.g., in case of the prohibition command "DO NOT make coffee on Saturday"; "Saturday" is defined by the fuzzy membership function shown in Fig. 3. Since the granularity of Saturday is 0.2, early Sunday morning and late Friday night count as "almost Saturday" (this is why a fuzzy set is used for the definition of the day instead of a crisp function).

This offers an efficient way to handle situations, where certain instructions are regularly given except some particular cases. And further, the definition of the occasions can hardly be defined sharply. The example presented here is the alarm setting. The person asks the robot to set the alarm to 7:00 a.m. every day. The forbidding command is formulated by the human after he/she is awakened at 7:00 on Saturday morning, as well. As "Saturday" is defined as a fuzzy notion, the robot will not set the alarm to 7:00 on the following late Friday evenings, Saturdays and early Sunday mornings. Parallel with this, the system can learn another alarming hypothesis, like "set the alarm to 9:00 on Saturdays". The latter will be handled as different command, because of the differing numeric parameters.

Since the command is in a pre-defined format (which is based on the strictly defined word order of the English language), the algorithm of the parsing phase is quite trivial. To mention an example, in case of instructions the first word is always the action and the second one is either the "all" word or (after the removal of the occurring articles, like "the") a noun that gives the object of the command.

Fig. 5 shows an illustration for the advanced command analysis. The system first parses the sentence to separate words, then analyses each part. From that, it produces a graph, where the bigger circles are references to the appropriate concepts of the knowledge base, while the smaller ones connected to them are features of those words (e.g. a pronoun belonging to the noun, is it plural or singular, etc.). Lastly, the squares denote the words-part of speech. This way the commands that the human users can give become more flexible, the users do not need to stick to a rigid order of words if it is unnatural in their native language.

3.2 Command Interpretation

The function of the *Command Interpretation Module* (*CIM*) is to determine which executive agent is able to carry out a given command. Its input is the pre-parsed command and its output is the address or reference of an executive agent. CIM only processes instruction commands since prohibition type commands are not needed to be executed. The algorithm searches for three key nodes, in order: the *object node*, the *action node* and the *executive node*.

These three nodes are needed to be connected via an ability edge (in such way that represents the following: the concept of the *executive node* can do the concept of the *action node* to the concept of the *object node*).

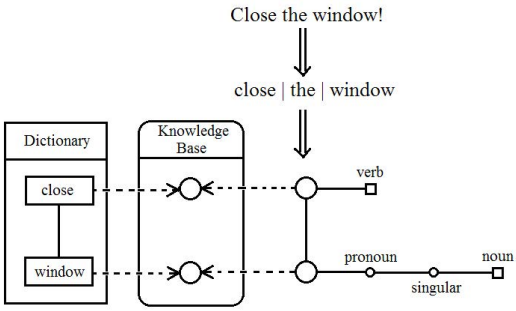


Fig. 5. Example for the command parsing

The *object node* is a node that either corresponds to the object of the command (through the dictionary) or can be reached from the corresponding node through inheritance and/or synonym edges with a constraint that the node is needed to have at least one instance (a node in the instance domain, which is bound to the *object node* with an instance edge).

The *action node* is a node that either is corresponding to the action of the command, or can be reached from the corresponding node through inheritance and/or synonym edges, similarly to the *object node*, with the difference that the action node does not need to have any instance. The *executive node* is a node that can be found the same way as the object nodes, with the difference that its instance is needed to be associated with the instance of the *object node*. Thus, the algorithm effectively does the following: First, it finds out what concept the object of the command is, then what action is needed to be done to the object, and finally, what concept can do that action to the object. If the algorithm cannot locate the three previously defined nodes, then it stops: the instruction cannot be carried out. If it finds three nodes that satisfy all of the constrains defined above, it returns the address or reference of the executive agent that is stored in the instance node of the *executive node*.

Let us see an illustrative example: The user gives the command: "Brew coffee!" (see Fig. 2). The algorithm first identifies node "drink" as the object node, since it is the ancestor of the node "coffee". Then it identifies node "make" as the action node, since it is the synonym of the node "brew". Finally, it identifies node "drink machine" as executive node. If it has an instance (though the instances are not shown in the figure) that is associated to the instance of "drink" (which is a void node), then the system returns the address/reference of the drink machine agent stored in the instance node.

3.3 Instruction and Execution

The task of the *Instructor Module* (*IM*) is to instruct the executive agents of the robot/robot system chosen by the interpreter to execute the task to provide the desired service for the user.

In advanced man-machine co-operation the communication is usually bi-directional, i.e. the robot (agent) has to be able to construct questions or give information in a way that the human user/partner can easily understand. In case of verbatim communication, this can be solved by the usage of voice synthesizers (see e.g. [11]) or by simply writing the message out to a screen. Either way, the usage of grammar rules is necessary, since communicating with the user in his/her native language is the most convenient for the person. To achieve this, an advanced, language dependent grammar rule representation and processing method is needed.

4 Knowledge Based Hypotheses Building

An intelligent robot also has to be able to learn human reactions (e.g. instructions) together with the circumstances in which they appear and automatically initiate (execute) actions if the conditions become similar to the learned situation. A part of the robot's intelligence, the so called *Autonomous Action Planner* (*AAP*) shall be responsible for learning via hypotheses and for decision making whether or not to take actions according to what the system has learned.

4.1 Hypothesis Training

The task of the *Hypothesis Trainer* (*HTM*) is to determine which executive agent can execute the command. Its input is the parsed command and has no outputs. The algorithm of the hypothesis training works, as follows: If the command is an instruction, then it searches for a *hypothesis* that has the same action, subject, and numerical parameters. (There can be only one hypothesis like that, i.e. it is sufficient to get the first found). If there is none found, then a new hypothesis is created using the parameters of the command and a new *trigger* and new *conditions* are added with the (environmental variable) nodes connected to the subject node with heuristic edges. There will be as many conditions as many heuristic edges are connected to the subject node. The value and granularity of each condition is derived from the current value and granularity of the environmental variable.

If there already is such a hypothesis, then the algorithm checks its triggers. If there is a trigger with conditions triggered by the current values of the environmental variables, the algorithm increases the justification of that trigger and end the algorithm. If there is not any trigger like that, then a new trigger is added using heuristics just like it is explained above.

If the command is a prohibition then the algorithm searches for a corresponding hypothesis with triggers. If it does not have any triggers, it adds a new trigger to it. If there is one then adds a new condition to all its triggers using the negated fuzzy membership function of the prohibition.

4.2 Hypothesis Trigger Checking

The task of the *Hypothesis Trigger Checking* (*HTCM*) is to frequently check the conditions of the hypotheses in the hypothesis storage. If one is triggered, the *HTCM* sends the command of the hypothesis to the *CIM*, which instructs the appropriate executive agents to carry out the command.

For the hypothesis building, consider the following example: The human gives certain commands to the intelligent robot at certain times (e.g., "open the curtains" after waking up at 7:00, "make coffee" at 7:33 and 12:00, etc.). The robot makes and manages hypotheses based on these commands. The justification threshold of the triggers of hypotheses is set e.g. to 2, thus the system is only able to give out the command of the triggered hypotheses if the command has been detected at least 2 times under the same circumstances.

The left hand side of Fig. 6 shows an example for a learned hypothesis based on the instruction "make coffee". The command has been given at 7:33 and 12:00 (the parameters after "value" are: the value and granularity of the environmental variable and the affirmativeness of the condition). (In this example, the mood of the user was found frustrated (2) and neutral (3)).

Continuing the example, consider that the man's weekday and weekend schedules are different, though the robot does not know about this on the first time, so it makes coffee at 7:33 on Saturday, as well. In reaction, the user gives a prohibition: "DO NOT make coffee on Saturday". Thus, the hypothesis based on command "make coffee" is modified by the complement of fuzzy membership function Saturday as is shown in the right hand side of Fig. 6, meaning that the robot will not make any coffee if the value of the environmental variable "day" equals to 6.

```
Hypothesis: make coffee
  -Trigger: justification=2
    Conditions:
    - Node: time | value: 733 | 15 | true
    - Node: mood | value: 2 | 0 | true
  -Trigger: justification=2
    Conditions:
    - Node: time | value: 1200 | 15 | true
    - Node: mood | value: 3 | 0 | true
```

```
Hypothesis: make coffee
  -Trigger: justification=2
    Conditions:
    - Node: time | value: 733 | 15 | true
    - Node: mood | value: 2 | 0 | true
    - Node: day  | value: 6 | 0.2 | false
  -Trigger: justification=2
    Conditions:
    - Node: time | value: 1200 | 15 | true
    - Node: mood | value: 3 | 0 | true
    - Node: day  | value: 6 | 0.2 | false
```

Fig. 6. Instruction (left) and prohibition (right) type hypothesis

5 Conclusions

Human-robot co-operation asks for efficient knowledge storage and representation models which have a direct effect on the accessibility to the knowledge and also on the flexibility and adaptivity of the application and update of the knowledge. In this paper, a new flexible graph based structure is proposed which principally belongs to the wider family of conceptual graphs, however the authors suggested a different (modified and extended) structure of the existing

ones. The representation form has been adapted to a new range of problems and is applied in a new field. The novelty of the presented structure mainly lies in its specific edge types and that it stores and handles theoretical (abstract) and practical (real-word) knowledge separately. It is also able to build in uncertain and ambiguous knowledge by using fuzzy variables. As result, the proposed model makes easy to store, retrieve, modify, and extend theoretical and practical knowledge, to interpret commands and to associate them with physical means and actions, to adapt to changes, learn and build in new (uncertain) knowledge which aspects are especially important when the agents/robots are involved in human-robot communication as well.

Acknowledgments. This work was sponsored by the Hungarian National Scientific Fund (OTKA 105846).

References

1. Vaughan, B., Han, J.G., Gilmartin, E., Campbel, N.: Designing and Implementing a Platform for Collecting Multi-Modal Data of Human-Robot Interaction. Acta Polytechnica Hungarica 9(1), 7–17 (2012)
2. Jokinen, K.A., Scherer, S.: Embodied Communicative Activity in Cooperative Conversational Interactions - Studies in Visual Interaction Management. Acta Polytechnica Hungarica 9(1), 19–40 (2012)
3. Sowa, J.F.: Conceptual Graphs for a Data Base Interface. IBM J. of Research and Development 20(4), 336–357 (1976)
4. Zhang, P., Peng, J., Xi, J.: A Concept Graph Based Knowledge Model for Software Architecture. WRI World Congress on Software Engineering 3, 125–128 (2009)
5. Palacio, M.P., Sol, D., Gonzalez, J.: Graph-based knowledge representation for GIS data. In: 4th Mexican International Conference on Computer Science, pp. 117–124 (2003)
6. Qu, Q., Qiu, J., Sun, C., Wang, Y.: Graph-Based Knowledge Representation Model and Pattern Retrieval. In: 5th International Conference on Fuzzy Systems and Knowledge Discovery, Dalian, vol. 5, pp. 541–545 (2008)
7. Lee, J.-H., Hashimoto, H.: Intelligent Space. International Conference on Intelligent Robots and Systems 2, 1358–1363 (2000)
8. Várkonyi-Kóczy, A.R., Tóth, A.A.: ISpace a Tool for Improving the Quality of Life. Journal of Automation, Mobile Robotics and Intelligent Systems 3(4), 41–45 (2009)
9. Tusor, B., Várkonyi-Kóczy, A.R.: Virtual Eto-Environment in iSpace. In: International Workshop on Advanced Computational Intelligence and Intelligent Informatics, Suzhou, pp. G2-5.1-8 (2011)
10. Várkonyi-Kóczy, A.R., Tusor, B.: Human-Computer Interaction for Smart Environment Applications Using Fuzzy Hand Posture and Gesture Models. IEEE Trans. on Instrumentation and Measurement 60(5), 1505–1514 (2011)
11. Raitio, T., Suni, A., Yamagishi, J., Pulakka, H., Nurminen, J., Vainio, M., Alku, P.: HMM-Based Speech Synthesis Utilizing Glottal Inverse Filtering. IEEE Transactions on Audio, Speech, and Language Processing 19(1), 153–165 (2011)

An α-Level OWA Implementation of Bounded Rationality for Fuzzy Route Selection

Andrew R. Buck, James M. Keller, and Mihail Popescu

Department of Electrical and Computer Engineering
University of Missouri
Columbia, Missouri, USA
AndrewBuck@mail.missouri.edu
{KellerJ,PopescuM}@missouri.edu

Abstract. As people move through uncertain environments, they are often presented with multiple route choices. Deciding which route to take requires an understanding of the environmental features and how they affect the person's interpreted cost of each route. These quantities can be appropriately modeled as fuzzy numbers to capture the inherent uncertainty in human knowledge. We present an approach to guide a person's decision-making process through an environment modeled as a fuzzy weighted graph, using an α-level OWA operator to implement the principle of bounded rationality. A cost value is computed for each possible route choice, which can then be used to rank the set of routes and make a decision.

1 Introduction

Human geography is a diverse field involving the study of human traits in geographic space. One aspect of human geography is the study of how people navigate through environments. In contrast to many computational path-planning algorithms, humans do not always make optimal decisions when moving in an environment. Rather, we make decisions based on a cognitive map built from spatial knowledge and experience [1]. Rarely do these maps contain perfect information, as locations and spatial relationships between objects are measured using humanistic concepts such as "There is a hill off in the distance," or "This path is about three miles long." This type of uncertainty can be modeled using fuzzy sets.

An environment can be viewed as a graph of discrete locations represented as vertices and path transitions represented as edges. A person, or agent, may assign a cost value to each path segment based on their personal interpretation of the environmental features of that segment and how those features affect their mobility along the path. The uncertainty inherent in the agent's perception is modeled by using fuzzy numbers to represent the costs. In order to evaluate the total cost of a route between two locations, an agent must aggregate the costs of each route segment. By studying decisions made by artificial agents in an agent-based modeling scheme, we can gain insight on how groups of people will

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_24,

move in their environment under stress conditions, one of the goals of human geography.

The focus of this paper is to present a method for determining the cost assigned to a particular route, based on environmental features and an agent's attributes. We use an α-level Ordered Weighted Average (OWA) operator [2] to implement bounded rationality [3], the idea that agents have limited resources with which to make decisions, resulting in sub-optimal choices. Once a route cost has been established in the form of a fuzzy number, a variety of path-planning algorithms can be used to guide the agent's decision-making process. These include standard fuzzy shortest path algorithms such as [4] and [5] or the genetic algorithm approach of [6]. The remainder of this paper is outlined as follows. In Sect. 2, we define the concepts of fuzzy numbers, fuzzy weighted graphs, and bounded rationality as implemented by an α-level OWA operator. In Sect. 3, we present an example scenario consisting of three different routes and show how different agent types evaluate the environment differently. Our conclusions and ideas for future work are given in Sect. 4.

2 Path Planning in Uncertain Environments

2.1 Fuzzy Numbers

A fuzzy number is a convex, normalized fuzzy set $A : \mathbb{R} \to [0, 1]$ that provides a way of representing uncertainty in the value of a real number. The membership function $\mu_A(x)$ gives the degree of membership that a specific value x has in the fuzzy number A. Using Zadeh's extension principle, we can define the arithmetic operators for fuzzy numbers, as well as other functions such as maximization and minimization. For a function $f(A, B)$ operating on two fuzzy numbers A and B, the resulting fuzzy number is given as

$$\mu_{f(A,B)}(z) = \sup_{z=f(x,y)} \min\big(\mu_A(x), \mu_B(y)\big). \tag{1}$$

Because fuzzy numbers are convex, we can use α-cuts and interval arithmetic to quickly compute the result of a fuzzy computation. An α-cut of a fuzzy number is an interval ${}^{\alpha}A = [l, r]$ such that $\mu_A(x) \geq \alpha, x \in [l, r]$. The decomposition theorem states that a fuzzy number is simply the union of all α-cuts, $\alpha \in [0, 1]$. For each value of α, the result of a convex, continuous function $f({}^{\alpha}A, {}^{\alpha}B)$ on the α-cuts of two fuzzy numbers ${}^{\alpha}A$ and ${}^{\alpha}B$ is computed as

$$\begin{aligned} f({}^{\alpha}A, {}^{\alpha}B) = f\big([a, b], [c, d]\big) &= [l, r], \\ l &= \min\big(f(a, c), f(a, d), f(b, c), f(b, d)\big), \\ r &= \max\big(f(a, c), f(a, d), f(b, c), f(b, d)\big). \end{aligned} \tag{2}$$

Although any fuzzy set that satisfies the conditions of convexity and normality can be used to represent a fuzzy number, we often use triangular membership functions for their simplicity. We define a triangular fuzzy number as a 3-tuple (a, b, c), where the interval $[a, c]$ is the support and b is the peak of the fuzzy number.

2.2 Fuzzy Weighted Graphs

An environment can be represented as a fuzzy weighted graph $\widetilde{G} = (\mathcal{V}, \mathcal{E}, \mathcal{X})$, where $\mathcal{V} = (v_1, \cdots, v_N)$ is the set of vertices representing the discrete locations in the environment, $\mathcal{E}$ is the set of edges $e_k = (v_i, v_j) \in \mathcal{V} \times \mathcal{V}$ representing possible transitions from one location to another, and $\mathcal{X}$ is a set of fuzzy weights assigned to each edge. For each edge e_k, we denote a vector of fuzzy numbers $\widetilde{\boldsymbol{X}}(e_k) = \left(\widetilde{X}_1(e_k), \cdots, \widetilde{X}_r(e_k)\right)$, where each element $\widetilde{X}_i(e_k)$ represents a different measured feature of the edge e_k (e.g. length, slope, path type, etc.).

For each edge $e_k = (v_i, v_j)$, we denote $tail(e_k) = v_i$ and $head(e_k) = v_j$. An s, t path $\boldsymbol{p}$ in $\widetilde{G}$ is an n-tuple $\boldsymbol{p} = (e_1, \cdots, e_n) \in \mathcal{E}^n$ such that $head(e_i) = tail(e_{i+1})$ for $i = 1, \cdots, n-1$. We denote the start of the path as $s = tail(e_1)$, and the end of the path as $t = head(e_n)$. $\mathcal{P}(s,t)$ is the set of all s, t paths. For any path $\boldsymbol{p} \in \mathcal{P}(s,t)$, we can define an aggregated weight vector $\widetilde{\boldsymbol{F}}(\boldsymbol{p}) = \left(\widetilde{F}_1(\boldsymbol{p}), \cdots, \widetilde{F}_r(\boldsymbol{p})\right)$, where $\widetilde{F}_i(\boldsymbol{p})$ is the aggregation of all fuzzy numbers $\widetilde{X}_i(e_k), e_k \in \boldsymbol{p}$ for the feature i. The choice of aggregation function depends on the feature, as some features such as distance are well suited for a summation-type aggregation, whereas other features such as slope might be better aggregated with a maximization operator.

2.3 Bounded Rationality

An agent decision-maker trying to plan a route from point s to point t will ultimately need to choose a path from the set $\mathcal{P}(s,t)$. To do this, the agent will need to have a method for comparing paths. For a given path $\boldsymbol{p} \in \mathcal{P}(s,t)$, the aggregated weight vector $\widetilde{\boldsymbol{F}}(\boldsymbol{p})$ provides a summarization of the various measurable features of the path. Not all agents are identical, however, so we define an agent-specific interpretation $\widetilde{\boldsymbol{A}}(\boldsymbol{p}) = \left(\widetilde{A}_1(\boldsymbol{p}) = \tilde{g}_1\left(\widetilde{F}_1(\boldsymbol{p})\right), \cdots, \widetilde{A}_r(\boldsymbol{p}) = \tilde{g}_r\left(\widetilde{F}_r(\boldsymbol{p})\right)\right)$ where each function $\tilde{g}_i\left(\widetilde{F}_i(\boldsymbol{p})\right)$ is defined independently for each feature. These functions define how much various environmental properties affect the agent's interpreted cost of a path. As a rule of thumb, the values should be scaled into units corresponding to the amount of effort the agent attributes to moving along a path with each of the various features. We avoid explicitly normalizing the resulting vector $\widetilde{\boldsymbol{A}}(\boldsymbol{p})$ to allow certain features to dominate the final cost in all circumstances. For example, most agents would consider a path that contains no off-road segments to be far less costly than a path that contains several off-road segments. In this case, the off-road feature should be scaled by a very large number to guarantee that it will be the dominant factor in the final cost evaluation. Care should be taken to ensure that all of the resulting elements of $\widetilde{\boldsymbol{A}}(\boldsymbol{p})$ are all appropriately scaled.

The principle of bounded rationality states that a decision-maker cannot always consider all sources of information and tends to utilize only the most prominent features when making a decision. We implement bounded rationality using an α-level OWA operator to reduce the agent interpretation vector

$\widetilde{\boldsymbol{A}}(\boldsymbol{p})$ to a single fuzzy cost value $\widetilde{C}(\boldsymbol{p})$. An α-level OWA operator is a mapping $\Phi_{\widetilde{\boldsymbol{W}}} : (\widetilde{A}_1(\boldsymbol{p}), \cdots, \widetilde{A}_r(\boldsymbol{p})) \mapsto \widetilde{C}(\boldsymbol{p})$ where $\widetilde{\boldsymbol{W}} = (\widetilde{W}_1, \cdots, \widetilde{W}_r)$ is a vector of fuzzy number weights defined on the domain $[0, 1]$. The *Alpha-Level Approach* defined in [2] provides a method to compute $\Phi_{\widetilde{\boldsymbol{W}}}$ using α-cuts. For each $\alpha \in [0, 1]$,

$$
{}^{\alpha}\Phi_{\widetilde{\boldsymbol{W}}}({}^{\alpha}\widetilde{A}_1(\boldsymbol{p}), \cdots, {}^{\alpha}\widetilde{A}_r(\boldsymbol{p})) = \left(\frac{\sum_{i=1}^{r} w_i a_{\sigma(i)}}{\sum_{i=1}^{r} w_i} \middle| \begin{matrix} w_i \in {}^{\alpha}\widetilde{W}_i \\ a_i \in {}^{\alpha}\widetilde{A}_i(\boldsymbol{p}) \\ i = 1, \cdots, r \end{matrix} \right), \tag{3}
$$

$$
\begin{aligned}
\text{where} \quad & \sigma : (1, \cdots, r) \to (1, \cdots, r) \\
\text{such that} \quad & a_{\sigma(i)} \geq a_{\sigma(i+1)} \ \forall \ i = 1, \cdots, r-1.
\end{aligned}
$$

From the set of ${}^{\alpha}\Phi_{\widetilde{\boldsymbol{W}}}$, the final cost value can be obtained as

$$
\widetilde{C}(\boldsymbol{p}) = \bigcup_{0 \leq \alpha \leq 1} \alpha \cdot {}^{\alpha}\Phi_{\widetilde{\boldsymbol{W}}}({}^{\alpha}\widetilde{A}_1(\boldsymbol{p}), \cdots, {}^{\alpha}\widetilde{A}_r(\boldsymbol{p})). \tag{4}
$$

An efficient algorithm to quickly compute the α-level OWA operator is given in [2]. By changing the weight vector, different aggregation operations can be defined, such as averaging the first k elements or considering only the single most influential feature.

3 Example

To demonstrate our method, consider the following hypothetical scenario. An agent is trying to reach a goal position on the opposite side of a large hill. The agent is presented with three route choices: through the woods, over the hill, or the long way around. The shortest route goes directly over the hill, however this route is very steep and unpaved. The next shortest route goes through a forrest which provides shade and has only a mild elevation change, but the route is still unpaved and also has a stream crossing with no bridge. The last route is the longest, but it is completely paved and has almost no elevation change. One can imagine three different types of agents that would prefer different routes based on their personal traits. For example, an athletic agent that does not care about elevation or path quality might prefer the direct route over the hill, whereas a less active agent might need to take the long route to avoid climbing or going off the paved route. Finally, a somewhat capable agent might prefer to take the path through the woods – even with the dirt path, water crossing, and elevation change – in order to avoid walking in the sun.

We model this scenario with the fuzzy weighted graph shown in Fig. 1. The fuzzy weights assigned to each edge are the triangular fuzzy numbers shown in Table 1. These represent the distance, slope, path quality, amount of shade, and number of water crossings as measured by the agent. Note that the shade values are defined so that unshaded routes have a greater cost value. It is appropriate

to use fuzzy numbers to represent these values, as an agent will likely not have perfect information. In this example, the route through the woods is A–B–C–E–F, the route over the hill is A–B–E–F, and the long way around is A–B–D–E–F. For each of these routes, we aggregate the features using fuzzy summation for distance, path, shade, and water, and using the fuzzy max operator for slope to represent how an agent may only care about the steepest part of a path. The resulting aggregated feature vectors are shown in Fig. 2.

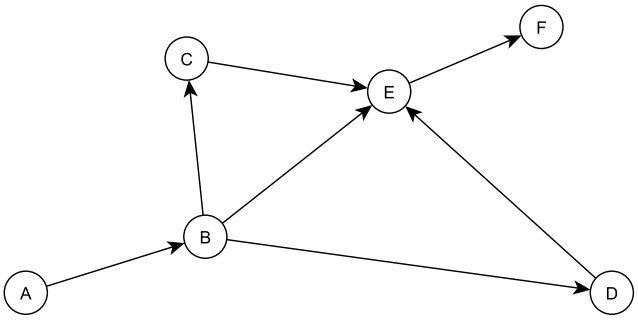


Fig. 1. Three Route Fuzzy Weighted Graph

Table 1. Fuzzy edge weights for the graph in Fig. 1

Edge	Distance	Slope	Path	Shade	Water
(A, B)	$(1, 2, 3)$	$(0, 0.64, 2.6)$	$(0, 0, 0.2)$	$(1, 2, 3)$	$(0, 0, 0.2)$
(B, C)	$(2, 4, 6)$	$(0.8, 2.8, 4.8)$	$(1.5, 3.5, 5.5)$	$(0, 0.5, 2.5)$	$(0, 0, 0.4)$
(B, D)	$(3.5, 7, 11)$	$(0, 0.57, 2.6)$	$(0, 0, 0.7)$	$(3.5, 7, 11)$	$(0, 0, 0.7)$
(B, E)	$(2.5, 5, 7.5)$	$(5.5, 7.5, 9.5)$	$(1.5, 4, 6.5)$	$(2.5, 5, 7.5)$	$(0, 0, 0.5)$
(C, E)	$(2.5, 5, 7.5)$	$(0.86, 2.9, 4.9)$	$(2, 4.5, 7)$	$(0, 0.5, 3)$	$(0, 1, 2.3)$
(D, E)	$(4, 8, 12)$	$(0, 0.7, 2.7)$	$(0, 0, 0.8)$	$(4, 8, 12)$	$(0, 0, 0.8)$
(E, F)	$(1, 2, 3)$	$(0, 0.25, 2.3)$	$(0, 0, 0.2)$	$(1, 2, 3)$	$(0, 0, 0.2)$

We now define three agent types with different interpretation functions. In this example, each feature is multiplied by a scalar value such that for an agent l, $\widetilde{A}_i^{(l)} = \tilde{g}_i^{(l)}\left(\widetilde{F}_i^{(l)}\right) = \beta_i^{(l)} \cdot \widetilde{F}_i^{(l)}$. The β values for the three agents in our example are given in Table 2. The first agent associates a moderate cost with steep and unshaded routes, as well as a high cost for water crossings. The second agent weights long routes, unshaded routes, and water crossings with an equally high cost. The third agent has a very high cost associated with steep routes, a somewhat high cost associated with unpaved routes, and a moderately high cost for water crossings.

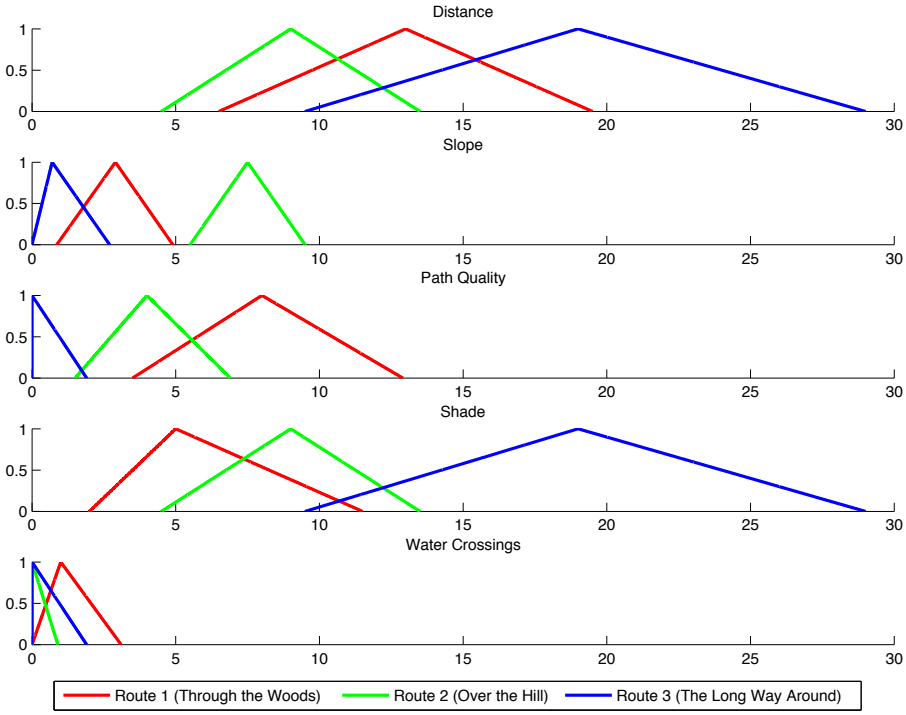


Fig. 2. Aggregation of Feature Values for Three Route Example

Table 2. Agent Interpretation Values (β)

Agent	Distance	Slope	Path	Shade	Water
1	1	10	1	10	100
2	10	1	1	10	10
3	1	100	50	1	10

Table 3. α-level OWA Weights

	$\widetilde{W}_1$	$\widetilde{W}_2$	$\widetilde{W}_3$	$\widetilde{W}_4$	$\widetilde{W}_5$
$\widetilde{\boldsymbol{W}}_{(\text{Max})}$	$(0, 0.5, 1)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
$\widetilde{\boldsymbol{W}}_{(\text{Top 2})}$	$(0.5, 1, 1)$	$(0.5, 1, 1)$	$(0, 0, 0)$	$(0, 0, 0)$	$(0, 0, 0)$
$\widetilde{\boldsymbol{W}}_{(\text{Average})}$	$(0, 0.2, 0.4)$	$(0, 0.2, 0.4)$	$(0, 0.2, 0.4)$	$(0, 0.2, 0.4)$	$(0, 0.2, 0.4)$

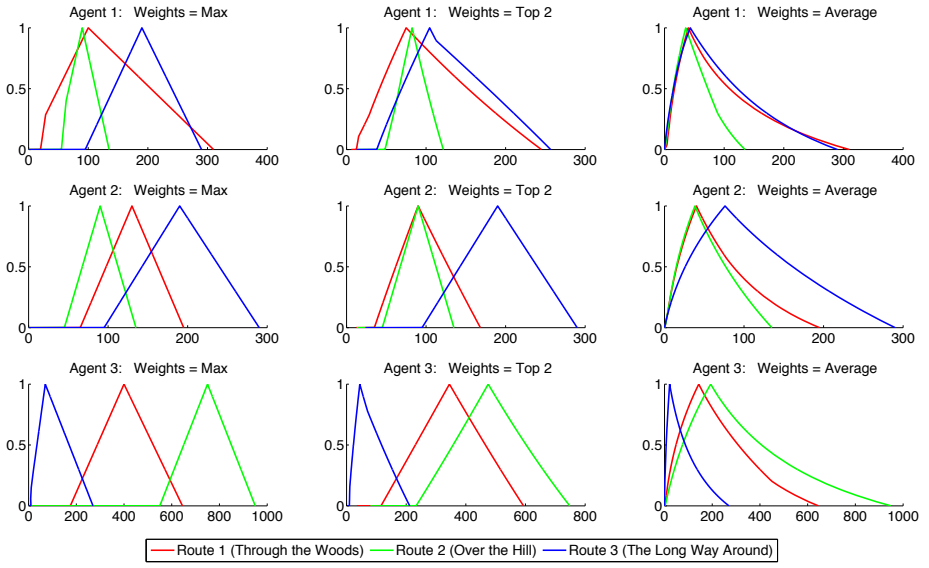


Fig. 3. Bounded Rationality Cost Evaluation for Three Route Example

For each agent type, we evaluate the example environment using three different sets of fuzzy number weights, shown in Table 3. The resulting cost evaluation for each route is shown in Fig. 3. We see that the first agent tends to prefer the forrest route, but due to the large cost associated with water crossings, this agent also likes the direct route over the hill. As the OWA weights move from the single most prominent feature toward the average of all features, the distinction between routes becomes less apparent. The second agent prefers the hilly route for the "max" weights, but considers the forrest route to be about as good as the hilly route for the other weights. The third agent clearly prefers the long way around for all weight choices. From these costs, the agents can use any appropriate fuzzy ranking method to pick a route to follow. A method such as the Liou and Wang index [7] that allows for an additional optimism/pessimism parameter would be appropriate for this type of problem.

4 Conclusion and Future Work

Fuzzy numbers are a natural way to represent how an agent interprets its environment. The α-level OWA operator allows an agent to aggregate multiple fuzzy route features using the principle of bounded rationality. This allows different types of agents to each interpret an environment in their own way and make unique decisions. The agent interpretation functions and the OWA weight vector are quite flexible, and can be defined to fit many different domains.

A logical extension of this work is to incorporate the cost evaluation into a general path-finding algorithm and an agent movement model. A fuzzy shortest-path algorithm that can return multiple possible routes would be a good way to provide several route choices to a decision-making agent. Agent movement could then be guided by following the least-cost route.

Acknowledgments. The authors wish to thank Dr. John Greer of the National Geospatial Intelligence Agency for support of this research under contract NGA HM 1582-10-C-0013.

References

1. Kitchin, R., Blades, M.: The Cognition of Geographic Space. I.B. Tauris, London (2002)
2. Zhou, S., Chiclana, F., John, R.I., Garibaldi, J.M.: Alpha-Level Aggregation: A Practical Approach to Type-1 OWA Operation for Aggregating Uncertain Information with Applications to Breast Cancer Treatments. IEEE Transactions on Knowledge and Data Engineering 23(10), 1455–1468 (2011)
3. Herbert, S.: A Behavioral Model of Rational Choice. In: Models of Man, Social and Rational: Mathematical Essays on Rational Human Behavior in a Social Setting. Wiley, New York (1957)
4. Hernandes, F., Lamata, M.T., Verdegay, J.L., Yamakami, A.: The Shortest Path Problem on Networks with Fuzzy Parameters. Fuzzy Sets and Systems 158(14), 1561–1570 (2007)
5. Cornelis, C., De Kesel, P., Kerre, E.E.: Shortest Paths in Fuzzy Weighted Graphs. International Journal of Intelligent Systems 19(11), 1051–1068 (2004)
6. Chakraborty, B., Maeda, T., Chakraborty, G.: Multiobjective Route Selection for Car Navigation System Using Genetic Algorithm. In: Proceedings of the 2005 IEEE Midnight-Summer Workshop on Soft Computing in Industrial Applications, pp. 190–195 (2005)
7. Liou, T., Wang, M.J.: Ranking fuzzy numbers with integral value. Fuzzy Sets and Systems 50(3), 247–255 (1992)

Indices for Introspection on the Choquet Integral

Stanton R. Price[1], Derek T. Anderson[1], Christian Wagner[2], Timothy C. Havens[3], and James M. Keller[4]

[1] Electrical and Computer Engineering Department
Mississippi State University, Mississippi State, MS, USA
sp438@msstate.edu , anderson@ece.msstate.edu
[2] School of Computer Science, Horizon Digital Economy Research Institute
University of Nottingham, Nottingham, UK
christian.wagner@nottingham.ac.uk
[3] Electrical and Computer Engineering, Computer Science
Michigan Technological University, Houghton, MI, USA
thavens@mtu.edu
[4] Electrical and Computer Engineering
University of Missouri, Columbia, MO, USA
kellerj@missouri.edu

Abstract. *Fuzzy measures* (FMs) encode the *worth* (or importance) of different subsets of information sources in the *fuzzy integral* (FI). It is well-known that the *Choquet FI* (CFI) often reduces to an elementary aggregation operator for different selections of the FM. However, FMs are often learned from training data or they are derived from the densities (worth of just the singletons). In these situations an important question arises; what is the resultant CFI really doing? Is it aggregating data relative to an additive measure, a possibility measure, or something more complex and unique? Herein, we introduce new indices (distance formulas) and fuzzy sets that capture the degree to which the CFI is behaving like a set of known aggregation operators. This has practical application in terms of gaining a deeper understanding into a given problem, guiding new learning methods and evaluating the CFI's benefit.

1 Introduction

Fuzzy measures (FMs) are often used to encode the (possibly subjective) *worth* of different subsets of information sources. The *Choquet fuzzy integral* (CFI), an aggregation operator, is a way to combine the information encoded in a FM with the (objective) evidence or support of some hypothesis, e.g., sensor data, algorithm outputs, expert opinions, etc. FMs can be obtained in a number of ways: *quadratic program* (QP) [1], learning algorithm (e.g., genetic algorithm [2], punishment-reward [3], gradient descent [4]), S-Decomposable measure (e.g., Sugeno λ-fuzzy measure [5], belief measure), etc. However, the vast majority of FM learning and density deriving techniques (i.e., a FM acquired from just the worth of the singletons) do not provide us with any knowledge about how the

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_25,

resultant CFI is truly aggregating the data. Characterization methods are needed to help us discover what the CFI is really doing relative to a given (learned or derived) FM. The applicability of such a tool is wide-ranging: from an increased low-level understanding of how the sources are being fused to possibly helping us learn which sources are needed and which are providing negligible information to the problem at hand. Such a tool can also help us justify when the CFI is indeed doing something unique by acting outside the bounds of a simple known aggregation operator (e.g., weighted sum). It also has the potential to aid us in predicting the benefit of adding (or aiding the design of) a new sensor in multi-sensor fusion [6, 7].

Herein, we put forth new indices (distance formulas) and corresponding fuzzy sets that help assess the degree of similarity a learned or density-derived FM, which drives the behavior of the CFI, has to a subset of known aggregation operators. As stated above, there are many potential applications for indices that characterize the CFI. The work put forth herein is a first step in terms of identifying such indices. Later, focused work will improve upon these building blocks, indices, and explore their role in fusion.

The remainder of this article is structured as follows. Section 2 is a quick review of the FM, followed by learning and density-based derivation methods used herein to acquire it. Equations for characterizing the FM's behavior is introduced in Section 3. Last, preliminary experimental results are reported in Section 4.

2 Background

The FM does not possess the restrictive property of additivity. Instead, it has the weaker property of monotonicity (and typically normality). With respect to a set of N information sources, $X = \{x_1, ..., x_N\}$, the FM is often used to encode the (possibly subjective) worth of each subset in the power set 2^X. In the CFI, the FM is the mechanism that ultimately dictates how information is aggregated.

Definition 1. ***(Fuzzy Measure)*** *For a finite set of N information sources, X, the FM is a set-valued function $g : 2^X \rightarrow [0, 1]$ with the following conditions:*

1. *(Boundary Conditions) $g(\phi) = 0$ and $g(X) = 1$,*
2. *(Monotonicity) If A, $B \subseteq X$ with $A \subseteq B$, then $g(A) \leq g(B)$.*

Note, if X is an infinite set, there is a third condition guaranteeing continuity.

Definition 2. ***(Choquet Fuzzy Integral)*** *For a finte set of N information sources, X, FM g, and partial support function $h : X \rightarrow [0, 1]$, the CFI is*

$$\int h \circ g = \sum_{i=1}^{N} \omega_i h(x_{\pi(i)}), \tag{1}$$

where $\omega_i = (G_{\pi(i)} - G_{\pi(i-1)})$, $G_{(i)} = g(\{x_{\pi(1)}, ..., x_{\pi(i)}\})$, $G_{\pi(0)} = 0$, $h(x_i)$ is the strength in the hypothesis from source x_i, and $\pi(i)$ is a sorting on X such that $h(x_{\pi(1)}) \geq \ldots \geq h(x_{\pi(N)})$.

In contrast to relatively simple and classical additive aggregation techniques, e.g., weighted sum, the CFI is a more powerful aggregation operator that can account for important *interactions* (when/if present) between different subsets of information sources. Interactions, as referred to herein, are the combining of hypotheses between the subsets of X (e.g., combining $h(x_1)$ with $h(x_2)$ results in an increase or decrease to the overall confidence of a decision). The way in which these interactions impact the aggregated result is guided by the selection of FM. An example of the CFI is the following. Consider the case of multi-sensor (e.g., electro optical infrared (EO/IR), ground penetrating radar (GPR), and visual spectrum (VS)) explosive hazard detection. Furthermore, assume these sensors are co-registered, meaning, we know the mapping from one pixel, (i,j), in sensor (image) k to its corresponding pixel in sensor (image) m, or each source is individually mapped into some global coordinate system. Therefore, $N = 3$ and x_1 is EO/IR, x_2 is GPR, and x_3 is VS. For a specific world location, such as (33.4526 , -88.7874), a hypotheses might be, "there is an explosive hazard at this world location." For simplicity, assume the information provided by each sensor is its individual strength, in [0,1], in the above hypothesis. Thus, $g(x_1, x_3)$ is the importance of EO/IR and VS relative to answering this hypothesis and the CFI is the combined belief that the location is a hazard and dangerous.

Next, we review a few FM learning and density deriving methods since they guide the work and are a large part of the proposed experiments. Specifically, we review learning a FM from data via QP, the class of S-Decomposable measures, possibility and necessity measures, and the Sugeno λ-FM.

2.1 Quadratic Program for Learning the FM from Data

In [8], Grabisch shows how to compute the FM using QP. Training data is used and the FM that makes the FI fit the data with minimum *sum of squared error* (SSE) is used. Let $T = \{(o_j, \alpha_j) : j = 1, ..., m\}$ denote the set of training data, where o_j is the the j^{th} object/instance and $\alpha_j \in [0, 1]$ is the label of o_j. Referring to the multi-sensor EHD system example from Section 2, instance o_j would be a $[1 \times N]$ vector with confidence values for each of the N information sources and α_j would have a label of target or not target. With respect to instance j, the CFI can be compactly represented in linear algebra form as $\boldsymbol{A}_{o_j}^t \boldsymbol{g} + h(o_j; x_{\pi(N)})$, where $\boldsymbol{g}$ is a vector (of size $2^N - 1 \times 1$) of lexicographic ordered FM terms (e.g., $\boldsymbol{g}(1) = g(\{x_1\})$, $\boldsymbol{g}(2) = g(\{x_2\})$, $\boldsymbol{g}(N+1) = g(\{x_1, x_2\})$, $\boldsymbol{g}(2^N - 1) = g(\{x_1, x_2, \ldots, x_N\})$, etc.) and $\boldsymbol{A}_{o_j}$ is a vector (of size $2^N - 2 \times 1$) of differences in h values. For example, if the largest h value index is k, then $\boldsymbol{A}_{o_j}(k) = h(o_j; x_{\pi(1)}) - h(o_j; x_{\pi(2)})$. The FM is computed by minimizing

$$\frac{1}{2}\boldsymbol{g}^t(2\sum_{j=1}^{m} \boldsymbol{A}_{o_j}\boldsymbol{A}_{o_j}^t)\boldsymbol{g} + \sum_{j=1}^{m}(2(h(o_j; x_{\pi(N)}) - \alpha_j)\boldsymbol{A}_{o_j}^t)\boldsymbol{g}, \tag{2}$$

subject to $(-C)\boldsymbol{g} - \boldsymbol{b} \geq \mathbf{0}$ and $\mathbf{0} \leq \boldsymbol{g} \leq \mathbf{1}$, where $\boldsymbol{b}$ is a vector (of size $N(2^{N-1} - 1) \times 1$) of all 0's except for the last N entries, which are of value -1, and C is a matrix (of size $N(2^{N-1} - 1) \times 2^N - 2$) of monotonic fuzzy constraints (see [8] for details).

2.2 S-Decomposable Measure

The Sugeno λ-FM is one of the most widely used methods for deriving a FM from just the densities. However, the Sugeno λ-FM is a member of a much larger class of FMs, S-decomposable measures.

Definition 3. ***(S-Decomposable Measure).*** *Let S be a t-conorm. A FM g is called an S-decomposable measure if $g(\phi) = 0$, $g(X) = 1$, and for all A, B such that $A \cap B = \phi$,*

$$g(A \cup B) = S(g(A), g(B)).$$

2.3 Possibility and Necessity Measures

Popular examples of S-decomposable measures include the following.

Definition 4. ***(Possibility Measure).*** *A FM Π is called a possibility measure if $\Pi(\phi) = 0$, $\Pi(X) = 1$, if $A \subseteq B$, $\Pi(A) \leq \Pi(B)$, and*

$$\Pi(A \cup B) = \max\left(\Pi(A), \Pi(B)\right). \tag{3}$$

Definition 5. ***(Necessity Measure).*** *A FM Nec is called a necessity measure if $Nec(\phi) = 0$, $Nec(X) = 1$, if $A \subseteq B$, $Nec(A) \leq Nec(B)$, and*

$$Nec(A \cap B) = \min\left(Nec(A), Nec(B)\right). \tag{4}$$

2.4 Sugeno λ-Fuzzy Measure

Definition 6. ***(Sugeno λ-Fuzzy Measure).*** *A measure g_λ is called a Sugeno λ-FM if it is a FM, [Def 1], and if for $A, B \subseteq X$ and $A \cap B = \phi$,*

$$g_\lambda(A \cup B) = g_\lambda(A) + g_\lambda(B) + \lambda g_\lambda(A) g_\lambda(B), \lambda > -1.$$

Sugeno showed that λ can be found by solving

$$\lambda + 1 = \prod_{i=1}^{N} \left(1 + \lambda g(\{x_i\})\right), \tag{5}$$

where Sugeno showed that there is exactly one real solution such that $\lambda > -1$.

3 Indices for Analysis of the Choquet Integral

In this section, we put forth a new set of indices (distance formulas) and fuzzy sets that capture the degree to which the CFI is behaving like a known aggregation operator for a given FM. Specifically, we put forth definitions for max, min, average, the generic case of an *ordered weighted average* (OWA), and the degree to which a FM is in accordance with a possibility FM. These FMs are based on the theoretical investigation of the CFI for different FMs by Keller et al. [9].

Before moving on, a few terms used herein need to be established. First, a *lattice* is induced according to the set of monotonicity constraints of the FM. It is simple to think about and talk about a FM in terms of its lattice. Second, a *node* in the lattice refers to the value of the FM for a given *non-empty* subset of X. *Layer* k in the lattice, denoted $L(k)$, is the set of all FM terms for subsets of 2^X that have cardinality equal to k. For example, if $N = 3$, $L(1) = \{g(x_1), g(x_2), g(x_3)\}$, $L(2) = \{g(x_1, x_2), g(x_1, x_3), g(x_2, x_3)\}$, and $L(3) = g(x_1, x_2, x_3) = g(X)$.

3.1 Maximum Operator

It is well-known that the CFI is a max operator when the value at each node (used herein when referring to non-empty subsets of X) in the lattice is 1. A naive approach for measuring the distance in which a given FM behaves like the max operator is

$$\frac{1}{2^N - 2} \sum_{I \in 2^X \setminus \{\phi, X\}} (1 - g(I)). \tag{6}$$

Equation 6 is a distance, which has a value of 0 when we have a max FM and a value 1 in the case of minimum (dual of max). It can be easily converted into a similarity by subtracting it from 1. However, Eqn. 6 is lacking as it assigns equal importance to each node in the lattice and it does little-to-nothing to guarantee the specific properties that we expect a max operator to possess. It is also very hard to interpret what the distance is really measuring (outside comparing it to the extreme case of min). It appears, to us, that whatever the measure is, it should calculate its distance relative to a FM of all 1s (max FM). We assert that a quality distance measure should enforce the idea that a given FM needs to indeed behave like an OWA and differences at different layers in the lattice are not equal (as we expect max to move to soft max and then a different operator from there). We begin by assigning a weight (penalty or cost) for each layer in the FM,

$$\boldsymbol{W} = \frac{\left[\frac{1}{N}, \ldots, 1\right]}{\sum_{i=1}^{N} \frac{i}{N}}. \tag{7}$$

Now, we put forth a formula that adheres to the concepts discussed above,

$$D_{\max} = \sum_{k=1}^{1} \frac{\boldsymbol{W}(k)}{2}(T_1 + T_4) + \left[\sum_{k=2}^{N} \frac{\boldsymbol{W}(k)}{3}(T_1 + T_2 + T_4)\right], \tag{8}$$

$$T_1 = 1 - \left(\frac{\sum_{I \in L(k)} g(I)}{|L(k)|}\right), \tag{9}$$

$$T_2 = \left(\frac{\sum_{I \in L(k)} g(I)}{|L(k)|}\right) - \left(\frac{\sum_{J \in L(k-1)} g(J)}{|L(k-1)|}\right), \tag{10}$$

$$T_3 = \frac{\sum_{I \in L(k)} g(I)}{|L(k)|}, \tag{11}$$

$$T_4 = \frac{\sum_{I \in L(k)} (g(I) - T_3)^2}{|L(k)| - 1}. \tag{12}$$

Specifically, T_1 measures the difference from the expected value of 1, T_2 is the extraction of the OWA weights (which should be $[1, 0, \ldots, 0]$ for max), and T_4 is an unbiased estimator of the variance of the values at layer k in the lattice. While $D_{\max}$ measures the distance of a given FM to max, what we really want is a membership degree in which 0 means not max and 1 means it is max. However, how should this membership function be defined on the domain $D_{\max}$? We propose that one specifies a definition for soft max and measures its corresponding $D_{\max}$ value. For example, in our experiments section we used $N = 3$ and obtained $D_{\max} \approx 0.0637$. Next, a membership function is selected; herein, we used a Gaussian shaped membership function whose peak is centered at 0 and σ is derived so that the 0.5 membership degree is at the corresponding location ($D_{\max}$) for soft max. In this respect, one calibrates the concept of compatibility relative to when we expect a FM to become increasingly dissimilar to max.

3.2 Minimum Operator

With respect to the min operator, each node in the lattice, except for $g(X)$ which is 1 by definition, should be 0. We also expect that the FM is an OWA and we calibrate it relative to soft min. As min and max are duals, their resulting calculations are largely similar,

$$D_{\min} = \sum_{k=1}^{1} \frac{\boldsymbol{W_2}(k)}{2}(T_3 + T_4) + \left[\sum_{k=2}^{N-1} \frac{\boldsymbol{W_2}(k)}{3}(T_3 + T_2 + T_4)\right], \tag{13}$$

where the weights are

$$\boldsymbol{W_2} = \frac{\left[1, \ldots, \frac{1}{N-1}\right]}{\sum_{i=1}^{N-1} \frac{i}{N-1}}. \tag{14}$$

Note, T_3 measures the difference of a layer to the expected value of 0. Furthermore, we follow a similar calibration process to that outlined above (Sec. 3.1) to calibrate min, with the only difference being we specify a definition for soft min instead of soft max.

3.3 Average Operator

A unique property of the average operator, with respect to the FM, is it has value $\frac{k}{N}$ at layer k in the lattice. This is captured in the following equation,

$$D_{avg} = \frac{1}{2^N - 2} \sum_{k=1}^{N-1} \sum_{I \in L(k)} \left| g\left(I\right) - \frac{k}{N} \right|. \tag{15}$$

Unlike max and min, it is not as clear what the membership function and specific calibration (point where the membership drops off to value 0.5 and shape of the function) should be. It could however, be calibrated or fit to a user's specification or application. Herein we fit a z-membership function which was calibrated to have membership value 0.5 at $(1/N)^2$. Note that the value used for calibration was experimentally determined based on our preliminary experiments.

3.4 OWA Operator

Based on the definitions above, measuring the degree to which the CFI, for a given FM, is compatible with an OWA operator is relatively simple. The criteria is that sets of equal size cardinality in the lattice have equal measure value,

$$D_{OWA} = \frac{1}{N-1} \sum_{k=1}^{N-1} \sqrt{T_4}. \tag{16}$$

Again, it is not obvious how to take this distance and construct a membership function that everyone would agree with that captures the exact rate at which a given FM is moving away from being an OWA operator. The membership function can be specified by a user or specific application. Herein, we fit a triangular membership function across the domain $[0, 1]$ such that the left and center points are at 0 and the right point is at 1. This membership function was chosen intuitively and supported experimentally through our preliminary experiments.

3.5 Possibility Measure

The final index put forth here captures the degree to which the CFI is aggregating data under the guidance of a possibility measure. A possibility measure has the unique distinction that for $A \subset X$, $g(A)$ is the maximum over the densities corresponding to A. We therefore put forth the following distance measure,

$$D_{Poss} = \frac{\sum_{A \in 2^X \setminus \{\phi, \{x_1\}, \ldots, \{x_N\}, X\}} \left| g(A) - \max_{i \in A} g(\{x_i\}) \right|}{2^N - 2 - N}. \tag{17}$$

As in the sections above, we empirically calibrated the membership function. Herein, we fit a triangular membership function to D_{Poss} such that the left and center points are at 0 and the right point is at 1.

4 Preliminary Results

In this section, we report preliminary results that demonstrate the behavior of our proposed indices. Once again, these indices are put forth in order to characterize the behavior of the CFI. The goal of this work is to explore a few such indices. The focus of this work is not to vigorously analyze the results of different learning or data driven experiments for different application domains. That is the subject of future work. Two sets of experiments were conducted: using known FMs (Sec. 4.1), and using learned and density derived FMs (Sec. 4.2). For simplicity of reporting and describing the following experiments, the following cases are performed for $N = 3$.

4.1 Experiment 1: Known Aggregation Operators

The experiments presented in Table 1 are put forth to demonstrate the behavior of our indices with respect to known user specified FMs.

Table 1. Results of proposed indices for the case of known FMs and $N = 3$

	User Specified FM		**Max**	**Min**	**Avg**	**OWA**	**Poss**
(a)	Max	dist.	0	0.44	0.5	0	0
		memb.	1	0	0	1	1
(b)	Soft Max OWA weights of (0.7, 0.2, 0.1)	dist.	0.07	0.35	0.3	0	0.2
		memb.	0.5	0	0	1	0.8
(c)	Min	dist.	0.36	0	0.5	0	0
		memb.	0	1	0	1	1
(d)	Average	dist.	0.18	0.22	0	0	0.33
		memb.	0.01	0.01	1	1	0.66
(e)	OWA weights of (0.52, 0.08, 0.40)	dist.	0.16	0.24	0.12	0	0.08
		memb.	0.04	0	0.35	1	0.92
(f)	Possibility Measure densities of (0.4, 0.6, 0.8)	dist.	0.12	0.31	0.18	0.15	0
		memb.	0.13	0	0.02	0.84	1
(g)	Sugeno λ-FM densities of (0.1, 0.1, 0.1)	dist.	0.28	0.1	0.27	0	0.25
		memb.	0	0.41	0	1	0.74
(h)	Sugeno λ-FM densities of (0.5, 0.2, 0.9)	dist.	0.13	0.35	0.25	0.27	0.07
		memb.	0.11	0	0	0.72	0.92

As expected, each of the simple aggregation operators have full membership to their respective index. Also, note that a given FM can be many types of FMs at once (i.e., for the max FM, it is also an OWA and possibility FM). Further, soft max, 1.(b), returns a membership of 0.5 to the max index, and the possibility measure, 1.(f), has full membership for its respective index. Interestingly, both of the Sugeno λ-FMs, one being subadditive, 1.(g), and the other superadditive, 1.(h), have relatively high possibility memberships, which indicate it is acting in accordance to a possibility FM despite the disparity between the two densities.

4.2 Experiment 2: Learned and Density Derived FMs

The baseline experiments presented in Table 2 are put forth to demonstrate the behavior of our indices with respect to learned and density derived FMs. For the case of the QP, 100 random samples were produced, labels were generated using a known FM (which varies per experiment), and the QP was then used to approximate the FM from that data (input plus known labels).

Table 2. Results of proposed indices for learned and density derived FMs for $N = 3$

	Method used to obtain FM		Max	Min	Avg	OWA	Poss
(a)	QP fit to data from soft max FM	dist.	0.07	0.35	0.3	0	0.2
	OWA weights of (0.7, 0.2, 0.1)	memb.	0.5	0	0	1	0.8
(b)	QP fit to 0.9 of OWA and 0.1 noise[1]	dist.	0.14	0.26	0.1	0	0.11
	OWA used was $[0.52, 0.08, 0.4]$	memb.	0.07	0	0.51	0.99	0.88
(c)	Sugeno λ-FM using densities	dist.	0.11	0.3	0.19	0	0.29
	learned in Table 2.b	memb.	0.19	0	0.01	0.99	0.7
(d)	QP fit to random (but valid) FM	dist.	0.2	0.23	0.18	0.21	0.36
	FM of $(0.5, 0.1, 0.11, 0.51, 0.79, 0.89, 1)$	memb.	0	0	0.02	0.78	0.64
(e)	QP fit to valid FM	dist.	0.24	0.14	0.16	0	0
	FM of $(0.33, 0.33, 0.33, 0.33, 0.33, 0.33, 1)$	memb.	0	0.14	0.1	1	1

The first experiment, shown in Table 2.(a), shows that the indices are still performing as expected when the FMs are learned using a QP. More interestingly, however, is comparing experiments 2.(b)-(c) with 1.(e). Memberships for all indices in 2.(b), are relatively close to the "calibrated" (expected) results computed in 1.(e) even with a slight amount of noise added to the OWA. However, 2.(c), the Sugeno λ-FM derived using the densities found in 2.(b), does not return memberships consistent with that found in 2.(b), which is undesired if we recognize 2.(b) as being the optimal FM for that data. In fact, the Sugeno

[1] In the case of noise, 90% of the label is from the OWA and 10% is random noise from the interval $[0, 1]$.

λ-FM appears to make the CFI act a bit more like a max operator. Additionally, as seen in 1.(g)-(h), the Sugeno λ-FM continues to have a high membership to behaving in accordance to possibility FMs. The last two experiments, 2.(d)-(e), show that the indices are able to characterize the behavior of FMs picked at random (but checked to ensure monotonicity) in a manner that is expected.

These preliminary results provide a proof of concept for the indices put forth herein. From these experiments it has been shown that the indices do return characterizations of the FM that reflect what one may expect to see for a given FM. It is important to note that a thorough investigation with respect to the experimental application of these indices is needed and planned as future work to further develop this FM characterization tool. This set of preliminary results gives a glimpse at the utility and need for such a tool; however, further analysis needs to be done. For example, a more intuitive and defined method for creating the fuzzy sets needs to be introduced, as these dictate the membership degree. Though there are areas of these indices that can likely be refined, this initial set of CFI characterization tools begins a movement towards filling a void in the understanding of what the CFI is really doing and how it is behaving.

5 Summary

In this paper, we put forth a set of new indices (distance formulas) and fuzzy sets to help address the question of what the CFI is doing for the case of learned or density derived FMs. The proposed tools have practical application in terms of gaining a deeper understanding into a given problem, guiding new learning methods (i.e., learning a FM that is a possibility measure, additive measure, etc.) and evaluating the benefit of using the CFI. Herein, we defined a few initial indices and we presented preliminary results for the behavior of these indices for the cases of a number of known FMs (and thus aggregation operators), simple learned (using the QP), and density derived FMs. While the results are preliminary, we believe they show the promise of such an approach to formally characterize the FM and CFI. In future work we will improve upon the definitions and further refine (calibrate) the fuzzy sets. In addition, we will then look to (experimentally) use these indices to help discover interesting and unknown behaviors of FMs and CFIs for the bigger purpose of guiding new theorems to help us better understand the aggregation behavior of mutli-sensor fusion systems.

References

1. Grabisch, M., Nguyen, H.T., Walker, E.A.: Fundamentals of uncertainty calculi with applications to fuzzy inference. Kluwer Academic Publishers (1994)
2. Anderson, D., Keller, J., Havens, T.: Learning fuzzy-valued fuzzy measures for the fuzzy-valued sugeno fuzzy integral. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. LNCS, vol. 6178, pp. 502–511. Springer, Heidelberg (2010)
3. Keller, J.M., Osborn, J.: Training the fuzzy integral. International Journal of Approximate Reasoning 15(1), 1–24 (1996)

4. Mendez-Vazquez, A., Gader, P., Keller, J., Chamberlin, K.: Minimum classification error training for choquet integrals with applications to landmine detection. IEEE Transactions on Fuzzy Systems 16(1), 225–238 (2008)
5. Sugeno, M.: Theory of Fuzzy Integrals and Its Applications. Ph.D. thesis. Tokyo Institute of Technology (1974)
6. Keller, J.M., Auephanwiriyakul, S., Gader, P.D.: New fuzzy set tools to aid in predictive sensor fusion. In: Proc. SPIE, vol. 4038, pp. 1497–1507 (2000)
7. Keller, J.M., Auephanwiriyakul, S., Gader, P.D.: Experiments in predictive sensor fusion. In: Proc. SPIE 4394, pp. 1047–1058 (2001)
8. Grabisch, M., Roubens, M.: Application of the choquet integral in multicriteria decision making. In: Fuzzy Measures and Integrals, pp. 348–374. Physica Verlag (2000)
9. Keller, J.M., Gader, P., Tahani, H., Chiang, J.H., Mohamed, M.: Advances in fuzzy integration for pattern recognition. Fuzzy Sets and Systems 65(23), 273–283 (1994)

Artificial Neural Network Modeling of Slaughterhouse Wastewater Removal of COD and TSS by Electrocoagulation

Dante A. Hernández-Ramírez[1], Enrique J. Herrera-López[1,⋆], Adrián López Rivera[2], and Jorge Del Real-Olvera[1,⋆]

[1] Centro de Investigación y Asistencia en Tecnología y Diseño del Estado de Jalisco A.C. Normalistas No. 800, Colinas de la Normal, CP. 44270 Guadalajara, México
[2] Facultad de Ingeniería Química y Ambiental, Universidad Veracruzana, Circuito Gonzalo Aguirre Beltrán s/n, Zona Universitaria, 91040, Xalapa, México
iqdahr@live.com.mx, {jdelreal,eherrera}@ciatej.mx
http://www.ciatej.net.mx/?lang=en

Abstract. A methodology for modeling the electrocoagulation of wastewater from the food industres, with high organic loads is proposed. The approach used is a nonlinear model based on Artificial Neural Networks (ANN), which is able to understand the interaction between the variables that define the process, to complement the traditional design of experiments. Where the interaction of variables determines in many cases, a large number of experiments to perform, that involve stages such as planning, organization and execution of experimental activities, also characterization and analysis of wastewater in order to remove chemical oxygen demand (COD) and total dissolved solids (TSS). From this approach it will be possible to find appropriate conditions for these parameters in order to enhance the contaminant removal process with specific routes (experimental conditions).

Keywords: Artificial Neural Networks, electrocoagulation, COD, TSS, wastewater, slaugtherhouse.

1 Introduction

Water is essential for life. In a growing world population water is needed to satisfy the necessity of drinkable water, for irrigating agriculture fields and for the industry. Residual substances directly derived from industries pollute the water. Strategies to overcome challenges in developing and adapting the technology for protection, conservation and recovery of this natural resource are currently under investigation. Residual liquids are substances at different concentrations generated as products or subproducts from industrial production processes. It is essential to treat industrial residual water before it is discharged to the sewers. This action must be taken since residual water has polluting action, which varies

⋆ Corresponding authors.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*, Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_26,

according to concentrations of pollutants. Wastewater is classified according to the type of industry that generates it.

The industries are classified into five groups:

- Consumer industries (textiles, tanning and laundry),
- Food production industries: canned foods, dairy products, fermented and distilled beverages, meat products and poultry, animal husbandry, sugar, pharmaceuticals, etc.
- Industries of materials: paper, photographic products, steel, foundry, oil fields and refineries, rubber, production of adhesives and automotive industry.
- Chemical industry: acids, detergents, starch, explosives, pesticides, phosphates and phosphorus, formaldehyde, plastics and resins. authors of the paper.
- Energy industries: power plants and nuclear power plants.

Water treatment is important into achieve water availability for man and agriculture, therefore quality standars must be fullfilled in order to assure water free of pollutants, under this approach diverse equipments and facilities have been developed to treat wastewater [1]. Electrocoagulation is an emerging technology used since 1906. During the last two decades it has been reported different applications of electrocoagulation such as: removing oil dispersed particles [2], grease and oil in the treatment of wastewater from electroplating processes, textile and water treatment processes [3], among others. New approaches to optimize the electrocaogulation process are required. A new interesting tool could be the artificial intelligence. Artificial Neural Networks (ANN) are tools of artificial intelligence intended to imitate the complex operation of organizing and processing information of the neurons in the brain. ANNs can identify patterns that correlate strongly a set of data which correspond to a class by a learning process, in which interconnected neuron weight are used to store knowledge about specific features identified within data. Then in this manuscript we use an ANN to model the electrocoagulation process in order to find strategies to optimize this process.

2 Background

One way to quantify the effectiveness of a water treatment process is based on the percentage removal of various parameters such as chemical oxygen demand (COD) and the Total Suspended Solids (TSS). The chemical oxygen demand (COD) is a parameter used to determine easily biodegradable matter. Is based on the use of a strong oxidant capable of oxidizing the non-biodegradable material. The amount of oxidant consumed is proportional to the amount of rustymatter [4]. The Total Dissolved Solids (TDS) are a measurement of the organic and inorganic substances, in molecular form, ionized or micro-granular, liquid containing,

in our case, water. To be considered TDS, substances should be small enough to pass a sieve or filter with size of two microns. If organic and inorganic substances cannot pass by a two-micron filtration and are indefinitely suspended or dissolved are called TSS (Total Suspended Solids). TSS is an indicator of the characteristics of water and the presence of chemical contaminants, that is, chemical composition and concentration of salts [5]. We can then define electrocoagulation as a process in which particles are destabilized contaminants that are suspended, emulsified or dissolved in an aqueous medium, inducing electric current in the water through parallel metal plates of different materials, being iron and aluminum the most widely used (Fig. 1) [6–8]. Electrocoagulation has been used in the treatment of wastewater from the food industry [9–12], these waters are characterized by high levels of COD and TSS besides high percentages of fats. The electrocoagulation process is affected by different factors. Among the most important are the nature and concentration of the contaminants, the wastewater pH and conductivity.

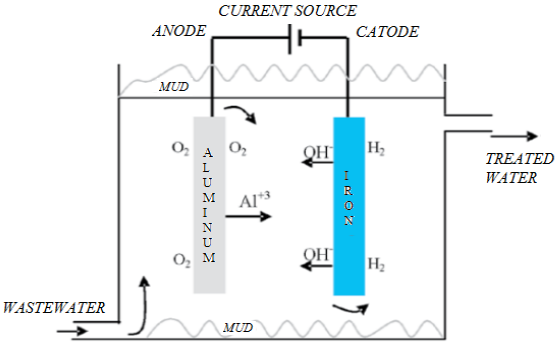


Fig. 1. Electrocoagulation system with aluminum anode and cathode of iron [13]

The electrocoagulation traditional modeling is based on statistical models from simple linear regression to determine the influence of any of the parameters (principals components analysis, ANOVA) and physicochemical behavior modeling (molecular interactions) [14]. These factors determine and control the reactions occurring in the system and the formation of coagulant. Based on this, it is necessary to define new methodologies for obtaining the maximum advantage of the contaminant removal system of the food industries. It must be recognized that this task is a joint effort that required optimization concepts in order to achieve succesful results with the fewest experiments, such as planning and organization of experimental activities, characterization and analysis through experimental designs, and proper integration through a computer system analysis and control of the results.

For this, we propose as an alternative the use of Artificial Neural Networks to understand and model the process, in turn to find appropriate parameters

and conditions (process route) to enhance the removal of contaminants. The advantage of using artificial neural networks is that they allow to model nonlinear correlations between the study variables, in addition to finding problems where there are adjustments missing data. Recent studies [15, 16] have been used to model synthetic waters removing contaminants present therein. This study is based on modeling the behavior of the electrocoagulation water coming from real industries. Therefore, an ANN was trained to predict conditions for modeling removal of COD and TSS of a slaughterhouse wastewater. From the variables measured in the laboratory the reponce of the electrocoagulation process a nonlinear relationship was used to predict the output of the system.

3 Methodology

We apply the methodological approach for the design and implementation of a nonlinear model based on soft computing technologies (ANN), which consists of 4 phases:

- Collection and generalization of data, selection of input and output signals, normalization of data.
- Construction of the topology of the ANN.
- Validation and Testing of artificial neural network (backpropagation algorithm).
- Comparison and discussion of the results generated by the ANN.

It is noteworthy to mention that our methodological approach emphasizes design topologies, which are calibrated via activation functions in the hidden layer and output layer, and later compared between the various topologies. The purpose is to obtain different nonlinear neural network models in order to analyze the behavior of each of them, with the overall behavior and thus retain the network model that has the best performance, the error calculation- RMS (according to the lowest RMSE - Root Mean Square Error). We were provided of a database of experimental results from which experiments were created for prospective shemes, and raises training of RNA for analysis. The summary of the selected variables from the databases for this modeling are summarized in Table 1. It has been found that working with parallel systems, while reducing the execution time of the RNA, generates more noise in the data. As handled two neural networks, one that considers output to% COD removal, and another that considers to% TSS removal. The entire data set was normalized in values between 0 and 1 simultaneously.

4 Results

Several topologies were studied for the amount of data to be used (60 series). A neural network with n-variables in the input, allows us to select two hidden

Table 1. Summary of Experiments

	Scheme 1	Scheme 2
Input variables	Concentration of wastewater: 100%, 70%, 50%, 30% pH: 5, 7, 9. Current Intensity: The intensity of current sought to keep values between 1 and 2 Amperes/m2.	Concentration of wastewater: 100% 70%, 50%, 30% pH: 5, 7, 9 Current Intensity: The intensity of current sought to keep values between 1 and 2 Amperes/m2. Time (min): 0, 15, 30, 45, and 60.
Output variables	COD% removal, experimental (mg/L). TSS% removal, experimental (mg/L).	COD% removal, experimental (mg/L). TSS% removal, experimental (mg/L).

Table 2. Summary of Experiments

Experiment 1	Experiment 2	Experiment 3	Experiment 4
Scheme 1	Scheme 1	Scheme 2	Scheme 2
Output 1: % Removal COD	Output 2: % Removal TSS	Output 1: % Removal COD	Output 2: % Removal TSS

Table 3. Comparison of RMSE

Experiment	ANN RMSE	Correlation between Real and Predicted Values
1	0.103148907	0.92307919
2	0.210019623	0.845899987
3	0.075617117	0.969321091
4	0.138088355	0.94450795

layers of 10 neurons each layer, an output layer of one neuron, and a function of sigmoidal activation. All data are considered representative so that the whole data is used to train the network, but the last 15 values will be considered for appropriate validation. The modeling is performed in Matlab^{TM} neural network simulator The experiments described are summarized in Table 2. In each of the experiments we used the backpropagation algorithm to train the ANN. The RMSE was analyzed for the data set used to obtain the results shown in Table 3. Is easy to see that experiment 3 responds best to the prediction of data compared to the other (from RMSE estimate of ANN and the tendency of the coefficient of correlation between actual and predicted normalized by ANN proposal). As noted Experiment 4 also yields good results, which helps us to understand that the most representative variables are from the scheme two. Figs. 2-5 show the results of the training of the networks (Table 3) and the desnormalization of data that show the results of training and validation of modeling.

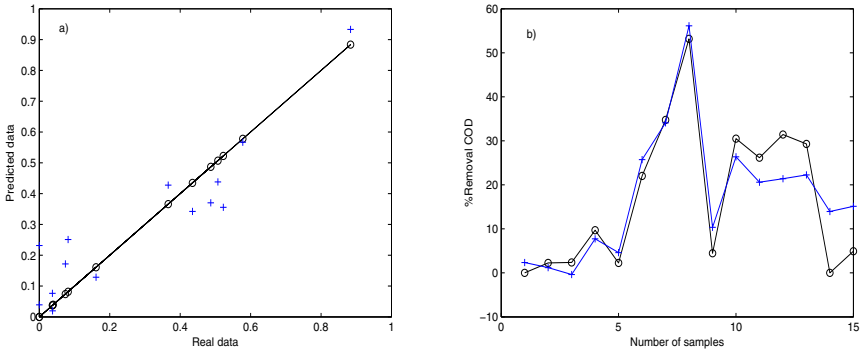


Fig. 2. Experiment 1 and Scheme 1- Output 1 (COD), a) correlation between the Real (o) and Predicted (+) values, b) behavior of the output of the ANN and the Experimental values

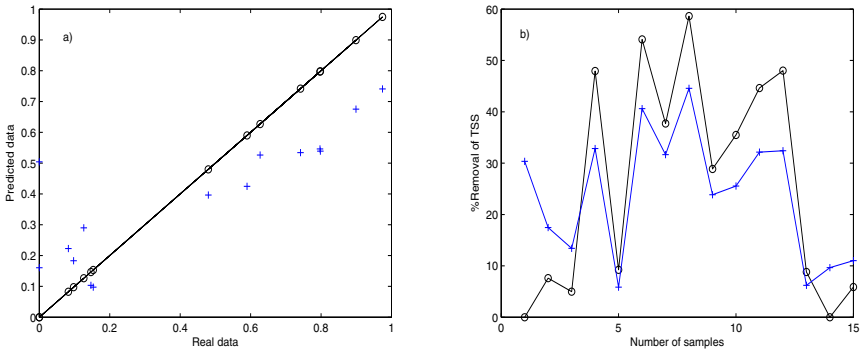


Fig. 3. Experiment 2 and Scheme 1- Output 2 (TSS), a) correlation between the Real (o) and Predicted (+) values, b) behavior of the output of the ANN and the Experimental values

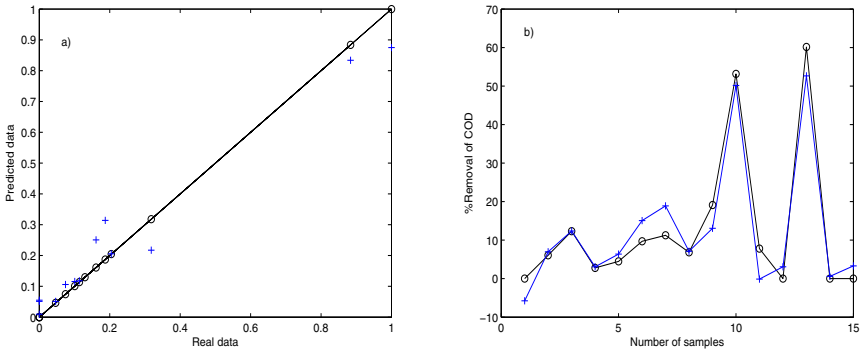


Fig. 4. Experiment 3 and Scheme 2- Output 1 (COD), a) correlation between the Real (o) and Predicted (+) values, b) behavior of the output of the ANN and the Experimental values

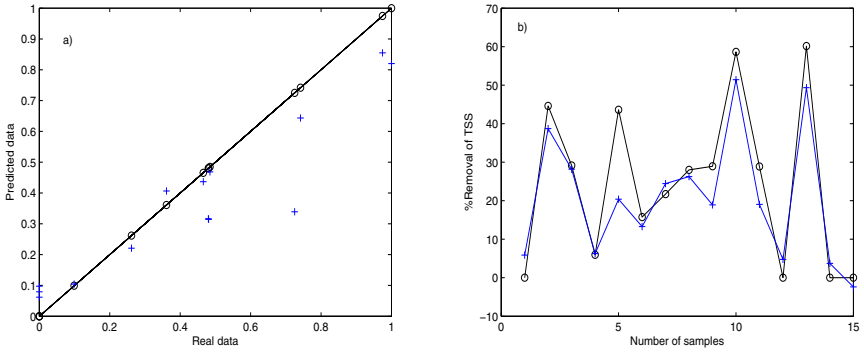


Fig. 5. Experiment 4 and Scheme 2- Output 2 (TSS), a) correlation between the Real (o) and Predicted (+) values, b) behavior of the output of the ANN and the Experimental values

5 Conclusion

From the results it is concluded that the ANN was capable to predict the COD and TSS removal. The Artificial Neural Network was able to understand the nonlinear relationship between these variables, which are considered independent quantized values so far, and the interaction of these variables to carry out the experiments, optimal results could allow removal of the pollutants. Therefore, we suggest testing different scenarios for possible future results.

Acknowledgments. The authors would like to thank the CONACYT for the support granted to the project 128894. We especially thank to anonymous referees for their helpful comments that have helped to improve the quality of this paper

References

1. Orozco, J.A.: La densidad de carga del electrolito como parámetro de control del proceso de electrocoagulación. Revista Ainsa 5, 3–30 (1985)
2. Petterson, J.W.: Industrial Wastewater Treatment Technology. Butterworth Publishers, Stoneham (1985)
3. Holt, P., Barton, G., Mitchell, C.: Electrocoagulation as a Wastewater Treatment. In: Proccedings The Third Annual Australian Environmental Engineering Research Event, Castlemaine, Victoria (1999)
4. Ramalho, R.S.: Tratamiento de Aguas Residuales. Ed. Reverté, S.A. España (1991)
5. Matcalf, Eddy: Wastewater Engineering, Treatment and Reuse. McGraw-Hill, New York (2003)
6. Holt, P.K., Barton, G.W., Mitchell, C.A.: The future for electrocoagulation as a localized water treatment technology. Chemosphere 59, 355–367 (2005)

7. Rajeshwar, K., Ibañez, J.: Environmental electrochemistry: Fundamentals and applications in pollution abatement. Academic Press limited, San Diego (1977)
8. Chen, G.: Electrochemical technologies in wastewater treatment. Sep. Purif. Technol. 38, 11–41 (2004)
9. Wiblbrett, G.: Limpieza y desinfección en la industria alimentaria. Acribia, Zaragoza (2000)
10. Centro de Actividad Regional para la Producción Limpia (CAR/PL). Prevención de la contaminación en la industria láctea. Secretaría de Ambiente y Desarrollo Sustentable. Buenos Aires (2002)
11. Chile, Comisin Nacional del Medio Ambiente regin Metropolitana. Guía para el control y prevención de la contaminación industrial. Fabricación de productos lácteos. Comisión Nacional del Medio Ambiente-Región Metropolitana. Santiago de Chile (1998)
12. Aymerich, S.: Conceptos para el tratamiento de residuos lácteos, pp. 1–19. Centro Nacional de Producción Más Limpia, Costa Rica (2000)
13. Restrepo, A.P., Arango, A., Garcés, L.F.: La Electrocoagulación: retos y oportunidades en el tratamiento de aguas. Producción Limpia 1, 58–77 (2006)
14. Hernández, L.P.S.: Investigacóin sobre procesos avanzados de tratamiento y depuración de aguas mediante electrocoagulación. PhD dissertation, E.T.S.I. Caminos, Canales y Puertos (UPM) (June 2011)
15. Ciprian-George, P., Renata, F., Marius, P., Silvia, C.: Neural Networks Based Models Applied to an Electrocoagulation Process. Environ. Eng. Manag. J. 10, 375–380 (2011)
16. Afshin, M., Hiua, D., Loghman, A., Leila, A., Anise, I.: Dye Removal Probing by Electrocoagulation Process: Modeling by MLR and ANN Methods. J. Chem. Soc. Pak. 34, 1056–1069 (2012)

Memetic Algorithm for Solving the Task of Providing Group Anonymity

Oleg Chertov and Dan Tavrov

Applied Mathematics Department
National Technical University of Ukraine "Kyiv Polytechnic Institute"
Kyiv, Ukraine
{chertov,dan.tavrov}@i.ua

Abstract. Modern information technologies enable us to analyze great amounts of primary non-aggregated data. Publishing them increases threats of disclosing sensitive information. To protect information about a single person, one needs to provide individual data anonymity. Providing group data anonymity presupposes protecting intrinsic data features, properties, and distributions. Methods for providing group anonymity need to protect the underlying data distribution, and also to ensure sufficient data utility after their transformation. In our opinion, the latter task is a problem which can be solved using only exhaustive search, therefore heuristic procedures need to be developed to find suboptimal solutions.

Evolutionary algorithms are heuristic guided random search techniques mimicking biological evolution by natural selection. They are inherently stochastic, which turns out to be a downside when converging to an optimum. Memetic algorithms are a combination of evolutionary algorithms and local search procedures. Applying local search increases convergence and enhances algorithm performance by incorporating problem specific knowledge.

In the paper, we introduce a memetic algorithm for providing group anonymity. We illustrate its application by solving a real data based problem of protecting military personnel regional distribution.

1 Introduction

According to the latest research conducted by the International Data Corporation [1], about 30% of digital information in the world need protection, and this number will rise to roughly 40% by 2020. In particular, it is necessary to protect publicly available results of various statistical surveys.

In this paper, by protecting statistical data we understand providing their individual (group) anonymity, i.e. [2] the property of individual respondents (groups of respondents) to be unidentifiable within a statistical dataset. Providing group anonymity implies concealing sensitive data distribution features, and also preserving sufficient level of their utility [3]. Existing methods solve this task in two stages. First, the data distribution is modified in order to mask its sensitive features. Then, the dataset is adjusted to fit the modified distribution.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_27,

Preserving data utility is a complex optimization task with no feasible algorithms developed for solving it (at least, known to the authors) other than exhaustive search. This task needs to be solved using heuristic approaches. Evolutionary algorithms are heuristic guided random search techniques that mimic biological evolution by natural selection [4, p. 15]. Memetic algorithms (MAs) were introduced in [5]. They mimic cultural evolution at the level of *memes* [6, p. 192]. Usually, MAs are implemented [7, p. 49] as evolutionary algorithms combined with local search procedures. Applying local search [4, p. 174] enables us to enhance the algorithm efficiency at solving certain tasks by utilizing problem specific knowledge, and to increase algorithm convergence. In this paper, we propose to apply MAs to solving the task of providing group anonymity not only at its second stage, but at the first one as well.

The rest of the paper is organized as follows. In Section 2, we discuss the notion of group anonymity. In Section 3, we outline the general MA structure and discuss its components. In Section 4, we illustrate MA by solving a real data based task of protecting regional distribution of military personnel working in the state of Massachusetts, the U.S. In Section 5, we draw conclusions and sketch possible directions of research to come.

2 Group Anonymity Basics

2.1 Generic Scheme of Providing Group Anonymity

Let the data about respondents be collected in a depersonalized *microfile* $\boldsymbol{M}$ with records r_i, $i = \overline{1,\rho}$, containing values of attributes w_j, $j = \overline{1,\eta}$. Let $\boldsymbol{w}_j$ denote the set of all the attribute w_j values.

Let us denote by $\boldsymbol{P} = \{P_1, \ldots, P_{l_p}\}$ the subset of $\boldsymbol{w}_p$. We will call w_p the *parameter attribute*, and we will also call P_i, $i = \overline{1,l_p}$, *parameter values*. Parameter values enable us to split initial microfile $\boldsymbol{M}$ into *submicrofiles* $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_{l_p}$.

Let us denote by $\boldsymbol{V} = \{V_1, \ldots, V_{l_v}\}$ the subset of the Cartesian product $\boldsymbol{w}_{v_1} \times \ldots \times \boldsymbol{w}_{v_t}$, where v_j are integers, $\forall j = \overline{1,t}$. We will call $\boldsymbol{w}_{v_j}$, $j = \overline{1,t}$, *vital attributes*, and we will also call V_i, $i = \overline{1,l_v}$, *vital value combinations*.

We will denote by $G(\boldsymbol{V},\boldsymbol{P})$ a *group*, i.e., a set of vital value combinations and parameter values. The group defines categories of respondents whose distribution needs protection.

The *task of providing group anonymity* (TPGA) lies [8] in modifying the microfile $\boldsymbol{M}$ for each G_i, $i = \overline{1,k}$, so that sensitive data features become confided. The generic scheme of providing group anonymity is as follows:

1. Construct a (depersonalized) microfile $\boldsymbol{M}$ of statistical data to be processed.
2. Define one or several groups $G_i(\boldsymbol{V}_i,\boldsymbol{P}_i)$, $i = \overline{1,k}$, representing categories of respondents to be protected.
3. For each i from 1 to k:
   - choose a *goal representation* $\Omega(\boldsymbol{M},G_i)$ which defines a dataset of arbitrary structure representing features of G_i in a way appropriate for providing group anonymity;

- perform data mapping using a *goal mapping function* $\Upsilon : \boldsymbol{M} \rightarrow \Omega_i$ $(\boldsymbol{M}, G_i)$;
- obtain a *modified goal representation* using a *modifying functional* $\Xi : \Omega_i (\boldsymbol{M}, G_i) \rightarrow \Omega_i^* (\boldsymbol{M}, G_i)$;
- obtain a *modified microfile* using an *inverse goal mapping function* $\Upsilon^{-1} : \Omega_i^* (\boldsymbol{M}, G_i) \rightarrow \boldsymbol{M}^*$.

4. Prepare the modified microfile for publishing.

The first three operations of step 3 comprise the *first stage* of solving the TPGA. The last operation of step 3 is the *second stage*.

We will deal with the most widely used goal representation, the *quantity signal* $\boldsymbol{q} = \left(q_1, \ldots, q_{l_p}\right)$ [9]. In the simplest case, q_i are quantities of respondents in submicrofiles $\boldsymbol{M}_i$, $i = \overline{1, l_p}$, whose vital attribute values belong to $\boldsymbol{V}$.

2.2 Minimizing Microfile Distortion While Performing Its Modification

Microfile modification boils down to altering the values of the parameter attribute p for certain records belonging to G in order to obtain the needed distribution.

To ensure that the overall number of records with a particular value of the parameter attribute p remains the same, the records should be altered in pairs. This procedure can be interpreted as a mutual swapping of records between different submicrofiles, which should be continued until the needed goal signal is obtained. In general, the choice of pairs of records to be swapped can be arbitrary. The only restriction is that one of them has to belong to group G, and the other one does not have to. However, in such case the needed distribution of vital attribute values can be achieved at the cost of distorting the distribution of the other attribute values potentially important for further statistical research.

In general, all the metrics for estimating microfile utility loss (information loss measures [10,11]) can be divided into metrics for ordinal (continuous) data and metrics for categorical (nominal) data.

For the ordinal data, standard deviation (1), absolute deviation (2), or mean variance (3) [10, p. 107] [11, p. 184] are usually applied:

$$\frac{1}{\rho n_{ord}} \sum_{i=1}^{\rho} \sum_{p=1}^{n_{ord}} \left(z_{ip} - z_{ip}^*\right)^2 \quad , \tag{1}$$

$$\frac{1}{\rho n_{ord}} \sum_{i=1}^{\rho} \sum_{p=1}^{n_{ord}} \left|z_{ip} - z_{ip}^*\right| \quad , \tag{2}$$

$$\frac{1}{\rho n_{ord}} \sum_{i=1}^{\rho} \sum_{p=1}^{n_{ord}} \frac{\left|z_{ip} - z_{ip}^*\right|}{z_{ip}} \quad , \tag{3}$$

where z_{ip}, z_{ip}^* are the values of the pth ordinal attribute of the ith record in initial and modified microfiles, respectively, n_{ord} is the number of ordinal attributes.

Among all the metrics presented above, only the last one takes into account that different attribute values can vary in magnitude. However, even it fails to provide adequate results if the values in the initial microfile (but not the modified one) approach zero. A suitable metric was proposed in [12] to deal with this situation:

$$\frac{1}{\rho n_{ord}} \sum_{i=1}^{\rho} \sum_{p=1}^{n_{ord}} \frac{\left| z_{ip} - z_{ip}^* \right|}{\sqrt{2} S_p} \quad , \tag{4}$$

where S_p is the standard deviation of the pth ordinal attribute values in the microfile.

It is impossible to directly apply (1–4) to categorical data. Several metrics have been proposed to deal with such data [10, p. 107–110]. For practical applications, it is primarily recommended to use metrics which lie in direct comparison of categorical values, since such metrics are always easy to interpret.

By *influential attributes* we will understand those ones whose distribution over parameter values is of interest for microfile data analysis.

Then, the minimal microfile distortion can be achieved through swapping the values of the parameter attribute between records close to each other with respect to their influential attribute values. We will call such an operation *swapping of the records between submicrofiles*. For instance, to provide group anonymity for a regional distribution of respondents with the certain occupation, one needs to swap the place of working between certain records. At the same time, to maximally ensure that the microfile data utility is preserved, the records to be swapped should be close, or even identical, with respect to their marital status, age, income level, and so forth.

In this paper, to quantitatively determine how close two records are, we will use the *influential metric* [13]:

$$\mathrm{InfM}\left(r, r^*\right) = \sum_{p=1}^{n_{ord}} \omega_p \left(\frac{r\left(I_p\right) - r^*\left(I_p\right)}{r\left(I_p\right) + r^*\left(I_p\right)} \right)^2 + \sum_{k=1}^{n_{nom}} \gamma_k \chi^2 \left(r\left(J_k\right), r^*\left(J_k\right) \right) \quad , \tag{5}$$

where I_p stands for the pth ordinal influential attribute (their overall number is n_{ord}), J_k stands for the kth categorical influential attribute (their overall number is n_{nom}), $r\left(\cdot\right)$ denotes the operator yielding the specified attribute value of the record r, $\chi\left(v_1, v_2\right)$ denotes the operator which is equal to χ_1 if values v_1 and v_2 fall into one category, and χ_2 otherwise, ω_p and γ_k are nonnegative weighting coefficients to be chosen according to the importance of certain attributes (the more improtant the attribute, the greater the coefficient).

Unlike the metrics introduced above, the influential metric (5):

1. Enables us to obtain the quantitative measure for the ordinal as well as the categorical attributes.
2. Takes into account only the influential attributes, i.e., those ones important for the subsequent microfile data analysis.
3. Enables us to quantitatively determine (using coefficients ω_p and γ_k) each attribute's importance.

It is virtually impossible to examine all the record pairs (r, r^*) to choose those ones which yield minimal value of (5). Thus, it is necessary to come up with heuristic approaches for choosing record pairs (r, r^*) that yield results acceptable from both the computational complexity and minimal value of (5) points of view.

2.3 Single-Stage Approach to Solving the Task of Providing Group Anonymity

As was mentioned in Section 2.1, the typical TPGA is usually solved in two stages.

At the first stage, the quantity signal is modified in order to conceal its sensitive features, such as [13, p. 77] its extremums, statistical features, trends, and others. However, in many cases it is necessary to perform such modification of the signal that its other important characteristics are preserved, or at least altered insignificantly.

Several techniques for modifying quantity signals described in the literature aim at preserving particular signal characteristics. E.g., the normalizing technique described in [14] enables us to conceal signal extremums with simultaneous preservation of such statistical features as the mean value and the standard deviation. Applying wavelet transforms [15] also makes it possible to mask signal extreme values, but it preserves signal wavelet details in the process. With the help of singular spectrum analysis [16], one can successfully conceal signal trend and preserve its cyclic and periodic components at the same time.

At the second stage of solving the TPGA, the microfile is adjusted in order to correspond to the modified quantity signal. As was stated in the previous section, such modification in ideal case needs to guarantee that the overall amount of distortion introduced in the microfile and expressed by (5) is minimal. Since it is impossible to achieve in practice, heuristic approaches need to be developed that modify the microfile within feasible time limits and guarantee that the distortion is sufficiently small.

Two main approaches have been proposed in the literature that meet these two requirements. Heuristic strategies introduced in [17] enable us to modify the microfile in a relatively short time, altering less than several percent of all the records in it. Memetic algorithm fully described in [18] outperforms heuristic strategies in finding better solutions of the task, however, it takes much more time to accomplish that.

Often, preserving data utility at the second stage (i.e., distorting the microfile as little as possible) is more important than at the first one (i.e., preserving quantity signal features). In such cases, no specific requirements are imposed on the modified quantity signal as the result of the first stage of solving the TPGA. It only needs to suffice as a solution to the TPGA. In this paper, we will consider the modified signal the solution to the TPGA if all the extremums are masked.

As was mentioned in the previous section, the amount of distortion being introduced into the microfile heavily depends on the particular choice of record pairs to be swapped between different submicrofiles. However, it is obvious that the overall amount of distortion depends not only on the heuristic approach

applied at the second stage, but also on the modified quantity signal itself. Some modified signals might lead to introducing smaller amount of distortion than the other ones.

The task of choosing such modified signal at the first stage of solving the TPGA that yields minimal microfile distortion at the second stage seems to be the one which can be solved only by exhaustive search. Therefore, in the section that follows, we propose to use memetic algorithms for solving the task of providing group anonymity by not only choosing what pairs of records to swap at the second stage, but also by choosing how many pairs should be swapped, and between which submicrofiles. In other words, we propose a method for solving the TPGA in only one stage, simultaneously searching for a modified signal and particular record pairs to be swapped in order to adjust microfile in accordance with this signal.

We call such an approach a *single-stage approach to solving the TPGA*.

3 Memetic Algorithm Structure

3.1 General Algorithm Structure

An MA for modifying microfile $\boldsymbol{M}$ implies carrying out the following steps:

1. Randomly generate initial population P of μ individuals, apply to each of them local search operator S.
2. Calculate fitness function $f(\boldsymbol{x})$ for each individual $\boldsymbol{x} \in P$.
3. Check termination condition. It if holds, stop, otherwise, go to 4.
4. Select λ pairs of parents.
5. Apply recombination operator R to each parent pair.
6. Apply mutation operator M to each of the offspring. Put resulting individuals into P'.
7. Apply local search operator S to each individual $\boldsymbol{x} \in P'$.
8. Calculate fitness function $f(\boldsymbol{x})$ for each individual $\boldsymbol{x} \in P'$.
9. Select μ fittest individuals from $P \cup P'$ and put them into P in place of the current ones.
10. Go to 3.

3.2 Individual Representation

Each individual in P represents a particular solution of the TPGA. It is a matrix $U = \|u\|_{Q\times 4}$. Elements of the first column u_{i1}, $\forall i = \overline{1,Q}$, stand for the indices of the microfiles from which the records belonging to the group G should be removed. They are usually defined by the user before solving the task. Elements of the third column u_{i3}, $\forall i = \overline{1,Q}$, stand for the indices of the microfiles to which the records belonging to G should be added. They, too, are specified by the user beforehand. Elements of the second column u_{i2} define the indices of the records from the submicrofiles $\boldsymbol{M}_{u_{i1}}$ to be removed, elements of the fourth column u_{i4}

define the indices of the records from the submicrofiles $\boldsymbol{M}_{u_{i3}}$ to be swapped with those ones defined by u_{i2}.

Each U row defines a pair of records from different submicrofiles to be swapped. Therefore, each individual U uniquely defines the modified quantity signal $\boldsymbol{q}^*$ (total number of occurrences of particular submicrofile indices in the columns 1 and 3), and also determines the particular way of obtaining it (each row defines the records to be swapped). Each individual U can have different number of rows.

Two restrictions are imposed on each individual U:

- total number of occurrences of a microfile index i in the first column of U cannot exceed q_i;
- each pair $\langle u_{i1}, u_{i2}\rangle$ or $\langle u_{i3}, u_{i4}\rangle$ $\forall i = \overline{1, Q}$ has to occur in U only once.

3.3 Fitness Function

The fitness function should represent [4, p. 19] a heuristic estimation of solution quality. In the case of the TPGA, the solution quality depends on two factors. First, the cost of obtaining the modified quantity signal q^* defined by the individual U in terms of (5) should be as minimal as possible. Second, the modified signal itself should suffice as a solution, i.e., extremums in $\boldsymbol{q}$ should be masked.

The first condition can be incorporated in the fitness function in a straightforward manner. The latter condition can be interpreted as a constraint on the TPGA, so it can be incorporated in the form of a penalty function [19]. Moreover, we introduce a penalty term to prevent individuals from growing indefinitely:

$$f(U) = C_{max} - \left(\sum_{i=1}^{Q} \mathrm{InfM}\left(\boldsymbol{M}_{u_{i1}}(u_{i2}), \boldsymbol{M}_{u_{i3}}(u_{i4})\right) + \Phi(\boldsymbol{\phi}, \boldsymbol{a}) + \Psi(Q) \right) , \tag{6}$$

where $\boldsymbol{M}(i)$ is the operator yielding the ith record of $\boldsymbol{M}$, Q is the number of rows in U, Φ stands for the penalty term associated with the quality of $\boldsymbol{q}^*$, $\boldsymbol{a}$ is the vector with the indices of the submicrofiles present in the first column of U, $\boldsymbol{\phi}$ is the vector with the weighting coefficients associated with such submicrofiles, Ψ stands for the penalty term associated with preventing individuals from growing indefinitely, C_{max} is a great positive constant needed to ensure that $f(U) > 0$ $\forall U$.

We propose to use the following penalty term associated with the quality of $\boldsymbol{q}^*$:

$$\Phi(\boldsymbol{\phi}, \boldsymbol{a}) = \sum_{i \in \boldsymbol{a}} \phi_i q_i^* \ . \tag{7}$$

3.4 Variation Operators

Recombination operator $R(U_{p1}, U_{p2})$ is applied with a high probability p_c to two parent individuals U_{p1} and U_{p2} with Q_{p1} and Q_{p2} rows, respectively, and returns two offspring individuals U_{o1} and U_{o2} with Q_{o1} and Q_{o2} rows, respectively.

We propose to use the operator reminiscent to both "cut" and "splice" operators introduced in [20]. It works by independently generating two *crossover points* k_{p1} and k_{p2} (random numbers in ranges $[0, Q_{p1}]$ and $[0, Q_{p2}]$, respectively), splitting both parents at appropriate points, and creating the offspring by exchanging the tails. In this case, $Q_{o1} = k_{p1} + Q_{p2} - k_{p2}$, $Q_{o2} = k_{p2} + Q_{p1} - k_{p1}$.

Mutation operator $M(U)$ is applied to the individual U with Q rows, and returns the mutated one U' with the same number of rows. We propose to view M as a superposition of four operators $M = M_4 \circ M_3 \circ M_2 \circ M_1$:

- M_1 is applied with a small probability p_{m_1} to the first U column as to the permutation; moving element u_{i1}, $i \in \{1, 2, \ldots, Q\}$, around involves moving element u_{i2} accordingly;
- M_2 is applied with a small probability p_{m_2} to the third U column as to the permutation; moving element u_{i3}, $i \in \{1, 2, \ldots, Q\}$, around involves moving element u_{i4} accordingly;
- M_3 is applied with a small probability p_{m_3} to the second U column as to the vector of categorical integer values;
- M_4 is applied with a small probability p_{m_4} to the fourth U column as to the vector of categorical integer values.

It is important to ensure that both restrictions imposed on the individuals are not violated as the result of applying variation operators.

3.5 Local Search Operator

Local search operator $S(U)$ is applied to the individual U with Q rows, and returns the modified one U' with the same number of rows. We propose to perform local search by carrying out the following steps:

1. For each U row i, $i = \overline{1, Q}$, carry out steps 2–4.
2. Generate a random number $r \in [0, 1]$.
3. If $r \leq p_{mem}$ ($r > p_{mem}$), assign to u_{i4} (u_{i2}) the index of a record from $\boldsymbol{M}_{u_{i3}}$ ($\boldsymbol{M}_{u_{i1}}$) closest to the record u_{i2} (u_{i4}) from $\boldsymbol{M}_{u_{i1}}$ ($\boldsymbol{M}_{u_{i3}}$) in terms of (5).
4. Go to 2.

Parameter p_{mem} should be great enough to ensure effective local search. However, it shouldn't approach unity to prevent premature algorithm convergence.

It is important to ensure that both restrictions imposed on the individuals are not violated as the result of applying local search operator.

3.6 Other Algorithm Parameters

Both *parent* and *survivor selection* can be chosen according to considerations other than those imposed by the individual representation. We propose to use tournament selection [21] as an easy-to-implement and efficient operator which does not require any global knowledge of the population [4, p. 63].

The population can be *initialized* by randomly generating matrices with different numbers of rows. It seems reasonable to generate elements u_{i1} with probabilities proportional to the values of the corresponding elements of $\boldsymbol{q}$.

The choice of such algorithm parameters as population size μ, crossover probability p_c etc. should be made individually for each task being solved.

4 Practical Results

Let us consider the task of masking the regional distribution of military personnel working in the state of Massachusetts, the U.S. We will use the 5-Percent Public Use Microdata Sample File corresponding to the 2000 U.S. census of population and housing [22]. The total of 141 838 records was taken for analysis.

We took "Place of Work PUMA" (PUMA standing for "Public Use Microdata Area") as the parameter attribute. Its values correspond to the codes of statistical areas of Massachusetts. We took each 10th value in the range 25010–25120 as the parameter values.

We took "Military Service" as the vital attribute. Its value "1" stands for "Active Duty," so it was taken as the only vital value.

A corresponding quantity signal is presented in Fig. 1 (solid line). Each signal element $1, 2, \ldots, 12$ corresponds to appropriate statistical area, with 1 corresponding to 25010, 2 corresponding to 25020 etc.

As we can see, there are extreme signal values in its elements 2, 7, 9, and 12. The records belonging to the group of military personnel need to be removed from them.

To minimize the overall influential metric (5), we took attributes "Sex," "Age," "Hispanic or Latino Origin," "Marital Status," "Educational Attainment," "Citizenship Status," and "Person's Total Income in 1999" as the influential ones. For the sake of simplicity, we considered all the influential attributes to be categorical, with the necessary parameters from (5) defined the following way: $\gamma_k = 1 \ \forall k = \overline{1,7}$, $\chi_1 = 1$, $\chi_2 = 0$. The metric thus defined shows the number of attribute values to be altered in order to provide group anonymity in the microfile.

We used the fitness function (6) in the following form:

$$f(U) = 1099 - \left(\sum_{i=1}^{Q} \sum_{k=1}^{7} \left(\operatorname{sign} \left| \operatorname{InfM} \left(\boldsymbol{M}_{u_{i1}} (u_{i2}, A_k) - \boldsymbol{M}_{u_{i3}} (u_{i4}, A_k) \right| \right) \right) + 1.2 \cdot q_2^* + q_7^* + q_9^* + q_{12}^* + \frac{4Q}{50} - 3 \right) ,$$

where $\boldsymbol{M}(i, j)$ yields the value of the jth attribute of the ith record in $\boldsymbol{M}$, A_i stands for the ith influential attribute, $i = \overline{1,7}$, $\operatorname{sign}(\cdot)$ stands for the function yielding 1 (-1) if its argument is positive (negative), or 0 if it equals 0, Q is the number of rows in U, $\boldsymbol{q}^*$ is the modified quantity signal obtained after performing the swaps defined by U, the penalty term $\frac{4Q}{50} - 3$ heavily discriminates against individuals with over 100 rows.

We chose the swap mutation [23] to be mutation operators M_1 and M_2, and the random resetting mutation [4, p. 43] to be mutation operators M_3 and M_4. The tournament size in the tournament selection was chosen to be 5. Other algorithm parameters were defined as follows: $\mu = 100$, $\lambda = 40$, $p_c = 1$, $p_{m_1} = p_{m_2} = p_{m_3} = p_{m_4} = 0.001$, $p_{mem} = 0.75$. We applied linear fitness scaling in the form presented in [24, p. 79] to prevent premature convergence.

We performed 30 independent runs of the MA. In each run, we terminated after having obtained 1000 generations. Among 3000 individuals from the last generations of each run, 2114 ones (or 70.47%) correspond to perfectly valid solutions of the TPGA. Some of the best solutions in terms of the overall metric (5) are presented in Fig. 1. The mean metric over all the solutions is 61.97, i.e., we need to alter only 0.04% of the attribute values in order to provide group anonymity.

To compare the results obtained using the single-stage approach at solving the TPGA, we also displayed on Fig. 1 the solution obtained in a conventional way and described in [18]. At the first stage, wavelet transforms were used, and at the second stage, both heuristic strategies and memetic algorithm were applied to minimize microfile distortion. The overall value of (5) after applying the best of heuristic strategies was 59, and the overall value of (5) after applying MA was 57.

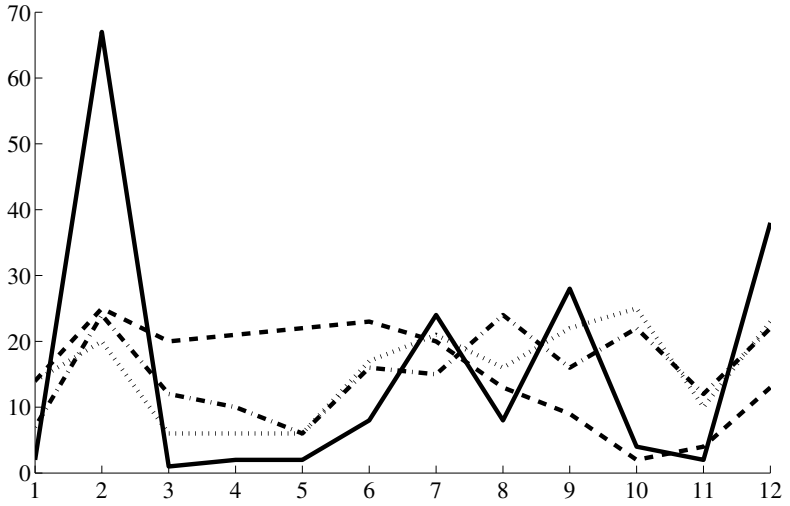


Fig. 1. Initial (solid line) and several modified quantity signals: the one with the metric 43 (dotted line), the one with the metric 49 (dashed-dotted line), the one obtained in [18] (dashed line)

5 Conclusions and Future Research

In the paper, we introduced a novel memetic algorithm for solving the task of providing statistical data group anonymity, and discussed its main features. We also illustrated its applicability to practical tasks by solving a real data based problem of masking military personnel regional distribution for the state of Massachusetts, the U.S.

We consider the following directions to be promising for future research: enhancing the algorithm efficiency by choosing appropriate operators, analyzing its efficiency as dependent on its parameters, combining the algorithm with other heuristic procedures described in the literature on providing group anonymity.

References

1. Gantz, J., Reinsel, D.: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East (2012), http://www.emc.com/leadership/digital-universe/iview/executive-summary-a-universe-of.htm
2. Pfitzmann, A., Hansen, M.: A Terminology for Talking about Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management, Version v0.34 (2010), http://dud.inf.tu-dresden.de/Anon_Terminology.shtml
3. Chertov, O., Pilipyuk, A.: Statistical Disclosure Control Methods for Microdata. In: 2009 International Symposium on Computing, Communication, and Control (ISCCC 2009). Proc. of CSIT, vol. 1, pp. 339–343. IACSIT Press, Singapore (2011)
4. Eiben, A.E., Smith, J.E.: Introduction to Evolutionary Computing. Springer, Heidelberg (2007)
5. Moscato, P.: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Toward Memetic Algorithms. C3P Report 826: Caltech Concurrent Computation Program, pp. 33–48. Caltech, CA (1989)
6. Dawkins, R.: The Selfish Gene: 30th Anniversary Edition. Oxford University Press, Oxford (2006)
7. Neri, F., Cotta, C.: A Primer on Memetic Algorithms. In: Neri, F., Cotta, C., Moscato, P. (eds.) Handbook of Memetic Algorithms. SCI, vol. 379, pp. 43–52. Springer, Heidelberg (2012)
8. Chertov, O., Tavrov, D.: Data Group Anonymity: General Approach. International Journal of Computer Science and Information Security 8(7), 1–8 (2010)
9. Chertov, O., Tavrov, D.: Group Anonymity. In: Hüllermeier, E., Kruse, R., Hoffmann, F. (eds.) IPMU 2010. CCIS, vol. 81, pp. 592–601. Springer, Heidelberg (2010)
10. Domingo-Ferrer, J., Tora, V.: Disclosure Control Methods and Information Loss for Microdata, Confidentiality, Disclosure, and Data Access. In: Doyle, P., Lane, J.I., Theeuwes, J.J.M., Zayatz, L.V. (eds.) Theory and Practical Applications for Statistical Agencies, pp. 91–110. Elsevier, Amsterdam (2001)
11. Mateo-Sanz, J.M., Domingo-Ferrer, J., Sebé, F.: Probabilistic Information Loss Measures in Confidentiality Protection of Continuous Microdata. Data Mining and Knowledge Discovery 11, 181–193 (2005)

12. Yancey, W.E., Winkler, W.E., Creecy, R.H.: Disclosure Risk Assessment in Perturbative Microdata Protection. In: Domingo-Ferrer, J. (ed.) Inference Control in Statistical Databases. LNCS, vol. 2316, pp. 135–152. Springer, Heidelberg (2002)
13. Chertov, O. (ed.): Group Methods of Data Processing. Lulu.com, Raleigh (2010)
14. Liu, L., Wang, J., Zhang, J.: Wavelet-Based Data Perturbation for Simultaneous Privacy-Preserving and Statistics-Preserving. In: 2008 IEEE International Conference on Data Mining Workshops, Pisa, pp. 27–35. IEEE Computer Society Press (2008)
15. Chertov, O., Tavrov, D.: Providing Group Anonymity Using Wavelet Transform. In: MacKinnon, L.M. (ed.) BNCOD 2010. LNCS, vol. 6121, pp. 25–36. Springer, Heidelberg (2012)
16. Tavrov, D., Chertov, O.: SSA-Caterpillar in Group Anonymity. In: Paper presented at the World Conference on Soft Computing, May 23-26. San Francisco State University, San Francisco (2011)
17. Chertov, O.R.: Minimizatsiia spotvoren pry formuvanni mikrofailu z zamaskovanymy danymy (In Ukrainian). Visnyk Skhidnoukrainskoho Natsionalnoho Universytetu imeni Volodymyra Dalia 8(179), 256–262 (2012)
18. Chertov, O.R., Tavrov, D.Y.: Memetychnyi alhorytm dlia modyfikatsii mikrofailu z minimizatsiieiu spotvoren u protsesi zabezpechennia hrupovoi anonimnosti. Shtuchnyi Intelekt (in press, 2013) (in Ukrainian)
19. Smith, A.E., Coit, D.W.: Penalty Functions. In: Bäck, T., Fogel, D.B., Michalewicz, Z. (eds.) Evolutionary Computation 2. Advanced Algorithms and Operators, pp. 41–48. Institute of Physics Publishing, Bristol (2000)
20. Goldberg, D.E., Korb, B., Deb, K.: Messy Genetic Algorithms: Motivation, Analysis, and First Results. Complex Systems 3, 493–530 (1989)
21. Brindle, A.: Genetic Algorithms for Function Optimization. Doctoral dissertation and technical report TR81-2, University of Alberta, Department of Computer Science (1981)
22. U.S. Census 2000. 5-Percent Public Use Microdata Sample Files, `http://www.census.gov/main/www/cen2000.html`
23. Syswerda, G.: Schedule Optimization Using Genetic Algorithms. In: Davis, L. (ed.) Handbook of Genetic Algorithms, pp. 332–349. Van Nostrand Reinhold, New York (1991)
24. Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989)

Takagi-Sugeno Approximation of a Mamdani Fuzzy System

Barnabas Bede[1] and Imre J. Rudas[2]

[1] Department of Mathematics
DigiPen Institute of technology
9931 Willows Rd. NE
Redmond, WA 98052, USA
bbede@digipen.edu
[2] Óbuda University
Bécsi út 96/B
H-1034 Budapest, Hungary
rudas@uni-obuda.hu

Abstract. In the present paper we construct a higher order Takagi-Sugeno fuzzy system that approximates a Mamdani fuzzy system, with arbitrary accuracy. The goal of this construction is to reduce the computational complexity of a fuzzy systems considered, also to replace a nonlinear operator by an approximate linear operator. The proposed methodology is fully constructive, so it does not require training of the Takagi-Sugeno fuzzy system. The construction combines Takagi-Sugeno systems with the classical Lagrange interpolation.

1 Introduction

The most successful and the most wide-spread applications of fuzzy sets theory proposed by L. Zadeh in [14] are fuzzy controllers [5], [8], [10] [1]. There is an extensive literature on the applications of fuzzy controllers, while their theory receives recently less attention [7]. Of course for a successful development of the topic both from theoretical and practical points of view one needs deeper investigation of aspects related to fuzzy controllers.

A single input single output fuzzy system (fuzzy system for short) consists typically of a fuzzifier, an inference system constructed by taking into account a fuzzy rule base, and a defuzzifier. If the fuzzy system is coupled with a classical discrete or continuous dynamical system, then it becomes a fuzzy controller. We can say that the fuzzy core of a controller of this type is a fuzzy system.

Mamdani fuzzy systems [5] are historically the first fuzzy controllers and they have the property that they can be derived very easily from linguistic (fuzzy) rules. The disadvantage of a Mamdani system is its computational complexity, which in general involves two numerical integrations. Takagi-Sugeno systems gain speed over Mamdani systems, but there is a price to pay, namely the linguistic interpretation is often replaced by data-driven and learning based methodologies [3], [12] for their construction.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_28,

In the present paper we construct a Takagi-Sugeno fuzzy system that can approximate a given Mamdani fuzzy system with arbitrary accuracy. The construction is combining Takagi-Sugeno fuzzy systems with classical Lagrange interpolation. This method is inspired by combined approximation operators in [4], similar to the ideas in [1] and [2], where Takagi-Sugeno fuzzy systems and Taylor expansions were combined.

2 Preliminaries

Let us denote by $C[a,b]$ the set of continuous functions on the $[a,b]$ interval. The uniform norm of a function $f \in C[a,b]$ is defined as

$$\|f\| = \sup_{x\in[a,b]} f(x).$$

Definition 1. *[9] Let (X,d) be a compact metric space and $([0,\infty),|\cdot|)$ the metric space of positive reals endowed with the usual Euclidean distance. Let $f: X \to [0,\infty)$ be bounded. Then the function*

$$\omega(f,\cdot) : [0,\infty) \to [0,\infty),$$

defined by

$$\omega(f,\delta) = \bigvee \{|f(x) - f(y)| x,y \in X,\ d(x,y) \le \delta\}$$

is called the modulus of continuity of f.

Theorem 1. *The following properties hold true*

i)

$$|f(x) - f(y)| \le \omega(f, d(x,y))$$

for any $x,y \in X$;

ii) $\omega(f,\delta)$ is nondecreasing in δ;

iii) $\omega(f,0) = 0$;

iv)

$$\omega(f,\delta_1+\delta_2) \le \omega(f,\delta_1) + \omega(f,\delta_2)$$

for any $\delta_1,\delta_2, \in [0,\infty)$;

v)

$$\omega(f,n\delta) \le n\omega(f,\delta)$$

for any $\delta \in [0,\infty)$ and $n \in \mathbb{N}$;

vi)

$$\omega(f,\lambda\delta) \le (\lambda+1)\cdot\omega(f,\delta)$$

for any $\delta,\lambda \in [0,\infty)$;

vii) f is continuous if and only if

$$\lim_{\delta\to 0} \omega(f,\delta) = 0.$$

A fuzzy rule base consists of linguistic rules (fuzzy if-then rules) of the following form:

$$\text{If } x \text{ is } A_i \text{ then } y \text{ is } B_i, i = 1, ..., n.$$

We assume that A_i are the antecedents in our fuzzy rule base and they are fuzzy sets having continuous membership grades while consequences B_i are integrable, $, i = 1, ..., n$.

We consider a Mamdani fuzzy system with Center of Gravity defuzzification. For $x \in [a, b]$ a crisp input, the fuzzy output is calculated as

$$B'(y) = \bigvee_{i=1}^{n} A_i(x) \wedge B_i(y),$$

which is subject to defuzzification and the result is

$$COG(B') = \frac{\int_c^d B'(y) \cdot y \cdot dy}{\int_c^d B'(y) dy}.$$

Combining these two relations we can write a Mamdani fuzzy system as

$$M(x) = \frac{\int_c^d \bigvee_{i=1}^{n} (A_i(x) \wedge B_i(y)) \cdot y \cdot dy}{\int_c^d \bigvee_{i=1}^{n} (A_i(x) \wedge B_i(y)) \cdot dy}.$$

Since A_i is continuous, and B_i integrable, $i = 1, ..., n$ we have $M(x)$ well defined and continuous.

A Takagi-Sugeno fuzzy system is described by fuzzy rules with fuzzy sets as antecedents and numerical values as consequences. We consider higher order Takagi-Sugeno fuzzy systems in the present paper. Such a fuzzy system is described by the if-then rules $i = 1, ..., n$:

$$\text{if } x \text{ is } A_i \text{ then } y_i = a_{m,i}x^m + a_{m-1,i}x^{m-1} + ... + a_{0,i}.$$

To aggregate the individual outputs of the fuzzy rules we calculate

$$y = \frac{\sum_{i=1}^{n} A_i(x) \cdot y_i}{\sum_{i=1}^{n} A_i(x)}.$$

As a conclusion the output of a higher order Takagi-Sugeno fuzzy system is given as

$$TS(x) = \frac{\sum_{i=1}^{n} A_i(x) \cdot (a_{m,i}x^m + a_{m-1,i}x^{m-1} + ... + a_{0,i})}{\sum_{i=1}^{n} A_i(x)}.$$

3 Takagi-Sugeno Approximation of a Mamdani Fuzzy System

The problem proposed and solved in the present paper is the following.

Given a Mamdani system find a Takagi-Sugeno fuzzy system that approximates its output with arbitrary accuracy. The advantage of such an approximation is its low computational complexity compared to the Mamdani approach, which involves numerical integration. The idea is present in the literature in several applications, whenever the sum is used as an aggregation operator rather than max-min composition (based on compositional rule of inference) or min- $\rightarrow$ compositions (based on generalized modus ponens). In this way we would like to fill in the gap that exists between theoretical findings of fuzzy logic and practical applications of fuzzy systems.

Of course since Takagi-Sugeno fuzzy systems can approximate any continuous function with arbitrary accuracy [3], [6], [12], [7], the existence of such Takagi-Sugeno fuzzy systems is theoretically ensured. In the presented approach we adopt a constructive approach, avoiding the use of adaptive techniques which are not easily leading to theoretical results. Adaptive techniques can be used to further improve performance of the Takagi-Sugeno controllers that we construct if we have to use them in a certain application.

Let us start with the Mamdani system

$$M(x)=\frac{\int_c^d \bigvee_{i=1}^n (A_i(x)\wedge B_i(y))\cdot y\cdot dy}{\int_c^d \bigvee_{i=1}^n (A_i(x)\wedge B_i(y))\cdot dy}. \tag{1}$$

Let us consider that both the antecedents and consequences are normal, such that $A_i(x_i)=1$ and $B_i(y_i)=1$, $i=1,...,n$. Let us consider the knot sequence $x_{-1}<x_0<a=x_1<...<x_n=b<x_{n+1}<x_{n+2}$. Let us denote by $(A)_0$ the 0-level set of any fuzzy set A, which is by definition the closure of its support, i.e. $(A)_0=cl(supp(A))$. We consider

$$(A_i)_0\subseteq[x_{i-1},x_{i+1}]$$

and

$$(B_i)_0\subseteq[\min\{y_{i-1},y_i,y_{i+1}\},\min\{y_{i-1},y_i,y_{i+1}\}],$$

$i=0,...,n$, which means that there are exactly two overlapping antecedents in the rule bases that we consider. Also, let us consider $y_{-1},y_0,y_{n+1},y_{n+2}$ auxiliary knots for the consequence part, without a restriction on the ordering for the consequences. We define $p_i(x)$ as the (fourth-order) Lagrange interpolation polynomial for the data $(x_{i-2},y_{i-2}),(x_{i-1},y_{i-1})$ (x_i,y_i), $(x_{i+1},y_{i+1}),(x_{i+2},y_{i+2})$, $i=1,...,n$,

$$p_i(x)=\sum_{k=i-2}^{i+2} l_k(x)y_k, l_k(x)=\prod_{j=i-2,j\neq k}^{i+2}\frac{x-x_k}{x_j-x_k}. \tag{2}$$

The Takagi-Sugeno fuzzy system associated to these polynomials is

$$TS(x)=\frac{\sum_{i=1}^n A_i(x)\cdot p_i(x)}{\sum_{i=1}^n A_i(x)}. \tag{3}$$

The following theorem shows that the Takagi-Sugeno fuzzy system considered is an approximation of the Mamdani system considered above.

Theorem 2. *Let us consider $M(x)$ a Mamdani fuzzy system as given in (1) and $TS(x)$ be a Takagi-Sugeno fuzzy system as in (3) with consequences as in (2). Then the following estimate holds true*

$$\|M(x) - TS(x)\| \leq 3 \max_{j=1,..,n} \omega(p_j, \delta) \tag{4}$$

with $\delta = \max_{i=0,n+2} |x_i - x_{i-1}|$.

Proof. We obtain successively:

$$|M(x) - TS(x)| = \left| M(x) - \frac{\sum_{i=1}^n A_i(x) \cdot p_i(x)}{\sum_{i=1}^n A_i(x)} \right|$$

$$= \left| \frac{\sum_{i=1}^n A_i(x) M(x)}{\sum_{i=1}^n A_i(x)} - \frac{\sum_{i=1}^n A_i(x) \cdot p_i(x)}{\sum_{i=1}^n A_i(x)} \right|$$

$$\leq \frac{\sum_{i=1}^n A_i(x) |M(x) - p_i(x)|}{\sum_{i=1}^n A_i(x)}. \tag{5}$$

Since there are only two overlapping consequences, i.e., $x \in (A_j)_0 \cup (A_{j+1})_0$, we get

$$\|M(x) - TS(x)\| \leq \frac{\sum_{i=j}^{j+1} A_i(x) |M(x) - p_i(x)|}{\sum_{i=j}^{j+1} A_i(x)}$$

Now we estimate $|M(x) - p_k(x)|$, $k \in \{j, j+1\}$ as follows

$$|M(x) - p_k(x)| = \left| \frac{\int_c^d \bigvee_{i=1}^n (A_i(x) \wedge B_i(y)) \cdot y \cdot dy}{\int_c^d \bigvee_{i=1}^n (A_i(x) \wedge B_i(y)) \cdot dy} - p_k(x) \right|$$

$$= \left| \frac{\int_c^d \bigvee_{i=1}^n (A_i(x) \wedge B_i(y)) \cdot (y - p_k(x)) \cdot dy}{\int_c^d \bigvee_{i=1}^n (A_i(x) \wedge B_i(y)) \cdot dy} \right|.$$

We use now Lemma 7.10 in [1] and we get

$$|M(x) - p_k(x)| \leq \frac{\int_c^d \bigvee_{i=1}^n (A_i(x) \wedge B_i(y)) \cdot |y - p_k(x)| \cdot dy}{\int_c^d \bigvee_{i=1}^n (A_i(x) \wedge B_i(y)) \cdot dy}.$$

We will use again the fact that there are two overlapping rules, so there exist two consecutive non-zero terms in the above maximum. Then we obtain

$$|M(x) - p_k(x)| \leq \frac{\int_{(B_j)_0 \cup (B_{j+1})_0} \bigvee_{i=j}^{j+1} (A_i(x) \wedge B_i(y)) \cdot |y - p_k(x)| \cdot dy}{\int_{(B_j)_0 \cup (B_{j+1})_0} \bigvee_{i=j}^{j+1} (A_i(x) \wedge B_i(y)) \cdot dy} \tag{6}$$

with $x \in [x_{j-1}, x_{j+2}]$. For each $y \in (B_j)_0 \cap (B_{j+1})_0$ we have

$$y \in [\min_{i=j-1,\ldots,j+2} y_i, \max_{i=j-1,\ldots,j+2} y_i],$$

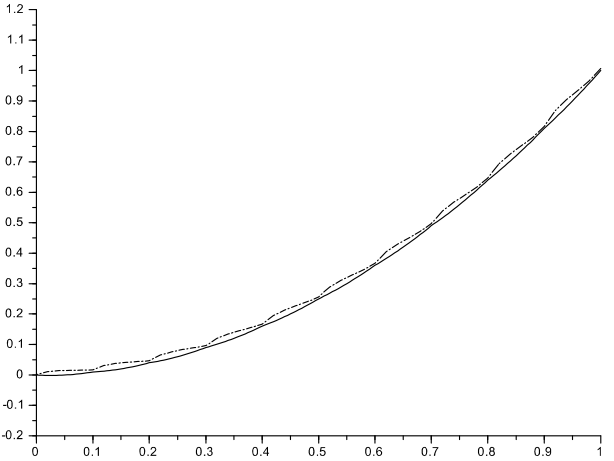


Fig. 1. Takagi-Sugeno fuzzy system (continuous line) that approximates a Mamdani fuzzy system (dash-dot line)

i.e.,

$$y \in [\min_{i=j-1,\ldots,j+2} p_k(x_i), \max_{i=j-1,\ldots,j+2} p_k(x_i)].$$

Since p_k, $k = j, j+1$ are continuous we obtain that there exists $z_j, z_{j+1} \in [x_{j-1}, x_{j+2}]$ with $p_j(z_j) = y$ and $p_{j+1}(z_{j+1}) = y$. We obtain

$$|y - p_j(x)| = |p_j(z_j) - p_j(x)| \leq \omega(p_j, |z_j - x|) \leq \omega(p_j, 3\delta),$$

where $\delta = \max_{i=0,..,n+2} |x_i - x_{i-1}|$. Similarly

$$|y - p_{j+1}(x)| = |p_{j+1}(z_{j+1}) - p_{j+1}(x)| \leq \omega(p_{j+1}, |z_{j+1} - x|) \leq \omega(p_{j+1}, 3\delta).$$

As a conclusion, from (6) finally we obtain

$$|M(x) - p_k(x)| \leq 3 \max_{k=j,j+1} \omega(p_k, \delta) \leq 3 \max_{j=1,..n} \omega(p_j, \delta),$$

relation which together with (5) leads to the estimate in the statement of the theorem, namely (4).

We consider in what follows an example. We consider a simple Mamdani fuzzy system that approximates different target functions, using 10 fuzzy rules. Its Takagi-Sugeno Approximation is constructed (see Figs 3, 3). We use triangular antecedents and consequences for the Mamdani system and triangular antecedents for the Takagi-Sugeno fuzzy system. The results confirm the theoretical findings of the paper.

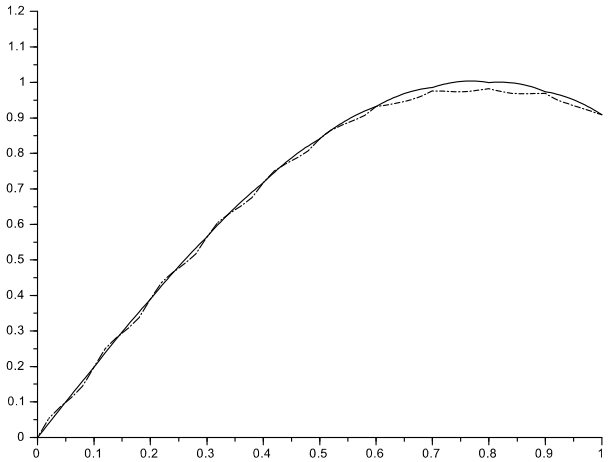


Fig. 2. Takagi-Sugeno fuzzy system (continuous line) that approximates a Mamdani fuzzy system (dash-dot line)

4 Conclusions and Further Research

As our conclusion let us formulate that we have proved that a Mamdani fuzzy system can be approximated by a Takagi-Sugeno fuzzy system in a fully constructive way. This open up new applications, where one can combine linguistic and data based rules, towards the development of new types of fuzzy systems.

References

1. Bede, B.: Mathematics of Fuzzy Sets and Fuzzy Logic. STUDFUZZ, vol. 295. Springer, Heidelberg (2013)
2. Bede, B., Rudas, I.J.: Approximation properties of higher order Takagi-Sugeno Fuzzy Systems. In: IEEE 2013 IFSA World Congress NAFIPS Annual Meeting Edmonton, Canada, pp. 1–6 (2013)
3. Kosko, B.: Fuzzy Systems as Universal Approximators. IEEE Transactions on Computers 43, 1329–1333 (1994)
4. Lucyna, R., Karolina, T.: Approximation by modified Favard operators. Commentationes Mathematicae 44, 205–215 (2004)
5. Mamdani, E.H., Assilian, S.: An experiment in linguistic synthesis with a fuzzy logic controller. J. Man Machine Stud. 7, 1–13 (1975)
6. Mitaim, S., Kosko, B.: The shape of fuzzy sets in adaptive function approximation. IEEE Transactions on Fuzzy Systems 9, 637–656 (2001)
7. Sonbol, A.H., Fadali, M.S.: TSK Fuzzy Function Approximators: Design and Accuracy Analysis. IEEE Transactions on Systems Man and Cybernetics 42, 702–712 (2012)
8. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man and Cybernetics 1, 116–132 (1985)

9. Stancu, D.D., Coman, G., Blaga, P.: Analiza numerica si teoria aproximarii, vol. II. University Press, Cluj-Napoca (2002) (in Romanian)
10. Sugeno, M.: An introductory survey of fuzzy control. Information Sciences 36, 59–83 (1985)
11. Tikk, D., Kóczy, L.T., Gedeon, T.D.: A survey on universal approximation and its limits in soft computing techniques. International Journal of Approximate Reasoning 33, 185–202 (2003)
12. Wang, L.X., Mendel, J.M.: Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. IEEE Transactions on Neural Networks 3, 807–814 (1992)
13. Ying, H.: General Takagi-Sugeno fuzzy systems with simplified linear rule consequent are universal controllers, models and filters. Information Sciences 108, 91–107 (1998)
14. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)

Alpha-Rooting Image Enhancement Using a Traditional Algorithm and Genetic Algorithm

Maryam Ezell, Azima Motaghi, and Mo Jamshidi

Department of Electrical Engineering
University of Texas at San Antonio
San Antonio, TX, 78249
{vfc907,dhr936}@my.utsa.edu, moj@wacong.org

Abstract. The application of soft computing in image/signal enhancement and comparing it with traditional methods will be discussed in this paper. This study presents two optimization methods for α-rooting image enhancement, which is a transform based method. The first method is a derivative-based optimization and the second one is Genetic Algorithm optimization. The parameter will be driven through optimization of measure of enhancement function (EME). The results from, the simulations show both methods are reliable; however, the first method has more computing cost.

Keywords: soft computing, genetic algorithm, alpha-rooting.

1 Introduction

Soft computing has been used to solve complex tasks such as pattern recognition, anomalies recognition and forecasting. The classical problem solving methods have been rapidly replaced with these new algorithms over the past decades. Artificial Neural Networks (ANN), Fuzzy Logic and Genetic Algorithms (GA) are the most popular methods in soft computing working based on human mind and genetic evolution [1]. Image enhancement composed of several techniques applied to the images, modify image perception.

There are two general methods for image enhancement; the first method is Spatial Domain Image Enhancement which directly manipulates the pixels of the images. Second is Frequency Domain methods. In this method, images are transferred into frequency domain, then a function is applied to modify their magnitudes, and finally the results are converted to time domain. In [2], Maini has proposed several image enhancement algorithms in spatial domain. Logarithmic transformations are applied in image enhancement where images are gray-scale and there are large ranges of values, the lower values are mapped into wider ranges in Logarithmic Transformation Methods. Powers-Law Transformations (Gamma Correction) is defined by $s = cr^{\gamma}$. This method is used for various values of γ in order to find the best performance. Histogram Processing is the most common spatial domain method, which improve image performance by spreading out higher bin intensity in the histogram.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_29,

It was proved that genetic algorithms are the most powerful unbiased optimization techniques for sampling a large solution space. Because of unbiased stochastic sampling, they were quickly adapted in image processing. They were applied for the image enhancement, segmentation, feature extraction, and classification, as well as the image generation. Genetic Algorithm is a heuristic that is routinely used to generate useful solutions to optimization and search problems [3]. Since in the α-rooting method the objective is to find the best parameter α which maximizes the EME function, this problem is considered an optimization problem and therefore Genetic Algorithm has been used as a solution. We will describe both methods in the following sections.

1.1 α-Rooting Image Enhancement

α-rooting filter is a Fourier Based Image Enhancement method. With α-rooting enhancement working in frequency domain is much easier. The procedure of transform based image enhancement has the following steps [4]:

- Transfer to frequency domain (DFT)
- Perform a function which modifies the magnitude (M)
- Transfer to the time domain

These steps are represented in figure 1.

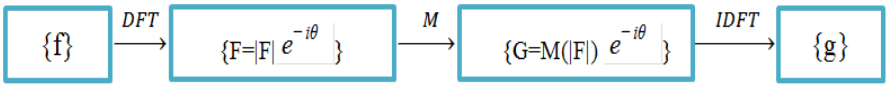


Fig. 1. Flowchart of the steps for image enhancement by α-rooting

Where f is the original image, M is a function which multiplies the transformed coefficients α, and g in the enhanced image. In this case M is define as

$$M(|F|) = |F|^{\propto}, \tag{1}$$

where the coefficient $\propto$ is in the range of [0, 1].

For measuring image enhancement we used the following equation which is explained in [2] :

$$EME_{\propto,k_1,k_2} = \frac{1}{k_1 k_2} \sum_{l=1}^{k_1} \sum_{k=1}^{k_2} 20\log\left(\frac{I^w_{max,k,l}}{I^w_{min,k,l}}\right) \tag{2}$$

where each image f(n,m) is divided to $k_1 \times k_2$ blocks w(n,m), α is the enhancement parameter, $I^w_{min,k,l}$ and , $I^w_{max,k,l}$ are the minimum and maximum intensity of each block. In this project, $k_1$and k_2 are constant and the objective is to find α that maximizes equation (2).

$$EME = \max\left(EME_{\propto,k_1,k_2}\right) \tag{3}$$

1.2 Genetic Algorithm

In this study, we mostly focus on image (contrast) enhancement of gray-scale images by applying new technology versus the classic one. The objective of this paper is to optimize α-rooting parameter which is a frequency based image enhancement method [4], [5]. Genetic Algorithm has been applied to calculate the required parameter. The proposed methodology has been implemented with MATLAB and MATLAB's GA toolbox.

The rest of the paper is organized as follows; first the paper describes the α-rooting algorithm in detail, which is a useful method in image enhancement. Then, the genetic algorithm used for enhancing the image will be explained. In the third section, the results acquired from both algorithms will be presented and compared. In the final section, we show our conclusion and make some discussions.

2 Methods

Genetic Algorithm is an artificial intelligence approach that is used to solve optimization problems. It mimics the metaphor of evolution of genes in the form of biological generation. The same as the evolutions of genes in nature, this process is not a directed, purposive process in this machine learning technique. The individual parameters that are to be optimized will be presented as a vector of random values at the beginning of the algorithm, which are similar to the base-4 chromosomes seen in living organisms' DNA. Each base of these chromosomes will then evolve to its final value through a random process.

Fig 2. shows the process of evolution in a simple GA optimization. It can be seen from the flowchart that this process contains the following steps [6]:

1. Start with generating random population of N chromosomes. N represents the number of parameters or chromosomes to be optimized.
2. Evaluate the fitness function $\phi(x)$ for each chromosome X in the population.
3. If the generated population does not satisfy the condition of fitness function, in reproduction parts, two parents (chromosomes) should be selected. The probability of selection of the parents increases for those chromosomes that have higher values for the fitness function.
4. Cross over the parents to produce new children.
5. Mutate new offspring at position n in the chromosome.
6. Replace newly generated population with old the one.
7. The new chromosome is generated now. The algorithm goes back to step 2.
8. When the created population satisfies the condition, the algorithm stops and the old population will be replaced with the new population as the output of the optimization.
9. If the end condition is satisfied, then stop.

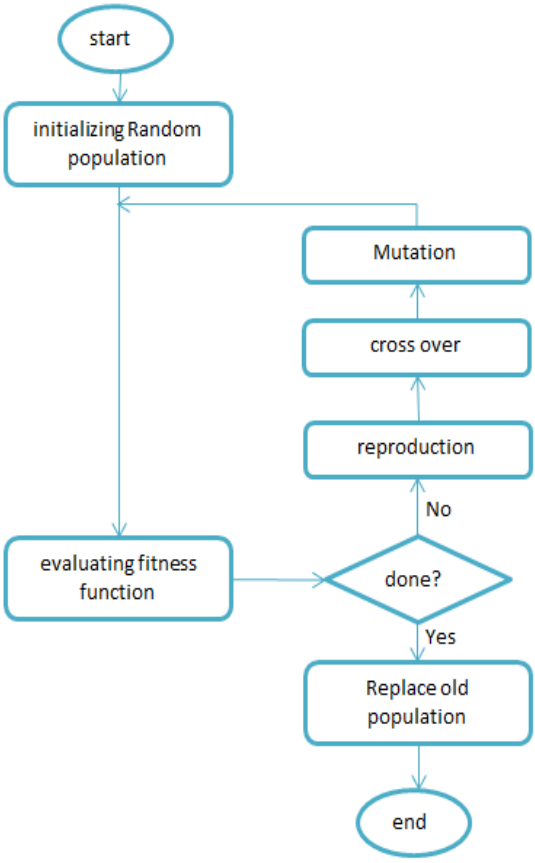


Fig. 2. Genetic algorithm flowchart

As mentioned earlier, α-rooting algorithm finds the value of alpha, which maximizes the value of EME. However, the algorithm must go through all the values of alpha to find this optimized value of α. In this research, GA is used to improve gray-scale image contrast by optimizing the parameter α. Therefore, the fitness function for the chromosome in GA (as alpha in only one parameter to be optimized) is used for the measure of enhancement as a fitness function.

3 Implementation and Results

In this part, we used the classical algorithm of α-rooting and GA to find α, which satisfied equations (2) and (3). The classical algorithm is based on sequence of points calculated by a deterministic computation. In each iteration the algorithm generates a single point [7]. The first method is using a sequence of points for α starting at 0.01

Table 1. Shows optimized parameter for both classical and GA algorithms. It also shows the number of iterations required by each method.

Methods	α	# of iterations
Classical algorithm (1st image)	0.7900	100
GA (1st image)	0.78199	37
Classical algorithm (2nd image)	0.7700	100
GA (2nd image)	0.77589	3
Classical algorithm (3rd image)	0.8600	100
GA (3rd image)	0.88204	4
Classical algorithm (4th image)	0.8800	100
GA (4th image)	0.93216	1

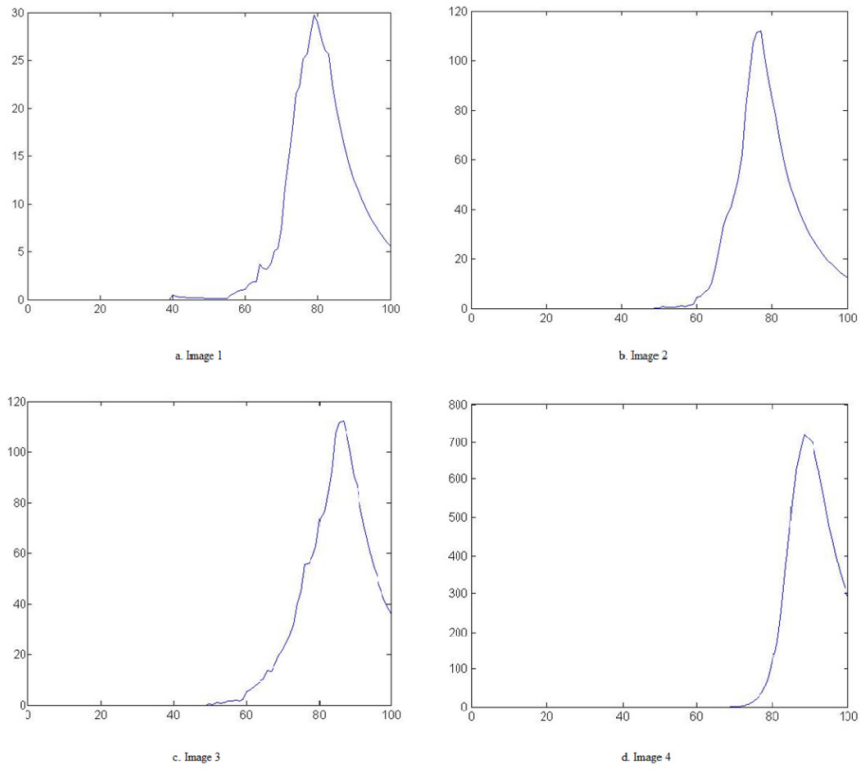


Fig. 3. The EME vs. α graph for 4 images

which increases at increments of 0.01. EME should be calculated for 100 iterations per image. In the second method, we use GA to optimize EME. α has been selected as a GA parameter and the inverse of EME is the fitness function. GA algorithm may find the best solution after a few generations.

Four black and white images were adopted from [8]. Both algorithms were used to find the parameter α. The Matlab GA toolbox was used to simulate the genetic algorithm. The sizes of the images vary between 359×460 to 1095×1692. The values of $k_1$and k_2 are 16 pixels. We added zeros to both sides of the images such that the image's pixels are divisible by $k_1$and k_2. Table 1 compares the results acquired from both models. It shows that both methods have achieved acceptable results while GA method requires less iteration to find the optimum α. This is usually equivalent to less computation time which shows GA has performed the task faster.

Fig 3. is a graphical representation of the value that maximizes the EME function for both methods. The results show that both methods perform well on the black and white images. However, it is worth mentioning that the genetic algorithm required fewer iterations to find the optimum value of α. For most of the images, the algorithm found the best value of α, with less than five iterations, while the classical α-routing algorithm should compute all the values in the range of α.

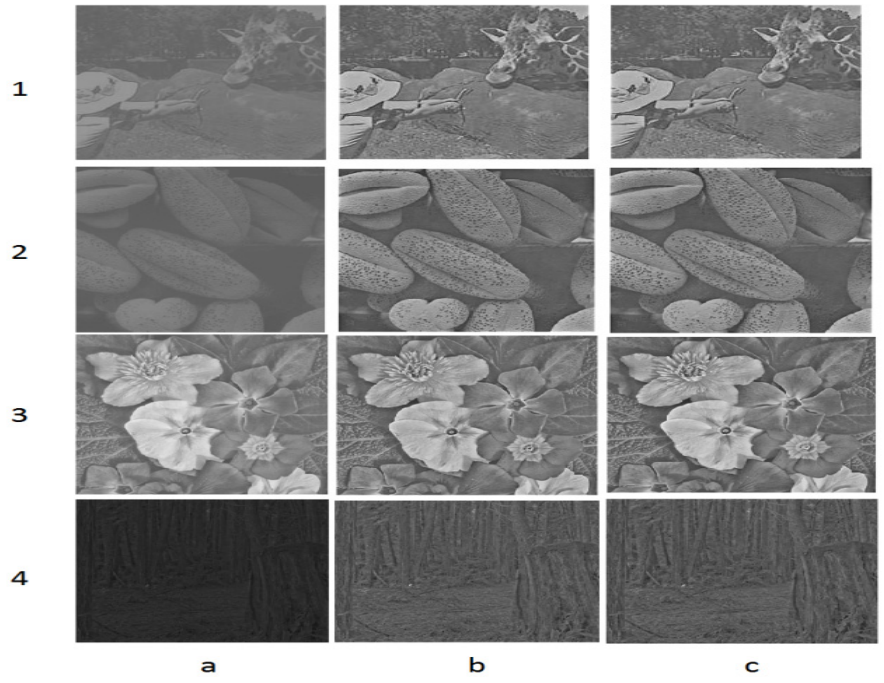


Fig. 4. a) Original image b) α-rooting enhanced image using GA c) α-rooting enhanced image using classical algorithm

4 Conclusion

In this paper, two methods have been used for contrast enhancement of gray-scale images using α-rooting. The first method is using an algorithm which increases α in increments of 0.01 (starting at α = 0.01) for 100 iterations per image to find the best α that maximizes the EME measure. The second method is using Matlab's GA toolbox to find the best α that minimizes the inverse of the EME measure. By comparing the results for four images we can see that both methods greatly improved the contrast of the images, but there is not a significant difference between the results generated by both methods. However, by using GA toolbox we get faster results, as it generates good results after a few generations, whereas the algorithm terminates after 100 iterations and then compares the results.

References

[1] Zadeh, L.A.: Fuzzy Logic, Neural Networks, and Soft Computing. Communication of the ACM 37(3), 77–84 (1994)
[2] Maini, R., Aggarwal, H.: A Comprehensive Review of Image Enhancement Techniques. Journal of Computing 2(3) (2010)
[3] Genetic Algorithm, http://en.wikipedia.org/wiki/Genetic_algorithm (July 7, 2013) (retrieved)
[4] Agaian, S.S., Panetta, K., Grigoryan, A.M.: Transform-Based Image Enhancement Algorithms with Performance Measure. IEEE Transactions on Image Processing 10(3), 367–382 (2001)
[5] Arslan, F.T., Grigoryan, A.: Fast splitting alpha-rooting method of image enhancement: tensor representation 15(11), 3375–3384 (2006)
[6] Paulinas, M., Ušinskas, A.: A Survey of Genetic Algorithms Applications for Image Enhancement and Segmentation. Information Technology and Control 36(3) (2007)
[7] Genetic Algorithm (2013), http://www.mathworks.com/discovery/genetic-algorithm.html (retrieved)
[8] Rafael, C., Gonzalez, R.E.: Digital Image Processing, 3rd edn., Upper Saddle River, NJ (2008)

Learning User's Characteristics in Collaborative Filtering through Genetic Algorithms: Some New Results

Oswaldo Velez-Langs[1] and Angelica De Antonio[2]

[1] Departamento de Ingenierías
Universidad de Bogotá, Jorge Tadeo Lozano
Carrera 4 # 22-61
Bogotá – Colombia
oswaldoe.velezl@utadeo.edu.co
[2] Facultad de Informática
Universidad Politécnica de Madrid
Campus de Montegancedo
Madrid-Spain
angelica@fi.upm.es

Abstract. This work presents an alternative approach (Genetic Algorithms approach) to traditional treatment of Recommender Systems (RSs). The work examines genetic algorithms possibilities to offer adaptive characteristics to these systems trough learning. The main goal, in addition to give a general view about RSs capabilities and possibilities, it is to provide a new example mechanism for to extend RSs learning capabilities (from user's personal characteristics), with the purpose of improve the effectiveness at time of to find recommendations and appropriate suggestions for particular individuals.

Keywords: Recommender Systems, Genetic Algorithms, User-Adapted Interaction.

1 Introduction

Usually in Internet, that is the main source of information for people that search answers and solutions for many situations, we use search engines typing key words, being too much the employed time for this task without finding the expected results. One alternative is to use Recommender Systems (RSs) [19, 26, 3] that offer to users an approach with their preferences, which are capable of suggesting the acquisition of any product. These systems filter the information, being classified in two main categories, depending on the information that use to suggest items. Those that only use information on the items and information with respect to objective user are call "Content Based" where the types, needs and inclinations of the users are determined in design time [9]. Alternative there are systems that do not use information on the items, but they do the suggestion using the known preferences of a group of users to predict the strangers preferences of a new user, the recommendations for this new user are based on these predictions [24]. This category is named "*Collaborative Filtering*" (CF).

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_30,

The users "collaborate" in the sense of any rating (evaluations on items) that put any user improve all the performance of the system [14]. The base of this technique is that a good interesting way to find content for a person is to find first other people that have similar interests and then to recommend items that those people prefer [6].

The Basic process is do a matching between the information that has about the profile of the active user[1] and the profiles of the other users that are already stored and of whose preferences has knowledge. The approach in this work describes a different alternative to traditional collaborative filtering, using Genetic Algorithms (GAs) like tuners in the process of profile matching, adapting finally these to individual preferences what impacts in a greater precision at moment of to predict that likes or not to one specific user. The paper is organized in the following way section 2 show a background related work, and related view to the collaborative filtering, CHC GA and *Learnable Evolutionary Model* (LEM) respectly, section 3 show the experiment framework of the proposal, section 4 show the method, in section 5 we can see some results and their analysis finally the section 6 talk about conclusions.

2 Related Work

Recommender Systems are alternative to search engines (like Google, Yahoo and others), instead of use keywords entered by the user, take information of a previous knowledge obtained from a group of users using the technique that is named Collaborative Filtering (CF). Rich [27] is considered like one of the first references in the topic. There is a long history of patents related to RS and CF, covering since [7] to [15]. In 1992 Goldberg et. al. coined the term of "Collaborative Filtering", in the context of a system for filtered of e-mails using binary flags [12]. Excellent studies and research can be found in [29, 18, 13].

Shardanand and Maes [29] designed a CF system for music (Ringo) and experienced with diverse number of distance measures among users, including Pearson's Correlation, restricted correlation and Cosine. They compare four recommendation algorithms based in the Medium Absolute Error (MAE) of predictions.

The e-Commerce system Amazon.com (http://www.amazon.com), requests to any user to qualify the articles recently bought. These qualifications are used as entrance for a recommendations engine helping to consumers to find others products that probably like. GroupLens is pioneer in the theme and carries many efforts in CF ([24, 18, 19, 14]). GroupLens team initially implemented CF systems based on neighbourhood to qualify articles of Usenet. They use a scale between 1 and 5, compute the distance using Pearson. Pennock and Horvitz [25] propose an axiomatic base for CF. CF does prediction of preferences, combining predilections of different users. Authors aim that the "addition" of preferences has been studied in the theory of Social Election from 1960 [1]. They maintain that only a simple model of more nearby neighbour satisfiers the axiomatic conditions. Delgado [9] uses an approach of

[1] The term "active user" refers to the people which the system works.

CF based in agents, developing some algorithms that combine the data of the qualifications with other sources of information such as geographical location of users. The especially heavy voting is used to combine the recommendations from different sources.

One approach named semi-automatic filtering that combine Genetic Algorithms and the called "learning form feedback" is presented in Sheth and Maes [30], this approach adapts of dynamic way to the changes of interest of the user, shown like one small population of filtering agents goes evolving to do selection of personalized Usenet news. In Min Tjoa et al [21], is presented a system called CIFS (Cognitive Information Filtering System) which applies an evolutionary model (Learning Classifier System) at the time of to learn from the user. CIFS filter e-mails, based in the ranking that do the user and in the checking of behaviour of him. CIFS use, with difference of the previous system, one Boolean recuperation framework in addition of vectorial. Additionally not only adapts to the user profiles but does a process of generation of the same. Amalthaea (Moukas and Maes [22]) is a multi-agent ecosystem for personalized filtering, which discovers and monitors information sites.

MyVU [11], is presented like a new generation of RSs, its focus is based on the behavior of consumption observed in the user and Interactive Evolutionary Computation. This system develops a management of the customer relations and marketing one-to-one in one virtual university domain. MyVU provides an adaptable Web interface for different members in one virtual university and presents recommendation routines for the places often utilized.

The work of Kim et al. [17] presents a novel recommender system that combines two methodologies, interactive evolutionary computation and content-based filtering method. Also, the proposed system applies clustering to increase the time efficiency. The system aims to effectively adapt and respond to immediate changes in users preference. Boumaza and Brun [4] present experiments and results on a standard benchmark data-set from the recommender system community that support the choice of the evolutionary approach and show that it leads to a high accuracy of recommendations and a high coverage, while dramatically reducing the size of the model (by 84%), also show that the evolutionary approach produces results able to generate accurate recommendations to unseen users, while easily allowing the insertion of new users in the system with little overhead.

The system presented in Salehi et al. [28] has two main modules. In the first module, weights of implicit or latent attributes of materials for learner are considered as chromosomes in genetic algorithm then this algorithm optimizes the weights according to historical rating. Then, recommendation is generated by Nearest Neighborhood Algorithm (NNA) using the optimized weight vectors implicit attributes that represent the opinions of learners. In the second, preference matrix (PM) is introduced that can model the interests of learner based on explicit attributes of learning materials in a multidimensional information model. Then, a new similarity measure between PMs is introduced and recommendations are generated by NNA. Finally Hu et al. [16] propose a generalized Cross Domain Triadic Factorization (CDTF) model over the triadic relation user-item-domain, which can better capture the interactions between domain-specific user factors and item factors. In particular,

they devise two CDTF algorithms to leverage user explicit and implicit feedbacks respectively, along with a genetic algorithm based weight parameters tuning algorithm to trade off influence among domains optimally

2.1 CHC Algorithm

CHC [10] is a genetic generational search algorithm that uses elitist selection. This technique introduces a balance between diversity and convergence. The CHC algorithm matches the parents randomly, but only to those pairs of strings which they defer one to each other in some number of bits (i.e., a crossing threshold) is allowed them to reproduce. The initial threshold is established $l/4$, where l is the string size. When there is no descent inserted in the new population during the elitist selection, the threshold is reduced in 1. The cross operator in CHC realizes a uniform crossing (HUX) and a random exchange in half of bits exactly that differ both between the parents. During the recombination phase, mutation in this type of algorithm is not applied. When descent in the population of a successive generation cannot be inserted and the crossing threshold has reached a value of 0, the CHC put new diversity inside the population through a restart form known as catastrophic mutation.

The catastrophic mutation uses the best individual in the population as a model to restart the population. The new population includes a string model's copy; the rest of the population is generated muting some bits percentage (i.e. 35%) in the string model. The next figure shows schema:

```
Procedure CHC
Begin
     t = 0
     d = L/4
     Initialize P (t)
     Evaluate Structures in P (t)
     While termination condition not satisfied do
     Begin
          t = t + 1
          Select_r C (t) from P (t - 1)
          Recombine Structures in C (t) forming C'(t)
          Evaluate Structures in C'(t)
          Select_s P (t) from C'(t) and P(t-1)
          If P (t) equals P (t-1)
               d - -
          If d < 0
     Begin
          Diverge P (t)
          d = r x (1.0 - r) x L
     End
     End
End
```

Fig. 1. CHC Algorithm

2.2 Learnable Evolutionary Model

LEM (Learnable Evolution Model) is similar to an ordinary GA in the sense of using a population of (usually bitstring) "chromosomes" whose fitnesses are measured, and

whose chromosomes compete to survive into the next generation, the model was proposed for Michalski in [20].

In a standard GA algorithm, the next generation of chromosomes is generated from the previous generation by eliminating the less fit from the population, by making copies of the fitter chromosomes, and then applying genetic operators, LEM uses a concept learning algorithm (called AQ, but specifically in this work we use one Decision Tree Algorithm like ID3) to find a classification rule which covers (matches) the fittest few chromosomes in each generation and does not cover the least fit chromosomes. The fittest few chromosomes are called the "*positive examples*", and the least fit are called the "*negative examples*". The aim of the learning algorithm is to find these classification rules. Once the classification rule which matches the +ve chromosomes and does not match the -ve chromosomes is found, it is used to generate other chromosomes which match the rule, that are added to the population. The new generation of chromosomes has its fitnesses measured and another GA/LEM loop begins.

The classification rule instantiates chromosomes that have high fitness values, because the rule itself was created from high fitness chromosomes. Hence the population fills with high fitness chromosomes, and faster than with blind genetic-operator GAs, because the classification rule steers the evolution into high fitness regions. The next figure shows schema:

```
1. initiate a random population of N chroms

2. while(evolution continues to occur)
       *  measure the fitness values of the population
       *  rank the chroms according to fitness
       *  select the +ve set of top fitness chroms from the popln
       *  select the -ve set of bottom fitness chroms from the popln
       *  use the AQ algorithm to find a classification rule that
          covers/matches the +ve chroms, but not the -ve chroms
       *  use the classification rule to instantiate new chroms
          that match the rule, to fill the population of N chroms
3. output the fittest chrom
```

Fig. 2. LEM Algortihm

3 Experimental Design

Now the frame of the experiment is illustrated, showing the data initially, the metric used and the tests on which we frame the carried out experiment.

3.1 Experimental Data and Sets

The data of *MovieLens* are used in the experiments which were collected by *GroupLens* (http://www.grouplens.org) of anonymous information which is composed

by 100.000 voting (ratings) of 943 users on 1.682 items in a scale of 1 to 5, each user has evaluated 20 items at least, in addition it has users' demographic information (age, gender, occupation). The prediction schema will generate estimations for existents evaluations in the data set. This is done in order to compare the algorithms effectiveness, using an error measurement between predictions and evaluation real values. The algorithms will use a data set when these evaluations are not present, guaranteeing a clean prediction process. In this way we can divide the data set in three groups:

- **Test set:** 9430 evaluations used in algorithm evaluation phase (pre-editions and error computing), which count with 10 items valued by a user chosen randomly.
- **Training set:** It counts with 90570 evaluations which are used in the Genetic Algorithms run and Pearson's Correlation. The training set is the only evaluations space that the algorithm knows.
- **Recommendation set:** It is the whole data set, which is composed by training and test sets.

3.2 Metrics

In order to have an idea of the quality of a prediction algorithm, there are three important dimensions that must be measured: coverage, efficiency and accuracy [7].

- The coverage is defined as a measurement of items percentage for which predictions can be made.
- The efficiency is a concept derived from IR and its measure is given by three main parameters: precision, recovery and relevance index.
- The accuracy measures how well the system was in the process of presenting items to the user.

The metrics to evaluate the precision of a prediction algorithm can be divided in two groups: precision's statistic metrics and decision support metrics [14].

3.2.1 Mean Absolute Error (MAE)

First using this metrics were Shardanand and Maes in [29]. This is defined as the absolute error average, computing as the difference between the rating given by the user and the prediction.

MAE is defined as:

$$MAE = \frac{1}{c}\sum_{j=1}^{c}\left|r_{i.j} - p_{i,j}\right| \tag{1}$$

Where c is the number of items that the user *i* has evaluated.

3.2.2 Normalized Absolute Error (NMAE)

Goldberg et Al [13] propose a Normalized Absolute Error (NMAE), which gives a precision measurement independent of the possible evaluations range.

$$NMAE = \frac{MAE}{|r_{\max} - r_{\min}|} \tag{2}$$

Other metrics in this group have been studied, such as mean squared error and the correlation between user ratings vector and predictions vector for the same items. However the conclusions which are obtained using any of these metrics are generally the same ones [13].

3.3 Tests

The tests will be made on the raised CF scheme, applying different variants in the implementation in order to compare their configurations on the *MovieLens* data set. The variants of the algorithm that we will use are: Standard Genetic Algorithm ([29, 33]), GA CHC ([34]), Learnable Evolutionary Model and Pearson's Algorithm.

3.4 Genetic Algorithm CHC

It is used in order to find a suitable weight configuration to the characteristics that compose the active user profile, having in mind the test set. In the implementation that we made, we determine two experimental variants in order to analyze and interpret the obtained results:

- **Variant 1:** The experimental variant is to use to users selected randomly which have at least one item in common with the active user in the experimentation phase, in prediction phase, without using a criterion to determine which of the selected users are in the active user's neighborhood but on the contrary including all of them in the neighborhood.
- **Variant 2:** During the experimentation process was observed that users selected randomly in training phase had, in most cases and in determined occasions all, a greater value to the determined threshold for the neighborhood. Therefore based on the threshold criterion, they should not be used these in next phases, reason why we decided to form the neighborhood set with all users of the DB. Proceeding to arrange them in ascending form according to the Euclidian Distance with the active user and only the top-N are selected (TOP-N).

3.5 Learnable Evolutionary Model

In this model we use one similar approach like in CHC case for experimental variants but involving the LEM in the proposal for tuning the process of profile matching. The LEM developed here use one Decision Tree algorithm (ID3) like learning component in place of AQ learning.

3.6 Pearson's Correlation

This algorithm only uses the votes (rating) that users emit on common items as criterion to determine the neighborhood degree between two users and build the neighborhood set in TOP-N form ordering DB users' neighborhood degree with the active user in descending form; it's different to the form in which is treated this problem by the Genetic Algorithm CHC which, in addition to this characteristic, it uses user information and characteristics of the evaluated item. The precision of each FC algorithm configuration is measured using the Mean Absolute Error (MAE) formula on made predictions, having in mind the test set. The results of the GA Standard, CHC and LEM are counteracted with the result of the Pearson's Correlation Algorithm.

4 Proposed Method

Our proposal is based on work made in [31] subsequent works like [33, 34] show the approaches GA standard and CHC respectively. Before initiating, the movies information must be processed in separated profiles, one for each person, defining the movie differences for that person.

The user profile j is denoted by $profile(j)$, and it's represented as an array of 22 values for the 22 characteristics considered from data set for each user. The profile has two parts: a variable part (the rating value, which changes according to the item that is considered in a given moment), and a fixed part (the others 21 values, which only are recovered once at beginning of the program). As the user j can have qualified many different movies, is defined $profile\,(i, j)$ as profile of the user j on the film i :

1 Rating	2 Age	3 Gender	4 Occupation...	22 18 Genre frequencies
5	**23**	**0**	**45**	**0000001000100000000**

Fig. 3. Profile(i,j). Profile of the User i on item j

The data that compose the users profiles of the data repository were treated in such way that only they change between 0 and 1. In order to do more comfortable the task of distance computing between profiles. Once the profiles are built, the process can begin. Given the active user A, a set of similar profiles to $profile(A)$ must be found.

4.1 Neighborhood Selection

The success of a CF system depends in high measurement of the effectiveness of algorithm in charge to find the more similar profiles set or neighborhood to the active user.

The neighborhood selection algorithm consists of three main tasks:

- Profiles selection
- Profiles correspondence
- Best profiles collection

These are showed in the following figure.

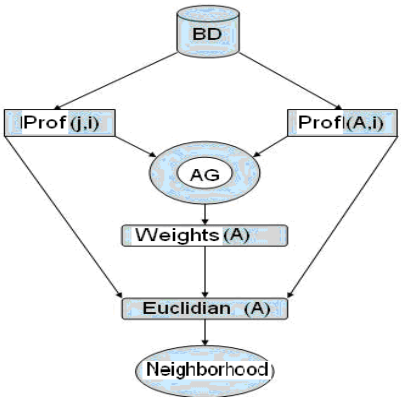


Fig. 4. Neighborhood Set Computing

4.2 Profiles Selection

In order to select the best possible profiles, the ideal scenario would be to use the whole database. However, this is not always the most feasible choice, especially when the database is very big and we have few computational resources. As result, in this work we choose a random sampling of 10 individuals for neighborhood selection.

4.3 Profiles Correspondence

After the profiles selection, the correspondence process computing the distance or similarity between the selected profiles and active user profile, using a distance function. This work focus in the matching process; the genetic algorithm is used to adjust the correspondence between profiles for each active user.

According to Bresse's et al [5] work, it seems that the most of the current recommender systems use algorithms that consider just the "voting information" as the only characteristic on which the comparison between two profiles becomes.

However in real life, the way in which we say that two people are similar is not only based on if they have complementary opinions about a specific subject, but in other factors such as historical precedents and personal details. If we apply this to the profiles adjust, things like demography and life style, this is age, user gender and film gender, must be taken into account. Each user gives an importance or priority to each

characteristic. These priorities can be quantified or numbered. For example, if a user prefers the recommendations based on the opinions of a same gender person, then this characteristic's weight for the gender must be higher than others. In order to implement a really personalized recommender system, the weights need to be captured and adjusted to reflect the user preferences. Here is when we implement a genetic algorithm for the weights evolution.

A potential solution to the problem of evolving the characteristics weight, $w(A)$, for the active user A, is represented as a weight set as it shows next:

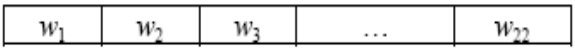


Fig. 5. Individual's phenotype in the population

Where w_f is the weight associated to the characteristic f which genotype is a string of binary values. Each individual has 22 genes, which are evolved by an elitist genetic algorithm (described in other section). The comparison between two profiles can be conduced now using a modified Euclidian distance function, which has into account multiple characteristics. $Euclidean(A, j)$ is the similarity between the active user A and user j.

$$Euclidean(A, j) = \frac{1}{z} \sum_{i=\lambda_1}^{\lambda_z} \sqrt{\sum_{f=1}^{22} wf * diff_{i,f}(A, j)^2} \tag{3}$$

Where:

- A is the active user
- j is a user provided by the profile selection process, besides $j \neq A$
- The common items that A and j has rated, are defined by the set $\lambda_1 \ldots\ldots \lambda_z$
- z is the number of films in common.
- wf, is the active user's weight for the characteristics f
- i is a common film item, where exist $profile(A,i)$ and $profile(i,i)$.
- $diff_{i,f}(A, j)$ is the difference in profile value for the characteristic f between the users A and j about the film i.

The profile value must be normalized before of doing the compute, when the weight for any characteristic is 0 such characteristic is ignored, thus it is possible to allow that the characteristic selection being adaptive to each user preference.

4.4 Best Profiles Collection

Once the Euclidian distances, $Euclidean(A, j)$, are found between $profile(A)$ and $profile(j)$ for all values of j which were selected by the profiles selection process, the best profiles collection algorithm is called, this algorithm classifies each $profile(j)$ according to its similarity with $profile(A)$. The system only uses those profiles that have a Euclidian distance lower than 0.25 as A's neighborhood.

4.5 Items Recommendation

In order to do a recommendation, considering the active user A and a neighborhood set of nearby profiles, it's necessary to find seen items and well rating items by the users in the neighborhood set that the active user has not seen. In our case, we take as recommendations those items that the users of the A's neighborhood have in common with a greater or equal voting to 4, presenting them to the active user through an interface. As the neighborhood set has those users that are similar to A, is very probable that the movies that has liked these users, likes to the active user.

4.6 Genetic Algorithm

As we mentioned above, we use a genetic algorithm to evolve the characteristics weights for the active user, and therefore to help to adjust the profiles correspondence process to the preferences and specific user personality.

For this task one GA standard, a type CHC genetic algorithm and one Learnable Evolutionary Model were chosen. In the implementation a simple binary genetic codification without sign is used, using 8 bits for each of 22 genes, beginning the GA with random genotypes. The genotype is represented in a phenotype (a characteristics weights set) turning the binary genes' alleles in to decimal. The characteristics weights can be computing from these real values. The total phenotype value then is computed adding the real values for the 22 characteristics. Finally, the weights values for each characteristic can be found dividing the real values by the total value, this represent a value in the range [0,1] being this value the assigned value to the characteristic.

4.6.1 Fitness Function

To compute the aptitude for this application is not a trivial task. Each weight set in the GA's population must be used by the profiles correspondence process into the Recommender System. Thus, the system must be executed again over the dataset for each new weight set; we do this in order to compute the aptitude.

But when a RS is executed only produce recommendation, not aptitudes. A poor weight set can result in a poor profiles neighborhood set for an active user and, therefore poor recommendations. A good weight set must result in a good neighbor

set, and good recommendations. For this reason it is required a method for compute the RS's quality, in order to assign an aptitude rate to a corresponding weights.

A solution would be to use the active user as an aptitude function. This would imply obtaining feedback from user asking him to evaluate the recommendations quality [5]. Its input could be used to help to derive the aptitude classification for the current characteristics weights set.

This aptitude classification would give a highly precise view from the user preferences. However, is slightly probable that all users are ready to take part in all the recommendations, because the needed time could be too much. Instead of this, we decided reformulate the problem as supervised learning task. As we described above, considering the active user A and a profiles neighborhood set, the recommendations to A can be done. In addition to these recommendations, is possible to predict what does **A** think about theirs. For example, if certain film it's suggested because similar users saw it, but those users only thought that the film was middle term, and then is probable that the active user thinks the same. Therefore, for the used data domain (*MovieLens*), for the system was possible to recommend new movies and predict how the active user would qualify each film.

The vote prediction formula $predict\_vote(A, j)$, for A on the item i, could be defined as follows:

$$predict\_vote(A,i) = mean_A + k\sum_{j=1}^{n} euclidean(A,j)(vote(j,i) - mean_j) \tag{4}$$

Where:

- $mean_j$ is the vote media for the user j
- k is a normalization factor such as the sum of the Euclidian distances is equal to 1
- $vote(j,i)$ is current vote that user j has given to item i
- n is the neighborhood size.

To compute the fitness measurement for a evolved weights set, the CF finds the profiles neighborhood set for the active user.

5 Results

To realize this type of experiments in which the methods are based in neighborhood techniques, the fact of how the system appropriately assigns the weight to the profile characteristics for a given user (active user) and the criterion to determine how closer is to other user, are very important in the moment to do the predictions.

There are many problems that might appear in a bad weighting of these criteria, which were evident during the different executions of the algorithms and that help us in the task of adjust the neighborhood selection phase. Next we show some

disadvantages that we had during the system execution and generated us a bad definition of the neighborhood set:

- From the users set chosen randomly for the GA execution, none of them have common items with the objective user, this generates null predictions and recommendations because doesn't have a neighborhood set for execute this proceedings.
- Users that have few common items; inclusive could be those that have one item with the current user, could be taken as neighborhood users rejecting users with more common items. This problem can be observed more easily in the Pearson's Correlation Algorithm, because the nature of this algorithm in which only the fact of how a user assigns the evaluations it's taken as criterion to assign the similarity degree. In some proposals it is chosen to assign importance to the co-evaluations number, for example n/50, being n the common evaluations number, for a very small number of n the correlation is too low.
- The neighborhood set is formed only by few users, this brings as consequence that predictions and recommendations only fit to a single items set.

In first place, as we mentioned previously, we analyzed the numbers of items evaluated for each one, we take the possible number of users that could be part of its neighborhood set too. During the implementation of the Genetic Algorithm (CHC) we take as criterion to determine the users' neighborhood of the sample, with the objective user, the followings:

a. The active user's neighborhood's members are those users with a Euclidian distance in relation with this inferior to 0.25.
b. The neighborhood's members are those users that have evaluated at least one item in common with the active user.

These criteria are taken in the planned tests to do.

The opposite case is used to determine the users' neighborhood in the implementation of the Pearson's Algorithm, in which the neighborhood is found using the nearest TOP-N users from whole DB's users set. In the predictions case, these are made in a test set of 14 items by user to evaluate each algorithm's effectiveness and its accuracy is analyzed with MAE metric, it must be in account that users voting (rating) to the items are entire data in the range of 1-5 while prediction generated by the algorithms is in real numbers in the same range.

Table 1 shows the averages of all executions for MAE behavior in our three approaches vs Pearson, Figure 6 is this same behavior but graphically

Table 1. MAE Average Comparation

	MAE_GA	MAE_CHC	MAE_LEM	MAE_Pearson
Average	0,63053	0,51389	**0,28811**	0,87127

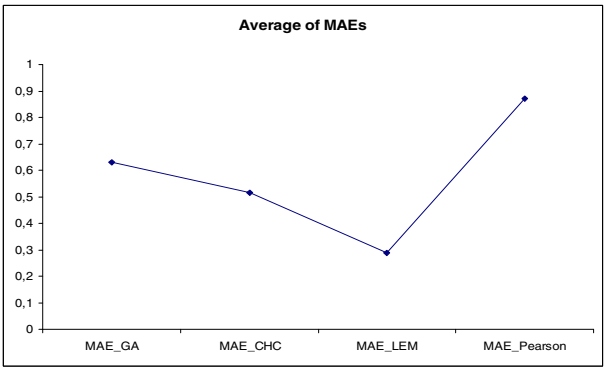


Fig. 6. MAE Average Comparation

The Figure 7 and Figure 8 show for two different random sets of ten users, the MAE average for our different approaches

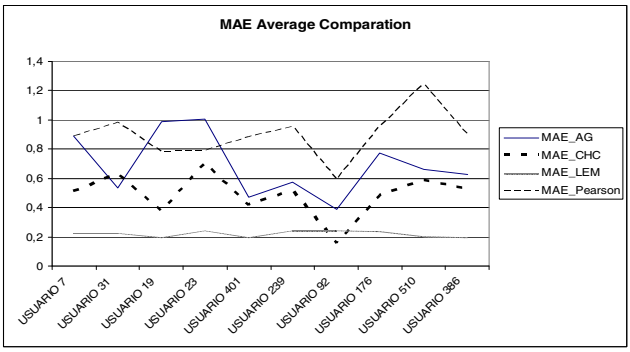


Fig. 7. MAE Average Comparation for ten users selected randomly (1)

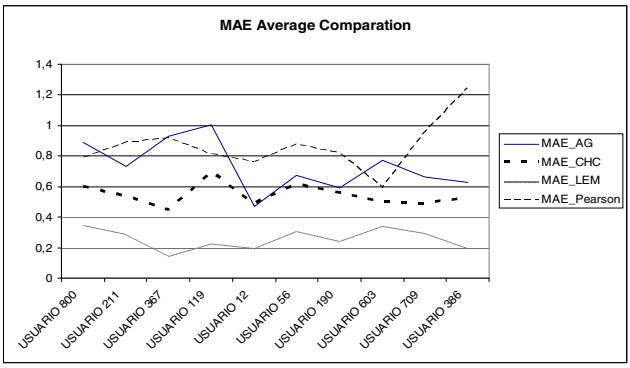


Fig. 8. MAE Average Comparation for ten users selected randomly (2)

Finally we can see some examples for three random different users (Figure 9 to Figure 11) related to the behavior of real rating vs predicted rating for GA standard, CHC and LEM approaches respectively,

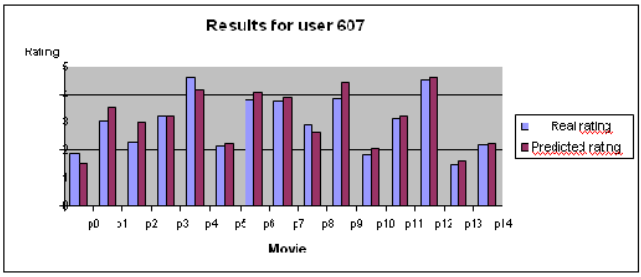


Fig. 9. Results of Real rating vs Predicted rating for one user selected randomly (1)

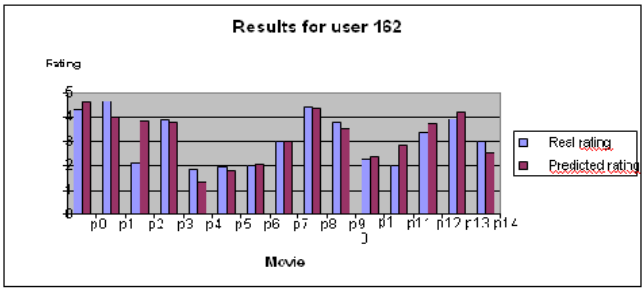


Fig. 10. Results of Real rating vs Predicted rating for one user selected randomly (2)

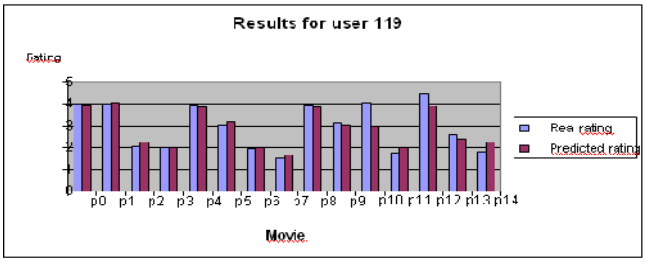


Fig. 11. Results of Real rating vs Predicted rating for one user selected randomly (3)

5.1 Results Analysis

Clearly we can see in our results that, in the most of the cases, GA standard outperform Pearson approach and the two consequent modifications: CHC and LEM approaches improve this behavior.

From the obtained results the principal aspect to stand out is the clear superiority of LEM proposal in relation with the Pearson's Correlation Algorithm, as the evaluation

metrics' graphs show to the realized predictions for the users used in the experiments. Although this not underestimate the utility of the techniques based on made vote by users, in our case is primary to show that besides of vote there are other characteristics like item's information and users' information that might help in the objective of improve the predictions and recommendations realized to a particular user. Although the visual predictions do not seem to have a high level of reliability, the metrics' result express the opposite. Although it's reflected that the predictions are maintained over the media both in application of Pearson's Algorithm like in the Genetic Algorithm approaches.

6 Conclusions

The use of the Evolutionary Computation, especially the CHC genetics algorithms and LEM, as alternative for treat the RS, showed to be a viable choice to deal with problems in predictions based on Collaborative Filtering. The realized tests allow us verify the effectiveness of these technique, because it obtained a higher accuracy in the prediction phase than the results obtained to apply the traditional Pearson's Correlation Algorithm.

The success of this type of experiments allow us know other techniques to deal with problems in the area and improve the expectations of new researches that try to improve the obtained behavior through conventional techniques, besides allow us to see other variables that aren't considered in conventional models.

References

[1] Arrow, K.J.: Social Choice and Individual Values, 2nd edn. Yale University Press (1963)
[2] Bentley, P.J., Corne, D.W.: Creative Evolutionary Systems. Morgan Kaufman Pub. (2001)
[3] Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowledge-Based Systems 46, 109–132 (2013)
[4] Boumaza, A., Brun, A.: From neighbors to global neighbors in collaborative filtering: an evolutionary optimization approach. In: Soule, T. (ed.) Proceedings of the Fourteenth International Conference on Genetic and Evolutionary Computation Conference (GECCO 2012), pp. 345–352. ACM, New York (2012)
[5] Breese, J., Heckermen, D., Kadie, C.: Empirical analysis of predictive algorithms for collaborative filtering. Technical report, Microsoft Research (October 1998)
[6] Chesani, F.: Recommendation Systems. Curso de verano en Ingeniería Informática (1992)
[7] Chislenko, A., et al.: US Patent 6092049: Method and apparatus for efficiently recommending items using automated collaborative filtering and feature-guided automated collaborative filtering (2000)
[8] Clerkin, P., Hayes, C., Cunningham, P.: Automated Case Generation for Recommender Systems Using Knowledge Discovery Techniques Trinity College Dublin
[9] Delgado, J.A. (2000). Agent-Based Information Filtering and Recommender Systems on the Internet. Instituto Tecnológico de Nagoya. Tesis PhD (2000)

[10] Eshelman, L.J.: The CHC adaptive search algorithm: How to have safe search when engaging in non-traditional genetic recombination. Foundations of Genetic Algorithms (1991)
[11] Geyer-Schulz, A., Hahsler, M., Jahn, M.: myVU: A Next Generation Recommender System Based on Observed Consumer Behavior and Interactive Evolutionary Algorithms. In: Gaul, W., Opitz, O., Schader, M. (eds.) Analisis de Datos – Scientific Modeling and Practical Applications, Studies in Classification, Data Analysis, and Knowledge Organization (2000)
[12] Goldberg, D., Nichols, D., Oki, B., Terry, D.: Using collaborative filtering to weave an information tapestry. ACM (1992)
[13] Goldberg, K., Roeder, T., Gupta, D., Perkins, C., Eigentaste, C.: A Constant Time Collaborative Filtering Algorithm. Universidad de California, Berkeley (2000)
[14] Herlocker, J., Konstan, J., Borchers, A., Riedl, J.: An algorithmic framework for performing collaborative filtering. In: SIGIR. ACM (1999)
[15] Hey, J.: System and method of predicting subjective reactions. US Patent 4870579 (1989)
[16] Hu, L., Cao, J., Xu, G., Cao, L., Gu, Z., Zhu, C.: Personalized recommendation via cross-domain triadic factorization. In: Proceedings of the 22nd International Conference on World Wide Web (WWW 2013), pp. 595–606. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva (2013)
[17] Kim, H.-T., Lee, J.-H., Wook Ahn, C.: A recommender system based on interactive evolutionary computation with data grouping. Procedia Computer Science 3, 611–616 (2011)
[18] Konstan, J.A., Bharat, K.: Integrated personal and community recommendations in collaborative filtering. In: CSCW Workshop. ACM, Boston (1996)
[19] Konstan, J., Miller, B., Maltz, D., Herlocker, J., Gordon, L., Riedl, J.: GroupLens: Applying collaborative filtering to usenet news. Communications of the ACM 40(3), 77–87
[20] Michalski, R.S.: Learnable Evolution Model: Evolutionary Process Guided by Machine Learning. Machine Learning 38, 9–40 (2000)
[21] Min Tjoa, A., Höfferer, M., Ehrentraut, G., Untersmayr, P.: Applying Evolutionary Algorithms to the Problem of Information Filtering. In: DEXA Workshop, pp. 450–458 (1997)
[22] Moukas, A., Maes, P.: Amalthaea: an evolving multi-agent information filtering and discovery system for the WWW. In: Autonomous Agents and Multi-agent Systems, pp. 59–88 (1998)
[23] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., Riedl, J.: Grouplens: An open architecture for collaborative filtering of netnews. In: Proceedings of the ACM Conference on Computer Supported Cooperative Work (1994)
[24] Pennock, D.M., Horvitz, E.: Analysis of the axiomatic foundations of collaborative filtering. In: Taller AAAI sobre Inteligencia Artificial para Comercio Electrónico, Conferencia Nacional Sobre IA. Universidad de Stanford, California (1999)
[25] Pennock, D.M., Horvitz, E.: Collaborative filtering by personality diagnosis: A hybrid memory- and model-based approach. In: IJCAI Workshop on Machine Learning for Information Filtering, Stockholm, Sweden (August 1999)
[26] Ricci, F., Rokach, L., Shapira, B., Kantor, P. (eds.): Recommender Systems Handbook. Springer (2011)
[27] Rich, E.: User modeling via stereotypes. Cognitive Science (1979)

[28] Salehi, M., Pourzaferani, M., Razavi, S.: Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model. Egyptian Informatics Journal 14(1), 67–78 (2013)
[29] Shardanand, U., Maes, P.: Social information filtering: Algorithms for automating word of mouth. In: ACM Conference on Computer Human Interaction, CHI (1995)
[30] Sheth, B., Maes, P.: Evolving agents for personalized information filtering. In: Artificial Intelligence for Applications Conference, USA, pp. 345–352 (1993)
[31] Ujjin, S., Bentley, P.: Learning User Preferences Using Evolution. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL 2002), Singapore (2002)
[32] Varian, H., Resnick, P.: Special issue on CF and Recommender Systems. Communications of the ACM 40(3)
[33] Velez-Langs, O.E., Santos, C.: Sistemas Recomendadores: Un Enfoque Desde Los Algoritmos Genéticos. Revista Industrial Data 9(1), 23–31 (2006) (in Spanish)
[34] Alcazar, J., Velez-Langs, O., Salah, J.: Algoritmos Genéticos como Herramientas de Aprendizaje de Características de Usuario en Sistemas Recomendadores. In: Actas del VI Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bio-inspirados 2009, Malaga, Spain, Febrero 11-13 (2009)

Fuzzy Sets Can Be Interpreted as Limits of Crisp Sets, and This Can Help to Fuzzify Crisp Notions

Olga Kosheleva[1], Vladik Kreinovich[1], and Thavatchai Ngamsantivong[2]

[1] University of Texas at El Paso, El Paso, TX 79968, USA
{olgak,vladik}@utep.edu
[2] Computer and Information Science, Faculty of Applied Sciences, King Mongkut's University of Technology North Bangkok, Bangkok 10800 Thailand
tvc@kmutnb.ac.th

Abstract. Fuzzy sets have been originally introduced as *generalizations* of crisp sets, and this is how they are usually considered. From the mathematical viewpoint, the problem with this approach is that most notions allow many different generalizations, so every time we try to generalize some notions to fuzzy sets, we have numerous alternatives. In this paper, we show that fuzzy sets can be alternatively viewed as *limits* of crisp sets. As a result, for some notions, we can come up with a unique generalization – as the limit of the results of applying this notion to the corresponding crisp sets.

1 Formulation of the Problem: Too Many Different Fuzzifications

Crisp Sets: Brief Reminder. Many properties are well-defined and objective ("crisp"). For example, a real number x is either positive or not positive, it is either smaller than 10 or not, etc. Each such crisp property can be described by a (crisp) *set* – namely, by the set S of all the objects that satisfy this property.

For each such set S and for each object x, either the object x belongs to the set S ($x \in S$) or the object x does not belong to the set S ($x \notin S$). A set S can therefore be equivalently described by its *characteristic function* $\mu_S(x)$ that assigns, to each object x, the truth value of the statement $x \in S$. In other words:

- If $x \in S$, i.e., if the object x satisfies the property defining the set, then $\mu_S(x) = 1$.
- On the other hand, if $x \notin S$, i.e., if the object x does not satisfy the desired property, then we take $\mu_S(x) = 0$.

Need for Fuzzy Sets. Humans routinely deal with properties which are not fully well-defined and not fully objective, such as "small", "young", etc. A large portion of our knowledge, of our experience, is described in terms of such properties.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*, Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_31,

To deal with such imprecise ("fuzzy") properties, L. Zadeh introduced the notion of a *fuzzy set*; see, e.g., [1,2,3]. The main idea behind fuzzy sets is that for fuzzy notions S, we no longer have a clear division into objects which absolutely satisfy this notion and objects which absolutely do not satisfy this notion.

For a crisp property like "positive", if we continuously increase a number x from negative values to positive ones, we first have numbers which are absolutely not positive, and then, at $x = 0$, we abruptly switch to numbers which are absolutely positive. In contrast, for a property like "small", as we increase values x from small to not small, we do not abruptly switch from small numbers to non-small ones, the transition is continuous. When the value x is very small, this value will be classified by everyone as absolutely small. Similarly, a very large value will be classified by everyone as absolutely not small. However, for intermediate values, we may differ whether this value is small or nor, and even a single person can hesitate.

To capture this phenomenon of a "smooth" transition between true and false, Zadeh decided to use values between 0 ("false") and 1 ("true") to describe the intermediate states of our beliefs. As a result, a *fuzzy set* S can be mathematically defined as a function $\mu_S(x)$ that assigns, to each possible object x into a number $\mu_S(x)$ from the interval $[0, 1]$.

Fuzzy Sets Have Been Very Successful. Zadeh's idea of capturing the fuzziness of human reasoning has led to numerous successful applications, in control, in clustering, etc.; see, e.g., [1,2].

Important Problem: How to Fuzzify? It is not always easy to apply fuzzy techniques. One of the reasons for this is that there are many alternative fuzzy techniques, and it is not clear which of these techniques we should use.

The reason for this variety of techniques is that, as defined above, the concept of a fuzzy set is a *generalization* of the concept of a crisp set. A crisp set can be defined as function that assigns, to each object x, a value from the 2-element set $\{0, 1\}$, while a fuzzy set is defined as a function that assigns to each x an element of the more general set $[0, 1]$. To extend a concept from crisp sets to fuzzy sets means that we need to extend an operation defined for two truth value 0 and 1 to all possible intermediate values.

This happens, e.g., when we define a complement to a set. In the classical case, this is easy: if an element x belongs to the set S, this element does not belong to the complement $-S$, and vice versa. In other words, if $\chi_S(x) = 1$, then $\chi_{-S}(x) = 0$, and if $\chi_S(x) = 0$, then $\chi_{-S}(x) = 1$. So, a complement is described by an operation that maps 0 to 1 and 1 to 0. We would like to generalize this operation to general fuzzy sets, i.e., to general numbers from the interval $[0, 1]$. It is clear that there are many different ways to extend an operation.

That is why in fuzzy applications, there are many generalizations of negation, many generalization of the union (different t-conorms) and of the intersection (different t-norms), different definitions of probability of a fuzzy event, etc. Which of these generalization should we choose? This is often not clear.

What We Do in This Paper. In this paper, we show that fuzzy sets can be also naturally interpreted as *limits* of crisp sets. Mathematically, this equivalent reformulation in terms of limits is equivalent to the original definition.

Computationally, the reformulation in terms of limits may even be much worse that the original definition, since we are replacing a reasonably simple notion of a function (from the set X of all objects to the interval $[0, 1]$), a notion routinely studied in high school, with a much more complex concept of limit of sets, a concept that is only studied by professional mathematicians.

However, from the viewpoint of generalizations, the limit definition has a clear advantage: once a fuzzy set S is represented as a limit of a sequence of crisp sets S_1, S_2, ..., then we can define, e.g., the probability of this fuzzy set as a limit of the probabilities of the corresponding crisp sets.

We will show that in some cases, this idea indeed enables us to select one definition among many. We will also show that this idea is not a panacea: sometimes, when the sequence S_n tends to S, the corresponding values do not tend to a limit; in such situations, we still have to choose an appropriate fuzzification.

We hope that, in addition to the above-described pragmatic use of this idea, it will also lead to an even wider acceptance of fuzzy set techniques: one may be reluctant to use generalizations, but it is a natural idea to use limits: this is why we routinely use real numbers which are, in effect, nothing else but limits of directly observable rational numbers; that is why we routinely use infinities which are nothing else but limits of real numbers, etc.

2 Polling Interpretation of Fuzzy Properties Naturally Leads to Fuzzy Sets as Limits of Crisp Sets

Polling Interpretation of Fuzzy Properties: Reminder. One of the standard ways to elicit the membership degrees $\mu_S(x)$ is by polling. We ask several (N) experts whether they would agree that an object x satisfies the corresponding property (e.g., whether the given number is small). If M out of N folks say that the given value x is small, we take the ratio $\dfrac{M}{N}$ as the desired degree $\mu_S(x)$.

Polling: Logistic Challenges. When a variable x only takes finitely many possible values, we can simply repeat the above procedure for all these values, and get all the desired degrees $\mu_S(x)$, i.e., get the full description of the corresponding fuzzy set. In reality, there are infinitely many possible values of each quantity x. The first challenge us that we can only ask folks about finitely many different values x.

The second challenge is that it is difficult for many people to think in abstract terms. When I see a person, I can tell whether, in my opinion the person is short or not; when I experience a certain temperature, I can tell whether this temperature corresponds to "warm" or not. However, if for me, a fuzzy threshold between short and non-short lies around 170 cm, and none of my friends are of this threshold height, it will be difficult for me to decide whether someone of

height exactly 170 cm is short or not, without actually observing such a person. Because of this, not only we are limited to finitely many possible values x, but we are also limited to values x corresponding to actual objects shown to the polled experts.

This leads to the third challenge: that we may be polling, e.g., medical doctors or geoscientists located at different parts of the word. It would be too expensive to fly the same patients to all the medical doctors, or to fly all geoscientists to the same earth formation. Realistically, each expert deals with his/her own values x, and our goal is to combine this data.

What Is the Direct Result of Polling. As a result of polling, we get a finite collection C of values x, some of which are marked, by an expert, as having the property S while others are marked as not having the property S. Let $C^+ \subseteq C$ denote the collection of all the values that experts marked as satisfying the desired property.

We are dealing with real-life objects. For two real-life objects, the probability that they have the exact same value of some quantity x is 0. We can therefore safely assume that all these values are different. For example, if we pick two rocks, it is highly improbable that they will have the exact same weight; their weights may be close, but they cannot be exactly equal.

Example. As an example, let us consider a situation when we describe what it means for a rock to be heavy. For this purpose, we ask geoscientists to present examples of rocks which they consider to be heavy and rocks which they consider to be not heavy. For each rock, we mark its weight. Then, C is the collection of the weights all these rocks, and $C^+ \subseteq C$ is the set of the weights of all the rocks which a geophysicist considered to be heavy.

Let us assume, for simplicity, that rocks of weight under 0.5 kg are considered light, rocks whose weight is 1 kg or more are considered heavy, and rocks of intermediate weights are considered heavy by some geophysicist and not heavy by others. For example, we can have 10 rocks of weights 0.1, 0.2, 0.4, 0.6, 0.7, 0.9, and 1.3, of which rocks of weight 0.6, 0.9, and 1.3 have been marked as heavy. In this case, $C = \{0.1, 0.2, 0.4, 0.6, 0.7, 0.9, 1.3\}$ and $C^+ = \{0.6, 0.9, 1.3\}$.

How Polling Techniques Process This Information. Based on the given set of samples marked by experts, how can we estimate the degrees $\mu_S(x)$ corresponding to different values x? For each value x, it is highly improbable that one of the experts actually dealt with this very value; we can only hope to find close values for which an expert has expressed his or her opinion. So, we take a neighborhood $(x-\varepsilon, x+\varepsilon)$ of the desired value x and, in this neighborhood, count the proportion of points which were marked by an expert as satisfying the property S. In other words, as an estimate for $\mu_S(x)$, we take the ratio

$$\mu_S(x) \approx \frac{\#(C^+ \cap (x-\varepsilon, x+\varepsilon))}{\#(C \cap (x-\varepsilon, x+\varepsilon))}, \tag{1}$$

where $\#(s)$ denote the number of elements in a set s and $C^+ \subseteq C$ is the collection of all the values from the set C which experts marked as satisfying the desired property.

To get a more accurate estimate, we need to elicit more opinions from the experts, which would enable us to get more points in the set C. This means that we are not just dealing with a single set C of such values, we get an increasing sequence of finite sets C_1, C_2, ..., C_n, ... corresponding to increasing number of points ($\#(C_n) \to +\infty$), so that

$$\mu_S(x) \approx \frac{\#(C_n^+ \cap (x-\varepsilon, x+\varepsilon))}{\#(C_n \cap (x-\varepsilon, x+\varepsilon))} \tag{2}$$

for all n, where and $C_n^+ \subseteq C_n$ is the collection of all the values from the set C_n which experts marked as satisfying the desired property.

Example. In the above rocks example, we continue eliciting information about heavy and not-heavy rocks. In this case, as C_n, we can take the set of all weights at the moment when we have collected information about n such rocks; then, C_n^+ is the set of all the rocks which the geoscientists considered to be heavy.

How Polling Techniques Process This Information (cont-d). In general, the more points, the more accurate the estimate; thus, the most accurate estimate corresponds to the limit $n \to +\infty$:

$$\mu_S(x) \approx \lim_{n \to +\infty} \frac{\#(C_n^+ \cap (x-\varepsilon, x+\varepsilon))}{\#(C_n \cap (x-\varepsilon, x+\varepsilon))}. \tag{3}$$

This limit, however, is not yet an exact value of $\mu_S(x)$, since this limit represents not just a single value x but the whole interval $(x-\varepsilon, x+\varepsilon)$. To get the exact value of $\mu_S(x)$, we therefore need to perform one more transition to a limit: by taking $\varepsilon \to 0$. After that, we should be able to get the exact value of the membership degree:

$$\mu_S(x) = \lim_{\varepsilon \to 0} \lim_{n \to +\infty} \frac{\#(C_n^+ \cap (x-\varepsilon, x+\varepsilon))}{\#(C_n \cap (x-\varepsilon, x+\varepsilon))}. \tag{4}$$

Let us show how this procedure can lead to a limit interpretation of fuzzy sets.

From Finite Lists C_n to Crisp Sets S_n: Crisp Case. Let us start our analysis of the situation with the case when the desired property is crisp. Suppose that we have a finite set C_n of values for each of which experts decided whether this value satisfies the desired property or not. For each given value x, how can we then decide whether the given value x satisfied the desired property?

A natural idea, as we have mentioned, is to check whether some value which is close to x have been classified by experts. The closer this already-classified value to x, the more confident we are that this element and the desired value x both satisfy or both do not satisfy the property S. Thus, a reasonable idea is to look for the element from C_n which is the closest to x:

- if this closest element satisfies the property S, then we conclude that the given value x also satisfies the property S;
- is this closest element does not satisfy the property S, then we conclude that the given value x does not satisfy S either.

Of course, there will be few threshold cases when the value x is exactly in between two values, one classifies as satisfying S and another classified as not satisfying S, but these values are rare, so we can arbitrarily classify them to S or to a complement to S.

Thus, for each n, we divide the set X of all possible values of x into two sets:

- the set of all the values x which are, based on the set C_n, classified as satisfying the property S; we will denote this set by S_n; and
- the set of all the values x which are, based on the set C_n, classified as not satisfying the property S; this second set is simply a complement $X - S_n$.

As we elicit more and more opinions from experts, we get sets C_n which have more and more points; moreover, we get more and more points within each interval. So in the limit, when we increase n, the corresponding sets S_n and $X - S_n$ becomes closer and closer to the actual sets S and $X - S$, in the sense that:

- if the value x actually satisfies the property S, i.e., if $x \in S$, then most probably, starting with some sufficiently large n, it will be recognized by this procedure as having this property;
- similarly, if the value x actually does not satisfy the property S, i.e., if $x \notin S$, then most probably, starting with some sufficiently large n, it will be recognized by this procedure as not having this property.

In other words, if we form the values $\chi_{S_n}(x)$, then for sufficiently large n, these values will coincide with $\chi(S)$, i.e., we will have $\chi_S(x) = \lim\limits_{n \to +\infty} \chi_{S_n}(x)$.

From Finite Lists C_n to Crisp Sets S_n: General (Fuzzy) Case. What happens in the general (fuzzy) case? In this case, based on each set of observations C_n, we can also subdivide the entire set X into two crisp subsets:

- the set S_n of all the values x for which the closest point from C_n is classified as having the property S, and
- the set $X - S_n$ of all the values x for which the closest point from C_n is classified as not having the property S.

Example. In the rocks example, $C_n = \{0.1, 0.2, 0.4, 0.6, 0.7, 0.9, 1.3\}$ and $C_n^+ = \{0.6, 0.9, 1.3\}$. Then, e.g.,

- for 0.62, the closest point from C_n is 0.6; this point classified as heavy, so 0.62 is assigned to the set S_n of heavy objects;
- on the other hand, for 0.68, the closest point from the set C_n is 0.7 which is classified as not heavy, so 0.68 is assigned to the set $X - S_n$ of not-heavy rocks.

The borderline points of the set S_n are the midpoints between the neighboring heavy and not-heavy values from C_n, i.e.:

- a midpoint 0.5 between 0.4 and 0.6,
- a midpoint 0.65 between 0.6 and 0.7, and
- a midpoint 0.8 between 0.7 and 0.9.

Thus, in this example, the set S_n takes the form $[0.5, 0.65] \cup [0.8, \infty)$.

From Finite Lists C_n to Crisp Sets S_n: General (Fuzzy) Case (cont-d). So far, the description is similar to the corresponding description of the crisp case. The difference is what happens in the intermediate values x for which the experts differ. For such intermediate values, if we start with a randomly selected collection of values around x, out of which experts classify a proportion $\mu_S(x)$ as satisfying the property S, then, as one can easily check, the proportion of points assigned to the set S_n will also be approximately the same. In other words, we will have

$$\mu_S(x) \approx \frac{\text{len}(S_n \cap (x-\varepsilon, x+\varepsilon))}{\text{len}(x-\varepsilon, x+\varepsilon)}, \tag{5}$$

where $\text{len}(s)$ denotes the total length of the set S:

- for an interval, it is exactly its length;
- for a union of several disjoint intervals, it is the sum of their lengths.

Similarly to the above formulas (3)–(4), to get an accurate value $\mu_S(x)$, we need to take more and more points n and narrower and narrower interval $(x-\varepsilon, x+\varepsilon)$. Then, we get

$$\mu_S(x) = \lim_{\varepsilon \to 0} \lim_{n \to +\infty} \frac{\text{len}(S_n \cap (x-\varepsilon, x+\varepsilon))}{\text{len}(x-\varepsilon, x+\varepsilon)}. \tag{6}$$

Example. In the rocks example, the set S_n of all the objects which are classified as heavy has the form $S_n = [0.5, 0.65] \cup [0.8, \infty)$.

- On the interval $[0, 0.5]$, the only value which is classified as heavy is the value 0.5. These values form a degenerate interval $[0.5, 0.5]$. The length of this interval is 0, so we have $\mu(x) \approx 0$.
- On the interval $[0.5, 1]$ of length 0.5, the value which are classified as heavy form the set $[0.5, 0.65] \cup [0.8, 1.0]$. The length of this set is $0.15 + 0.2 = 0.35$, so we have $\mu(x) \approx \dfrac{0.35}{0.5} = 0.7$.
- On the interval $[1, 2]$ of length 1, all the values are classified as heavy, so we have $\mu(x) \approx 1$.

As we collect more data, we get a better approximation to the desired membership function.

Resulting Idea: Fuzzy Set As a Limit of Crisp Sets. Similarly to describing a crisp set S as a limit of the corresponding crisp sets S_n, we can thus formally describe a fuzzy set as a limit of crisp sets if the formula (6) is satisfied.

We can now define operations on fuzzy sets as limits of operations on the corresponding sets S_n – when such a limit exists. Let us describe this idea in precise terms.

3 Fuzzy Sets as Limits of Crisp Sets: Definitions and Results

Definition 1. *We say that a fuzzy set S with a membership function $\mu_S(x)$ is a* limit *of a sequence of (crisp) sets S_n, and denote it as $S_n \to S$, if for every $x \in X$, the formula (6) holds.*

Comment. A similar definition can be formulated for fuzzy subsets of a plane, a 3-D space, or, more generally, a multi-D space; in this case:

- to describe a neighborhood of a point $x = (x_1, \ldots, x_d)$, it is reasonable to use, e.g., boxes

$$(x_1 - \varepsilon_1, x_1 + \varepsilon_1) \times \ldots \times (x_d - \varepsilon_d, x_d + \varepsilon_d)$$

  instead of intervals;
- instead of a length of a set, we need to use a more general Lebesgue measure; e.g., area for sets in a plane, volume for 3-D sets. etc.

Let us first show that the above intuitive idea indeed works, i.e., that generic fuzzy sets can indeed be represented as limits of crisp sets.

Proposition 1. *Every fuzzy set with a continuous membership function can be represented as a limit of crisp sets.*

Proof. One way to describe the corresponding set S_n is to divide the real axis into intervals $\left[\frac{k}{n}, \frac{k+1}{n}\right)$ corresponding to different integers k, and to divide each such interval of length $\frac{1}{n}$ into two parts:

- a part $\left[\frac{k}{n}, \frac{k}{n} + \mu\left(\frac{k}{n}\right) \cdot \frac{1}{n}\right)$, a portion $\mu\left(\frac{k}{n}\right)$, is assigned to the set S_n, while
- the remaining part $\left(\frac{k}{n} + \mu\left(\frac{k}{n}\right) \cdot \frac{1}{n}, \frac{k+1}{n}\right)$ is assigned to the complement of the set S_n.

One can easily check that for the resulting sequence of sets

$$S_n = \bigcup_k \left[\frac{k}{n}, \frac{k}{n} + \mu\left(\frac{k}{n}\right) \cdot \frac{1}{n}\right),$$

the equation (6) holds for every x. The statement is proven.

Let us show that this enables us to uniquely describe probability of a fuzzy set.

Definition 2. *Let $\rho(x)$ be a continuous probability density, let $P(s) \stackrel{\text{def}}{=} \int_s \rho(x)\,dx$ be the resulting probability measure, and let S be a fuzzy set. We say that a real number $P(S)$ is a* probability *of a fuzzy set S if for every sequence of crisp set S_n with $S_n \to S$, we have $P(S_n) \to P(S)$.*

Proposition 2. *For every fuzzy set S with a continuous membership function $\mu(x)$, its probability is well-defined and equal to $P(S) = \int \mu(x) \cdot \rho(x)\,dx$.*

Comment. This result provides one more justification for the original Zadeh's definition of the probability of a fuzzy set [4].

Another case when the limit idea enables us to select a unique generalization is the case of a complement.

Definition 3. *We say that a fuzzy set S' is a* complement *to a fuzzy set S if for every sequence S_n of crisp sets for which $S_n \to S$, we have $-S_n \to S'$ for the sequence of their complements $-S_n$.*

Proposition 3. *For every fuzzy set S with a continuous membership function $\mu(x)$, its complement S' is well-defined and its membership function is equal to $\mu_{S'}(x) = 1 - \mu(S)$.*

The Limit Idea Is Not a Panacea. While the above idea works well for defining probability, it is not a panacea. Let us show, for example, that this idea does not lead to a unique definition of a union or intersection of two fuzzy sets.

Indeed, ideally, we should be able to define the intersection of two fuzzy sets S and S' in a similar manner:

- we say that a fuzzy set is a union $S \cup S'$ of fuzzy sets S and S' if for every two sequences of crisp sets $S_n \to S$ and $S'_n \to S'$ imply $S_n \cup S'_n \to S \cup S'$;
- similarly, we say that a fuzzy set is an intersection $S \cap S'$ of fuzzy sets S and S' if for every two sequences of crisp sets $S_n \to S$ and $S'_n \to S'$ imply $S_n \cap S'_n \to S \cap S'$.

Alas, it turns out that for different sequences S_n and S'_n, we get different limits. Indeed, let us consider, for example, the two identical fuzzy sets $S = S'$ both corresponding to complete ignorance $\mu_S(x) = \mu_{S'}(x) = 0.5$. For this membership function, the construction from the proof of Proposition 1 leads to the sets

$$S_n = \bigcup_k \left[\frac{k}{n}, \frac{k}{n} + \frac{1}{2} \cdot \frac{1}{n}\right). \tag{7}$$

If we use these sets as sequences S_n and S'_n corresponding to both fuzzy sets S and S', then we get $S_n \cup S'_n = S_n \cap S'_n = S_n$, and thus, for the limit fuzzy sets, we get $\mu(x) = 0.5$ for all x.

Alternatively, we can still use S_n as a sequence of crisp sets approximating the set S while using a different sequence

$$S'_n = \bigcup_k \left[\frac{k}{n} + \frac{1}{2}\cdot\frac{1}{n}, \frac{k+1}{n}\right) \tag{8}$$

to approximate the set S'. In this case:

- the union $S_n \cup S'_n$ is the whole real line, so in the limit (6), we get $\mu(x) = 1$ for all x;
- on the other hand, the intersection $S_n \cap S'_n$ consists of midpoints of all intervals $\left[\frac{k}{n}, \frac{k+1}{n}\right)$, so here, in the limit (6), we have $\mu(x) = 0$.

4 Discussion and Future Work

Discussion. In this paper, we described how a fuzzy set can be represented in terms of several crisp sets – specifically, a fuzzy set is represented as a *limit* of crisp sets.

Another well-known way of representing a fuzzy set with a membership function $\mu_S(x)$ in terms of crisp sets is a representation in terms of alpha-cuts $\{x : \mu_S(x) \geq \alpha\}$. The main difference is that

- to uniquely determine a fuzzy set, we need to know *all* its alpha-cuts, while
- we do not need all limit sets S_n to uniquely determine a fuzzy set: it is sufficient to know the sets S_{n_k} for some sequence $n_k \to \infty$ (e.g., for $n_k = k^2$).

Future Work. In mathematical terms, the property that

$$S_n \to S \text{ implies } P(S_n) \to P(S)$$

is known as *continuity*. In these terms, we can say that:

- probability is a continuous function of sets (in the sense of convergence $S_n \to S$);
- complement is a continuous operation, while
- union and intersection are *discontinuous* operations.

For such discontinuous operations, instead of a single limit value, we have an *interval* of possible limit values. So maybe we can extend the limit idea to interval-valued fuzzy sets?

Acknowledgments. This work was supported in part by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from the National Institutes of Health, and by a grant N62909-12-1-7039 from the Office of Naval Research.

The authors are thankful to the anonymous referees for valuable suggestions.

References

1. Klir, G.J., Yuan, B.: Fuzzy Sets and Fuzzy Logic. Prentice Hall, Upper Saddle River (1995)
2. Nguyen, H.T., Walker, E.A.: First Course In Fuzzy Logic. CRC Press, Boca Raton (2006)
3. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
4. Zadeh, L.A.: Probabilisty measures of fuzzy events. Journal of Mathematical Analysis and Applications 23(2), 421–427 (1968)

How to Gauge Accuracy of Measurements and of Expert Estimates: Beyond Normal Distributions

Christian Servin[1,2], Aline Jaimes[2], Craig Tweedie[2], Aaron Velasco[2], Omar Ochoa[2], and Vladik Kreinovich[2]

[1] Information Technology Department, El Paso Community College, El Paso, TX 79915, USA
[2] Cyber-ShARE Center, University of Texas at El Paso, El Paso, TX 79968, USA
{christians,omar}@miners.utep.edu, ajaimes@mines.utep.edu, {ctweedie,aavelasco,vladik}@utep.edu

Abstract. To properly process data, we need to know the accuracy of different data points, i.e., accuracy of different measurement results and expert estimates. Often, this accuracy is not given. For such situations, we describe how this accuracy can be estimated based on the available data.

1 Formulation of the Problem

Need to Gauge Accuracy. To properly process data, it is important to know the accuracy of different data values, i.e., the accuracy of different measurement results and expert estimates; see, e.g., [3–5]. In many cases, this accuracy information is available, but in many other practical situations, we do not have this information. In such situations, it is necessary to extract this accuracy information from the data itself.

Extracting Uncertainty from Data: Traditional Approach. The usual way to gauge of the uncertainty of a measuring instrument is to compare the result $\widetilde{x}$ produced by this measuring instruments with the result $\widetilde{x}_s$ of measuring the same quantity x by a much more accurate ("standard") measuring instrument.

Since the "standard" measuring instrument is much more accurate than the instrument that we are trying to calibrate, we can safely ignore the inaccuracy of its measurements and take $\widetilde{x}_s$ as a good approximation to the actual value x. In this case, the difference $\widetilde{x} - \widetilde{x}_s$ between the measurement results can serve as a good approximation to the desired measurement accuracy $\Delta x = \widetilde{x} - x$.

Traditional Approach Cannot Be Applied for Calibrating State-of-the-Art Measuring Instruments. The above traditional approach works well for many measuring instruments. However, we cannot apply this approach for calibrating state-of-the-art instrument, because these instruments are the best we have. There are no other instruments which are much more accurate than these ones – and which can therefore serve as standard measuring instruments for our calibration.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*, Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_32,

Such situations are ubiquitous; for example:

- in the environmental sciences, we want to gauge the accuracy with which the Eddy covariance tower measure the Carbon and heat fluxes; see, e.g., [1];
- in the geosciences, we want to gauge how accurately seismic [2], gravity, and other techniques reconstruct the density at different depths and different locations.

How State-of-the-Art Measuring Instruments Are Calibrated: Case of Normally Distributed Measurement Errors. Calibration of state-of-the-art measuring instruments is possible if we make a usual assumption that the measurement errors are normally distributed with mean 0. Under this assumption, to fully describe the distribution of the measurement errors, it is sufficient to estimate the standard deviation σ of this distribution.

There are two possible approaches for estimating this standard deviation. The first approach is applicable when we have several similar measuring instruments. For example, we can have two nearby towers, or we can bring additional sensors to the existing tower. In such a situation, instead of a single measurement result $\widetilde{x}$, we have two different results $\widetilde{x}^{(1)}$ and $\widetilde{x}^{(2)}$ of measuring the same quantity x. Here, by definition of the measurement error, $\widetilde{x}^{(1)} = x + \Delta x^{(1)}$ and $\widetilde{x}^{(2)} = x + \Delta x^{(2)}$ and therefore, $\widetilde{x}^{(1)} - \widetilde{x}^{(2)} = \Delta x^{(1)} - \Delta x^{(2)}$.

Each of the random variables $\Delta x^{(1)}$ and $\Delta x^{(2)}$ is normally distributed with mean 0 and (unknown) standard deviation σ (i.e., variance σ^2). Since the two measuring instruments are independence, the corresponding random variables $\Delta x^{(1)}$ and $\Delta x^{(2)}$ are also independent, and so, the variance of their difference is equal to the sum of their variances $\sigma^2 + \sigma^2 = 2\sigma^2$. Thus, the standard deviation σ' of this difference is equal to $\sqrt{2} \cdot \sigma$. We can estimate this standard deviation σ' based on the observed differences $\widetilde{x}^{(1)} - \widetilde{x}^{(2)}$ and therefore, we can estimate σ as $\dfrac{\sigma'}{\sqrt{2}}$.

This approach is not applicable in the geosciences applications, when we usually have only one seismic map, only one gravity map, etc. In such situations, we have several measurement results $\widetilde{x}^{(i)}$ with, in general, different standard deviations $\sigma^{(i)}$. For every two measuring instruments i and j, the difference $\widetilde{x}^{(i)} - \widetilde{x}^{(j)}$ is normally distributed with the variance $\left(\sigma^{(i)}\right)^2 + \left(\sigma^{(j)}\right)^2$. By comparing actual measurement results, we can estimate this variance and thus, get an estimate e_{ij} for the sum. As a result, e.g., for the case when we have three different measuring instruments, we get three values e_{ij} for which:

$$e_{12} = \left(\sigma^{(1)}\right)^2 + \left(\sigma^{(2)}\right)^2; \quad e_{13} = \left(\sigma^{(1)}\right)^2 + \left(\sigma^{(3)}\right)^2;$$

$$e_{23} = \left(\sigma^{(2)}\right)^2 + \left(\sigma^{(3)}\right)^2.$$

Here, we have a system of three linear equations with three unknowns, from which we can uniquely determined all three desired variances $\left(\sigma^{(i)}\right)^2$:

$$\left(\sigma^{(1)}\right)^2 = \frac{e_{12} + e_{13} - e_{23}}{2}; \quad \left(\sigma^{(2)}\right)^2 = \frac{e_{12} + e_{23} - e_{13}}{2};$$

$$\left(\sigma^{(3)}\right)^2 = \frac{e_{13} + e_{23} - e_{12}}{2}.$$

Need to Go Beyond Normal Distributions, and Resulting Problem. In practice, the distribution of measurement errors is often different from normal; this is the case, e.g., in measuring fluxes [1]. In such cases, we can still use the same techniques to find the standard deviation of the measurement error. However, in general, it is not enough to know the standard deviation to uniquely determine the distribution: e.g., we may have (and we sometimes do have) an asymmetric distribution, for which the skewness is different from 0 (i.e., equivalently, the expected value of $(\Delta x)^3$ is different from 0).

It is known that in this case, in contrast to the case of the normal distribution, we cannot uniquely reconstruct the distribution of Δx from the known distribution of the difference $\Delta x^{(1)} - \Delta x^{(2)}$. Indeed, if we have an asymmetric distribution for Δx, i.e., a distribution which is not invariant under the transformation $\Delta x \to -\Delta x$, this means that the distribution for $\Delta y \stackrel{\text{def}}{=} -\Delta x$ is different from the distribution for Δx. However, since $\Delta y^{(1)} - \Delta y^{(2)} = \Delta x^{(2)} - \Delta x^{(1)}$, the y-difference is also equal to the difference between two independent variables with the distribution Δx and thus, distribution for the difference $\Delta y^{(1)} - \Delta y^{(2)}$ is exactly the same as for the difference $\Delta x^{(1)} - \Delta x^{(2)}$. In other words, if we know the distribution for the difference $\Delta x^{(1)} - \Delta x^{(2)}$, we cannot uniquely reconstruct the distribution for Δx, because, in addition to the original distribution for Δx, all the observations are also consistent with the distribution for $\Delta y = -\Delta x$.

This known non-uniqueness naturally leads to the following questions:

- first, a theoretical question: since we cannot uniquely reconstruct the distribution for Δx, what information about this distribution can we reconstruct?
- second, a practical question: for those characteristics of Δx which can be theoretically reconstructed, we need to design computationally efficient algorithms for reconstructing these characteristics.

2 Technique for Solving the Problem

Technique to Use. To solve these questions, let us use the Fourier analysis technique.

What we want to find is the probability density $\rho(z)$ describing the distribution of the measurement error $z \stackrel{\text{def}}{=} \Delta x$. In order to find the unknown probability density, we will first find its Fourier transform $F(\omega) = \int \rho(z) \cdot e^{\mathrm{i}\cdot\omega\cdot z}\, dz$. By definition, this Fourier transform is equal to the mathematical expectation of the function $e^{\mathrm{i}\cdot\omega\cdot z}$: $F(\omega) = E\left[e^{\mathrm{i}\cdot\omega\cdot z}\right]$. Such a mathematical expectation is also known as a *characteristic function* of the random variable z.

Based on the observed values of the difference $z^{(1)} - z^{(2)}$, we can estimate the characteristic function $D(\omega)$ of this difference: $D(\omega) = E\left[e^{\mathrm{i}\cdot\omega\cdot(z^{(1)} - z^{(2)})}\right]$. Here,

$$e^{\mathrm{i}\cdot\omega\cdot(z^{(1)} - z^{(2)})} = e^{(\mathrm{i}\cdot\omega\cdot z^{(1)}) + (-\mathrm{i}\cdot\omega\cdot z^{(2)})} = e^{\mathrm{i}\cdot\omega\cdot z^{(1)}} \cdot e^{-\mathrm{i}\cdot\omega\cdot z^{(2)}}.$$

Measurement errors $z^{(1)}$ and $z^{(2)}$ corresponding to two measuring instruments are usually assumed to be independent. Thus, the variables $e^{\mathrm{i}\cdot\omega\cdot z^{(1)}}$ and $e^{-\mathrm{i}\cdot\omega\cdot z^{(2)}}$ are also independent. It is known that the expected value of the product of two independent variables is equal to the product of their expected values, thus,

$$D(\omega) = E\left[e^{\mathrm{i}\cdot\omega\cdot z^{(1)}}\right] \cdot E\left[e^{-\mathrm{i}\cdot\omega\cdot z^{(2)}}\right],$$

i.e., $D(\omega) = F(\omega)\cdot F(-\omega)$. Here, $F(-\omega) = E\left[e^{-\mathrm{i}\cdot\omega\cdot z}\right] = E\left[\left(e^{\mathrm{i}\cdot\omega\cdot z}\right)^*\right]$, where t^* means complex conjugation, i.e., an operation that transforms $t = a + b\cdot\mathrm{i}$ into $t^* = a - b\cdot\mathrm{i}$. Thus, $F(-\omega) = F^*(\omega)$, and the above formula takes the form

$$D(\omega) = F(\omega)\cdot F^*(\omega) = |F(\omega)|^2.$$

In other words, the fact that we know $D(\omega)$ means that we know the absolute value (modulus) of the complex-valued function $F(\omega)$.

In these terms, the problems becomes: how can we reconstruct the complex-valued function $F(\omega)$ if we only know its absolute value?

3 Is It Possible to Estimate Accuracy?

How to Use Fourier Techniques to Solve the Theoretical Question. First, let us address the theoretical question: since, in general, we cannot reconstruct $\rho(z)$ (or, equivalently, $F(\omega)$) uniquely, what information about $\rho(z)$ (and, correspondingly, about $F(\omega)$) can we reconstruct?

To solve this theoretical question, let us take into account the practical features of this problem. First, it needs to be mentioned that, from the practical viewpoint, we need to take into account that the situation in, e.g., Eddy covariance tower measurements is more complex that we described, because the tower does not measure *one* single quantity, it simultaneously measuring *several* quantities: carbon flux, heat flux, etc. Since these different measurements are based on data from the same sensors, it is reasonable to expect that the resulting measurement errors are correlated. Thus, to fully describe the measurement uncertainty, it is not enough to describe the distribution of each 1-D measurement error, we need to describe a joint distribution of all the measurement errors $z = (z_1, z_2, \ldots)$. In this multi-D case, we can use the multi-D Fourier transforms and characteristic functions, where for $\omega = (\omega_1, \omega_2, \ldots)$, we define $F(\omega) = E\left[e^{\mathrm{i}\cdot\omega\cdot z}\right]$, with $\omega\cdot z \stackrel{\mathrm{def}}{=} \omega_1\cdot z_1 + \omega_2\cdot z_2 + \ldots$

Second, we need to take into account that while theoretically, we can consider all possible values of the difference $z^{(1)} - z^{(2)}$, in practice, we can only get values which are proportional to the smallest measuring unit h. For example, if we measure distance and the smallest distance we can measure is centimeters, then the measuring instrument can only return values 0 cm, 1 cm, 2 cm, etc. In other words, in reality, the value z can only take discrete values. If we take the smallest value of z as the new starting point (i.e., as 0), then the possible values of z take the form $z = 0$, $z = h$, $z = 2h$, ..., until we reach the upper bound $z = N\cdot h$ for

some integer N. For these values, in the 1-D case, the Fourier transform takes the form

$$F(\omega) = E\left[e^{\mathrm{i}\cdot\omega\cdot z}\right] = \sum_{k=0}^{N} p_k \cdot e^{\mathrm{i}\cdot\omega\cdot k\cdot h},$$

where p_k is the probability of the value $z = k\cdot h$. This formula can be equivalently rewritten as $F(\omega) = \sum\limits_{k=0}^{N} p_k \cdot s^k$, where $s \stackrel{\text{def}}{=} e^{\mathrm{i}\cdot\omega\cdot h}$. Similarly, in the multi-D case, we have $z = (k_1 \cdot h_1, k_2 \cdot h_2, \ldots)$, and thus,

$$e^{\mathrm{i}\cdot\omega\cdot k\cdot h} = e^{\mathrm{i}\cdot\omega\cdot(k_1\cdot h_1 + k_2\cdot h_2 + \ldots)} = e^{\mathrm{i}\cdot\omega_1\cdot k_1\cdot h_1} \cdot e^{\mathrm{i}\cdot\omega\cdot k_2\cdot h_2} \cdot \ldots,$$

so we have

$$F(\omega) = \sum_{k_1=0}^{N_1} \sum_{k_2=0}^{N_2} \ldots p_k \cdot s_1^{k_1} \cdot s_2^{k_2} \cdot \ldots,$$

where $s_k \stackrel{\text{def}}{=} e^{\mathrm{i}\cdot\omega_k\cdot h_k}$. In other words, we have a polynomial of the variables $s_1, s_2, \ldots$:

$$P(s_1, s_2, \ldots) = \sum_{k_1=0}^{N_1} \sum_{k_2=0}^{N_2} \ldots p_k \cdot s_1^{k_1} \cdot s_2^{k_2} \cdot \ldots$$

Different values of ω correspond to different values of $s = (s_1, s_2, \ldots)$. Thus, the fact that we know the values of $|F(\omega)|^2$ for different ω is equivalent to knowing the values of $|P(s)|^2$ for all possible values $s = (s_1, s_2, \ldots)$.

In these terms, the theoretical question takes the following form: we know the values $D(s) = |P(s)|^2 = P(s) \cdot P^*(s)$ for some polynomial $P(s)$, we need to reconstruct this polynomial. In the 1-D case, each complex-valued polynomial of degree N has, in general, N complex roots $s^{(1)}$, $s^{(2)}$, etc., and can, therefore, be represented as $|P(s)|^2 = \text{const} \cdot (s - s^{(1)}) \cdot (s - s^{(2)}) \cdot \ldots$ In this case, there are many factors, so there are many ways to represent it as a product – which explains the above-described non-uniqueness of representing $D(s)$ as the product of two polynomials $P(s)$ and $P^*(s)$

Interestingly, in contrast to the 1-D case, in which each polynomial can be represented as a product of polynomials of 1st order, in the multi-D case, a generic polynomial *cannot* be represented as a product of polynomials of smaller degrees. This fact can be easily illustrated on the example of polynomials of two variables.

To describe a general polynomial of two variables $\sum\limits_{k=0}^{n} \sum\limits_{l=1}^{n} c_{kl} \cdot s_1^k \cdot s_2^l$ in which each of the variables has a degree $\le n$, we need to describe all possible coefficients c_{kl}. Each of the indices k and l can take $n+1$ possible values 0, 1, ..., n, so overall, we need to describe $(n+1)^2$ coefficients.

When two polynomials multiply, the degrees add: $s^m \cdot s^{m'} = s^{m+m'}$. Thus, if we represent $P(s)$ as a product of two polynomials, one of them must have a degree $m < n$, and the other one degree $n - m$. In general:

- we need $(m+1)^2$ coefficients to describe a polynomial of degree m and
- we need $(n-m+1)^2$ coefficients to describe a polynomial of degree $n-m$,
- so to describe arbitrary products of such polynomials, we need $(m+1)^2 + (n-m+1)^2$ coefficients.

To be more precise, in such a product, we can always multiply one of the polynomials by a constant and divide another one by the same constant, without changing the product. Thus, we can always assume that, e.g., in the first polynomial, the free term c_{00} is equal to 1. As a result, we need one fewer coefficient to describe a general product: $(m+1)^2 + (n-m+1)^2 - 1$.

To be able to represent a generic polynomial $P(s)$ of degree n as such a product $P(s) = P_m(s) \cdot P_{n-m}(s)$, we need to make sure that the coefficients at all all $(n+1)^2$ possible degrees $s_1^k \cdot s_2^l$ are the same on both sides of this equation. This requirement leads to $(n+1)^2$ equations with $(m+1)^2 + (n-m+1)^2 - 1$ unknowns.

In general, a system of equations is solvable if the number of equations does not exceed the number of unknowns. Thus, we must have

$$(n+1)^2 \le (m+1)^2 + (n-m+1)^2 - 1.$$

Opening parentheses, we get

$$n^2 + 2n + 1 \le m^2 + 2m + 1 + (n-m)^2 + 2 \cdot (n-m) + 1 - 1.$$

The constant terms in both sides cancel each other, as well as the terms $2n$ in the left-hand side and $2m + 2 \cdot (n-m) = 2n$ in the right-hand side, so we get an equivalent inequality $n^2 \le m^2 + (n-m)^2$. Opening parentheses, we get $n^2 \le m^2 + n^2 - 2 \cdot n \cdot m + m^2$. Cancelling n^2 in both sides, we get $0 \le 2m^2 - 2 \cdot n \cdot m$. Dividing both sides by $2m$, we get an equivalent inequality $0 \le m - n$, which clearly contradicts to our assumption that $m < n$.

Let us go back to our problem. We know the product $D(s) = P(s) \cdot P^*(s)$, and we want to reconstruct the polynomial $P(s)$. We know that this problem is not uniquely solvable, i.e., that there exist other polynomials $Q(s) \ne P(s)$ for which $D(s) = P(s) \cdot P^*(s) = Q(s) \cdot Q^*(s)$. Since, in general, a polynomial $P(s)$ of several variables cannot be represented as a product – i.e., is "prime" in terms of factorization the same way prime numbers are – the fact that the two products coincide means that $Q(s)$ must be equal to one of the two prime factors in the decomposition $D(s) = P(s) \cdot P^*(s)$. Since we know that $Q(s)$ is different from $P(s)$, we thus conclude that $Q(s) = P^*(s)$.

By going back to the definitions, one can see that for the distribution $\rho'(x) = \rho(-x)$, the corresponding polynomial has exactly the form $Q(s) = P^*(s)$. Thus, in general, this is the *only* non-uniqueness that we have: each distribution which is consistent with the observation of differences coincides either with the original distribution $\rho(x)$ or with the distribution $\rho'(x) = \rho(-x)$. In other words, we arrive at the following result.

Answer to the Theoretical Question. We have proven that, in general, each distribution which is consistent with the observation of differences $\Delta x^{(1)} - \Delta x^{(2)}$ coincides either with the original distribution $\rho(x)$ or with the distribution $\rho'(x) \stackrel{\text{def}}{=} \rho(-x)$.

4 Practical Question: How to Gauge the Accuracy

How to Use Fourier Techniques to Solve the Practical Question: Idea. We want to find a probability distribution $\rho(z)$ which is consistent with the observed characteristic function $D(\omega)$ for the difference. In precise terms, we want to find a function $\rho(z)$ which satisfies the following two conditions:

- the first condition is that $\rho(z) \geq 0$ for all z, and
- the second condition is that $|F(\omega)|^2 = D(\omega)$, where $F(\omega)$ denotes the Fourier transform of the function $\rho(x)$.

One way to find the unknown function that satisfies two conditions is to use the method of successive projections. In this method, we start with an arbitrary function $\rho^{(0)}(z)$. On the k-th iteration, we start with the result $\rho^{(k-1)}(z)$ of the previous iteration, and we do the following:

- first, we project this function $\rho^{(k-1)}(z)$ onto the set of all functions which satisfy the first condition; to be more precise, among all the functions which satisfy the first condition, we find the function $\rho'(x)$ which is the closest to $\rho^{(k-1)}(z)$;
- then, we project the function $\rho'(z)$ onto the set of all functions which satisfy the second condition; to be more precise, among all the functions which satisfy the second condition, we find the function $\rho^{(k)}(x)$ which is the closest to $\rho'(z)$.

We continue this process until it converges.

As the distance between the two functions $f(z)$ and $g(z)$ – describing how close they are – it is reasonable to take the natural analog of the Euclidean distance: $d(f,g) \stackrel{\text{def}}{=} \sqrt{\int (f(z) - g(z))^2\, dz}$. One can check that for this distance function:

- the closest function in the first part of the iteration is the function $\rho'(z) = \max(0, \rho^{(k-1)}(z))$, and
- on the second part, the function whose Fourier transform is equal to

$$F^{(k)}(\omega) = \frac{\sqrt{|D(\omega)|}}{|F'(\omega)|} \cdot F'(\omega).$$

Thus, we arrive at the following algorithm.

How to Use Fourier Techniques to Solve the Practical Question: Algorithm. We start with an arbitrary function $\rho^{(0)}(z)$. On the k-th iteration, we start with the function $\rho^{(k-1)}(z)$ obtained on the previous iteration, and we do the following:

- first, we compute $\rho'(z) = \max(0, \rho^{(k-1)}(z))$;
- then, we apply Fourier transform to $\rho'(z)$ and get $F'(z)$;
- after that, we compute $F^{(k)}(\omega) = \dfrac{\sqrt{|D(\omega)|}}{|F'(\omega)|} \cdot F'(\omega)$;
- finally, as the next approximation $\rho^{(k)}(z)$, we take the result of applying the inverse Fourier transform to $F^{(k)}(\omega)$.

We continue this process until it converges.

Acknowledgment. This work was supported in part by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070), by the National Science Foundation grants HRD-0734825, HRD-1242122, DUE-0926721, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from the National Institutes of Health, and by a grant N62909-12-1-7039 from the Office of Naval Research.

References

1. Aubinet, M., Vesala, T., Papale, D. (eds.): Eddy Covariance – A Practical Guide to Measurement and Data Analysis. Springer, Hiedelberg (2012)
2. Hole, J.A.: Nonlinear high-resolution three-dimensional seismic travel time tomography. Journal of Geophysical Research 97, 6553–6562 (1992)
3. Ochoa, O., Velasco, A., Servin, C.: Towards Model Fusion in Geophysics: How to Estimate Accuracy of Different Models. Journal of Uncertain Systems 7(3), 190–197 (2013)
4. Ochoa, O., Velasco, A.A., Servin, C., Kreinovich, V.: Model Fusion under Probabilistic and Interval Uncertainty, with Application to Earth Sciences. International Journal of Reliability and Safety 6(1-3), 167–187 (2012)
5. Rabinovich, S.: Measurement Errors and Uncertainties: Theory and Practice. American Institute of Physics, New York (2005)
6. Servin, C., Ochoa, O., Velasco, A.A.: Probabilistic and interval uncertainty of the results of data fusion, with application to geosciences. In: Abstracts of 13th International Symposium on Scientific Computing, Computer Arithmetic, and Verified Numerical Computations SCAN 2008, El Paso, Texas, September 29-October 3, p. 128 (2008)

Automatic Sintonization of SOM Neural Network Using Evolutionary Algorithms: An Application in the SHM Problem

Rodolfo Villamizar, Jhonatan Camacho, Yudelman Carrillo, and Leonardo Pirela

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones,
Universidad Industrial de Santander. Grupo CEMOS. UIS. E3T.
Carrera 27, calle 9, Ciudad UniversitariaBucaramanga, Colombia
Rovillam@uis.edu.co, jhonatan.uis@gmail.com,
{yudelman,Leanpimu}@hotmail.com

Abstract. This paper is a contribution to the Structural Health Monitoring problem, solved by using case based reasoning and Self Organizing Maps. The expert system described in this paper is able to detect, locate and quantify stiffness percentage changes in a mechanical engineering structure. In order to overcome issues relating large number of parameters involved in the training stage it was applyed differential evolutive algorithms. Proper indexes to evaluate the training quality were proposed in order to increase diagnosis reliability. The algorithms were tested using the UBC ASCE Benchmark. The numerical implementation shows decreasing in the identification errors with respect to those obtained by selecting manually network training parameters.

Keywords: Structural Health Monitoring, Case-Based-Reasoning, Differential Evolutionary Algorithm.

1 Introduction

The objective in Structural Health Monitoring (SHM) is to detect, locate and quantify damages in order to predict the useful life in engineering structures [1]. In this sense, several methodologies have been developed in recent years [2-5]. They use digital signal processing techniques and artificial intelligence algorithms to identify changes in structure models. The application presented in this paper allows the identification of percentual stiffness changes in mechanical structures. The methodology is an extension of previous work [6], and is based on discrete wavelet transform (DWT), principal component analysis (PCA), case based reasoning (CBR) and self organizing maps (SOM). The contribution of the present work is a procedure which allows automatic tuning of the algorithms via differential evolutive approaches. This procedure allows overcoming isues related with the tuning of an expert system which require high experience. Automatic tuning reduces the poor performance of expert systems tuned with heuristic rules.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_33,

2 Structural Health Monitoring Algorithm

Figure 1 shows the scheme used for damage identification.

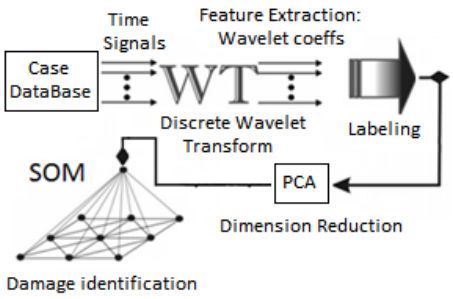


Fig. 1. Damage identification procedure

The steps in order to identify critical changes in a structure are the following:

i. Case DataBase Construction: It is a memory array which contains full dynamic simulations of the model structure. They describe damage patterns and facilitate the search of similar damages. When a new record is sensed it is posible to classify its severity using information previously storaged. The similarity measure is commonly euclidean distance. In figure 1, the SOM evaluate this similarity **[7]**.
ii. Apply Discrete Wavelet Transform to time signals of acceleration sensors located in some elements of the structure. The resulting detail and approximation wavelet coefficients are the feature vector.
iii. Reduce the feature vector using Principal Component Analysis in order to extract the most relevant information.
iv. Obtain a diagnosis by analogy using Case-Based Reasoning [7]. In this sense a SOM neural network is used.

For a more detailed description of the methodology the reader is refered to a previous work [6]

2.1 Benchmark Case

The mechanical structure used for numerical validation corresponds to the UBC *Benchmark* (see figure 2). It is a steel *frame* with 4 floors located at the *British Columbia University*, Canada. The ASCE facilitates a numerical model with 120 freedom degrees [4]. Each floor of the model has 9 columns (vertical elements), 12 beams (horizontal elements) and 8 braces (diagonal elements) which sum 116 finite elements. Each floor contains 29 elements (columns, beams and *Braces).*

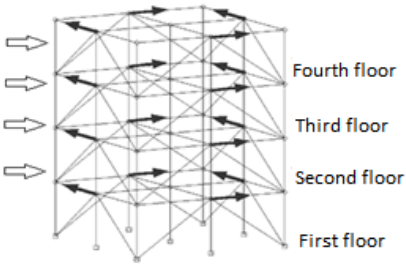


Fig. 2. Benchmark structure

Damage detection consist in identify variations of the nominal matriz stifness. In this sense, the objective of the expert systems is to know if the stifness of a specific element (beam, column or brace) suffered alterations compared with its nominal value. It is possible recording 16 acceleration signals. The expert system also classifies the element with damage in the respective type (beam, column or brace) and locates the floor of the element with damage.

The cases DataBase are simulations with percentual stiffness variations in the range [0, 50%] taking into account 1, 2, 3 or 4 elements simultaneously. These variations are represented by the 16 acceleration records. Th estructure was excited using whithe noise and a total of 6500 cases were stored.

2.2 Error Indexes

The following error indexes were estimated in order to evaluate the algorithm performance:

- **Dimension error:** This index relates the number of elements with damage, where $Dim_{máx}$ has a value of 4.

$$\%(error_{Dim}) = \left|\frac{Dim_{Est} - Dim_{Real}}{Dim_{máx}}\right| \times 100 \quad (1)$$

- **Severity error:** This index takes into account the magnitude of the percentual stifness variation. It is calculated using the average estimations for each element. It is defined by:

$$\%(error_{Sev}) = \sum_{i=1}^{n} \left|\frac{Sev_{Esti}(i) - Sev_{Real}(i)}{Sev_{Real}(i)}\right| / n \times 100 \quad (2)$$

- **False Positive error:** A false positive occurs when the expert system detects damage and actually there is no damage.

$$\%(error_{FP}) = \frac{\#Positive\ false}{\#Cases\ without\ damage} \times 100 \quad (3)$$

- **False Negative error:** A false negative occurs when the expert system does not detect damage and actually there is a damage:

$$\%(error_{FN}) = \frac{\#Negative\ False}{\#Case\ with\ damage} \times 100 \quad (4)$$

- **Type error:** Corresponds to the nmber of elements which were classified wrongly on a specific type (column, beam or brace).

- **Floor error:** It is the number of elements located wrongly in the respective floor.

- **Element error:** Take into account the error relating a specific element.

3 Automatic Tuning of SOM Neural Network

This section is focused on describe the procedure in order of automatically tuning SOM neural networks. In this sense, an evolutive diferential algorithm (ED) is implemented. When a SOM neural network is trained, it is necessary to select the following parameters:

- **Normalization method:** It is necesary to normaliza de data input between different options. Data normalization avoids false dominant clusters.

- **Output neurons number:** It is the *clusters* number, according to [6-11] it is recommended to use a range between $5\sqrt{n}$ and $\frac{9}{10}n$, where n is the number of training cases.

- **Grid structure and Map shape:** it is the local topology map.

- **Neighborhood function:** it is the interation between reference vectors and affects the precision and generalization of the SOM network.

SOM quality is commonly described by the following indexes [12, 13]:

- **Topographical error:** It is a measurement of topology preservation [14]. It should be near to zero.

- **Distortion:** Shows how well each neuron represents the input data [11].

- **Histogram uniformity:** It is a measurement of the cases distribution in the clusters. Ideally, each cluster should be containing cases of the same type and there is not be empty clusters.

3.1 Evolutive Diferential Algorithm

In order to tune automatically the SOM network, the evolutive diferential algorithm minimizes a weigthed averaged of the SOM quality and error indexes.

$$f(\vec{x}_{i,j}) = \sum_{i=1}^{n} w_i * e_i + DS \quad (5)$$

Where w_i: weight factor, e_i: error indexes, DS: estándar deviation of cross validation.

Diferential evolutive algorithm (ED) was proposed by Storm y Price [8-9] in 1998. It is a technique base don the evolution of real valued vectors (poblation - individuals) which are posible solutions in the search space. The difference between ED and other Evolutive algorithms (AEs) lies in the use of linear combinations of individuals of the current population [15-16]. Figure 3 details the ED algorithm sequence.

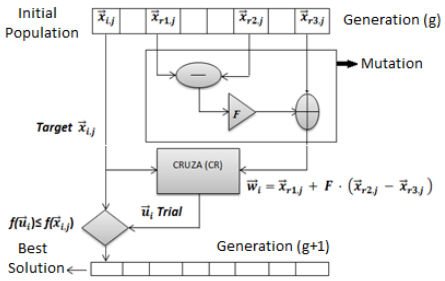


Fig. 3. Evolutive diferential algorithms

3.2 Individual Codification

The first step in the ED algorithm is to codify each individual of the poblation. Each gene of the chromosome corresponds to one SOM tuning parameter. The figure 4 shows a graphical representation for the chromosome of one individual.

$$X_{i,j} = [g_1 \quad g_2 \quad \cdots \quad g_D]$$

Figure 4 Graphical representation of an individual chromosome.

Table 1 describes the posible parameters in the tunning stage for a neural network.

Table 1. Parameter codification

Parameter	Options	Code	Parameter	Options	Code
Normalization method (g_1)	Variance	1	Grid structure (g_3)	Hexagonal	1
	Linear	2		Rectangular	2
	Logaritmic	3	Output Neurons (g_4)	Random value	X
	Logistic	4			
Map shape (g_2)	Laminate	1	Neiborhood function (g_5)	Gaussian	1
	Cylindrical	2		Cut Gaussian	2
	Thoroidal	3		Bubble	3

3.3 Initial Population

The initial population (pop) is a matriz of size [NPxD]. The rows (NP) are the number of individuals and the columns (D) are the chromosome length. Each individual of the population is generated in a random way taking into account the codification rules expressed in table1.

3.4 Mutation

The mutation operator uses three individuals of the population selected randomly. One ($\vec{x}_{r1,j}$) will be the base vecor and the rest ($\vec{x}_{r2,j}$ and $\vec{x}_{r3,j}$) the differential vectors [10]. This operator adds the proportional difference between $\vec{x}_{r2,j}$ and $\vec{x}_{r3,j}$ to $\vec{x}_{r1,j}$ in order to obtain the new individual mutated ($\vec{w}_i$):

$$\vec{w}_i = \vec{x}_{r1,j} + F \cdot \left(\vec{x}_{r2,j} - \vec{x}_{r3,j}\right) \quad (6)$$
$$\text{r1, r2, r3} \in \{1, 2, \ldots, NP\}$$

F [0.4 y1] is the constant mutation and it establishes a diferentiation range between the individuals' x_{r2} and x_{r3}. This constant helps with convergence issues.

3.5 Recombination

After mutation, it is applied the parents crossing operator over each individual $\vec{x}_{i,j}$ (target) in order to generate a trial individual $\vec{u}_i$. This intermédiate individual $\vec{u}_i$ is created mixing the components of $\vec{w}_i$y $\vec{x}_i$ using a probability factor Cr ∈ [0, 1].

$$\vec{u}_i = \begin{cases} \vec{w}_{i,(j)} \; si \; r \leq Cr; \\ \vec{x}_{i,(j)} \; en \; otro \; caso \end{cases} \quad (7)$$

In (7), r is a random value [0 – 1]. If r is less than the cross coefficient (CR), the j position of the mutated individual $\vec{w}_{i,(j)}$ is changed with the j element of the al $\vec{u}_{i,(j)}$ [10].

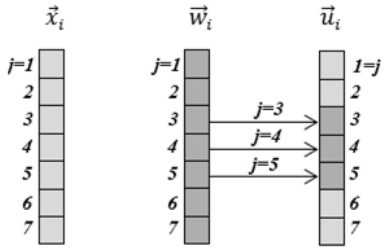


Fig. 4. Cruza operator

The cross coefficient CR is selected in a random way. [15-17].

3.6 Survivers Selection

The selection operator is evaluated over the *target* ($\vec{x}_{i,g}$) and *trial* ($\vec{u}_{i,g}$) vectors to decide which individuals will make part of the new eneration ($\vec{x}_{i,g+1}$). The selection operator evaluates the fitness function $f(\vec{x}_i)$:

$$x_{i,g+1} = \begin{cases} u_{i,g} f(\vec{u}_i) \leq f(\vec{x}_i), \\ \vec{x}_{i,g} \; otherwise \end{cases} \tag{8}$$

3.7 Terminal Condition

Following aspects are taking into account to stop iterations in the ED algorithm: *Fitness* reach an average value and Maximun numbers of iterations.

Empirically was found that 20 iterations and 6.5% variations of the fitness value works fine for the application. More tan 20 iterations require simulations time hihger than 48 hours.

4 Numerical Results

Figure 5 shows the fitness value after 20 generations and the SOM parameters for the best individual are $X_{i,j} = [g_1 = 4, \quad g_2 = 1, \quad g_3 = 2, \quad g_4 = 1800, \; g_4 = 3]$.

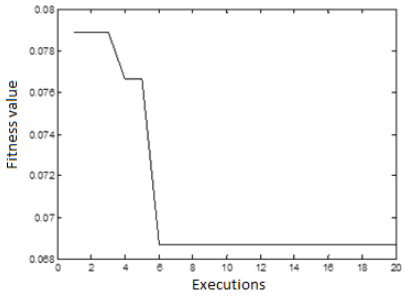


Fig. 5. Fitness evolution

The error indexes values using 3 cross validation groups are shown in table 2.

In table 2 *errFl* and *errTy* correspond to the floor and type error classification respectively; whereas FN and FP are the negative and false positve errors. The indexes nl, nh, and nv correspond to clusters with low, high and empty number of cases using such a threshold the average between the number of case and the neurons in the SOM. Automatic tuning of SOM network is slightly better than using default parameters.

Figure 6 shows the average errors for the location of the element and the severity of damage using automatic tunning.

Table 2. Upper: Error indexes for best individual. Lower: Error indexes using default parameters.

Group 1		Group 2		Group 3	
FP	**FN**	**FP**	**FN**	**FP**	**FN**
35	0	44	4	38	0
errTy	errFl	errTy	errFl	errTy	errFl
0.016	0.074	0.016	0.069	0.018	0.070
SOM Quality					
Nl		**nh**		**nv**	
0.4124		0.5876		0.4124	
0.4177		0.5823		0.4177	
0.4186		0.5814		0.4186	

Group 1		Group 2		Group 3	
FP	**FN**	**FP**	**FN**	**FP**	**FN**
4	31	8	33	8	34
errTy	errFl	errTy	errFl	errTy	errFl
0.06	0.2	0.06	0.18	0.06	0.18
SOM Quality					
nl		**nh**		**nv**	
0.28		0.73		0.27	
0.27		0.73		0.27	
0.26		0.74		0.26	

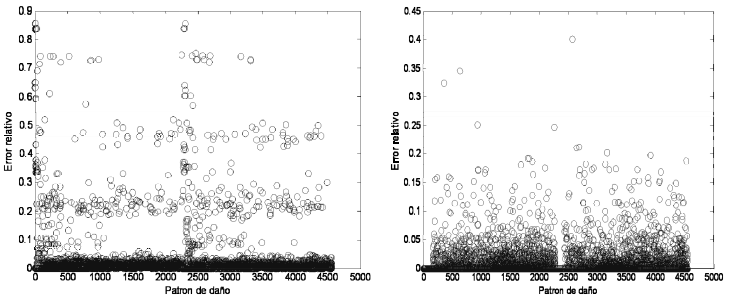


Fig. 6. Left: mean Element Error. Rigth: Mean severity error

Figure 7 shows error indexes relating the SOM quality.

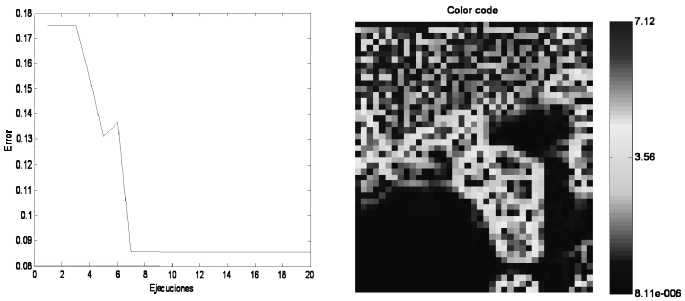


Fig. 7. Left: Topographical Error. Rigth: Distorsion measurement Error

According to figure 7, the ED algorithm decrease the topographical error for sucesive iterations, and distortion measurements are near toz ero (Average distortion is E =0.7664). It is a signal than ED improves the SOM quality.

5 Conclusions

Tuning automatically the SOM neural network improve its quality measurements. However it requires high computational cost even when almost 40% of clusters are empty. Also, the identification errors are slightly low than using default parameters for the SOM.

In the fitness function the weigthing was the same and all errors decrease without dominant tendencies. For future works is possible study the effect of use different weigths, for example, give most importance to the floor and type errors.

The most critical parameter in the SOM training is the clusters numbers whereas parameters relating the shape map and grid structure are not relevant for the damage detection problem previousled exposed. Also, it was found that cut Gaussian neighborhood function have poor performance.

It is possible to include damage location and quantification for the ASCE numerical benchmark using CBR approach with automatic tuning of SOM stage.

References

1. Quintero, A., Villamizar, R.: Estado del arte en monitorización de salud estructural: Un enfoque basado en agentes inteligentes. Ciencia e Ingeniería Neogranadina 20-1, 117–132 (2010)
2. Giraldo, D.: A Structural Health Monitoring Framework for Civil Structures. Doctoral Dissertation thesis. Washington, University (2006)
3. Caicedo, J.M., Johnson, E.A.: Structural Health Monitoring of Flexible Civil Structures. Doctoral Dissertation thesis. Washington, University (2003)
4. Structural Control and Earthquak Engineering Laboratory. Washington University. Disponible en internet: Consultado Julio (2013), `http://mase.wustl.edu/wusceel/quake/`

5. Marulanda, J., Thomson, P., Caicedo, J.M.: Monitoreo de Salud Estructural de Puentes Metálicos. In: Proceedings of the First Colombian Workshop on Steel Structures, Cartagena, Colombia, September 26-28 (2001)
6. Camacho, J., Villamizar, R.: Structural Health Monitoring by Using Pattern Recognition. International Journal of Artifitial intelligence – IJAI 5(A10) (Autum 2010) ISSN 0974-0635
7. Sainz, H.: DSMus Razonamiento basado en casos CBR. Obtenido de (2008), http://www.imaisoftware.com/openlab/data/proyectos/DSMus/presentacion.pdf
8. Storn, R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization overcontinuous spaces. J. of Global Optimization 11(4) (1997)
9. Price, K.V., Storn, R.M., Lampinen, J.A.: Differential Evolution A Practical Approach to Global Optimization Natural Computing Series. Springer, Berlin (2005)
10. Kenneth, V.P., Rainer, M.S., Jouni, A.: Lampinen.DifferentialEvolution A Practical Approach to Global Optimization. Natural Computing Series. Springer, Berlin (2005)
11. Kohonen, T.: Self-Organizing Maps, vol. 30. Springer (2001)
12. Marín, J.M.: Los mapas auto-organizados deKohonen. Universidad Carlos III de Madrid (2009)
13. Laboratory of Computer and Information Science (CIS) "Adaptive Informatics Research Centre". Department of Computer Science and Engineering at the Hel-sinki University of Technology. Citado Julio de (2013), http://www.cis.hut.fi/somtoolbox/
14. Marín, J.M.: Introducción a las Redes Neuronales Aplicadas. Curso de Expertos de U.C.M (consultado 6 de febrero de (2013), http://halweb.uc3m.es/esp/Personal/personas/jmmarin/esp/DM/tema3dm.pdf
15. Santana, L.V.: Un Algoritmo Basado en Evolución Diferencialpara Resolver Problemas Multiobjetivo. Tesis de maestría. Centro de Investigación y de Estudios Avanzadosdel Instituto Politécnico Nacional Departamento de Ingeniería Eléctrica Sección de Computación, México DF. Noviembre de (2004)
16. Monterrosa, C.: Analizando la Relación entre elFactor de Escalamiento y los-Tamaños de Población en elAlgoritmo de EvoluciónDiferencial al Resolver Pro-blemasde Optimización con Restricciones. Dirección General de Educación Superior Tecnológica Instituto Tecnológico de Tuxtla Gutiérrezdepartamento de Ingeniería en Sistemas Computacionales. México, Enero de (2010)
17. Swagatam, D., Ajith, A.: Automatic Clustering Using an Improved Diffe-rential Evolution Algorithm (2008), http://www.softcomputing.net/smca-paper1.pdf (Consultado 6 de febrero de 2013)

Density-Based Fuzzy Clustering as a First Step to Learning Rules: Challenges and Solutions

Gözde Ulutagay[1] and Vladik Kreinovich[2]

[1] Department of Industrial Engineering, Izmir University, Uckuyular-Izmir, Turkey
gozde.ulutagay@gmail.com
[2] University of Texas at El Paso, El Paso, TX 79968, USA
vladik@utep.edu

Abstract. In many practical situations, it is necessary to cluster given situations, i.e., to divide them into groups so that situations within each group are similar to each other. This is how we humans usually make decisions: instead of taking into account all the tiny details of a situation, we classify the situation into one of the few groups, and then make a decision depending on the group containing a given situation. When we have many situations, we can describe the probability density of different situations. In terms of this density, clusters are connected sets with higher density separated by sets of smaller density. It is therefore reasonable to define clusters as connected components of the set of all the situations in which the density exceeds a certain threshold t. This idea indeed leads to reasonable clustering. It turns out that the resulting clustering works best if we use a Gaussian function for smoothing when estimating the density, and we select a threshold in a certain way. In this paper, we provide a theoretical explanation for this empirical optimality. We also show how the above clustering algorithm can be modified so that it takes into account that we are not absolutely sure whether each observed situation is of the type in which we are interested, and takes into account that some situations "almost" belong to a cluster.

1 Formulation of the Challenges

Clustering Is How We Humans Make Decisions. Most algorithms for control and decision making take, as input, the values of the input parameters, and transform them into the optimal decision (e.g., into an optimal control value). Humans rarely do that. When facing a need to make a decision – e.g., where to go eat, which car or which house to buy, which job offer to accept – we rarely write down all the corresponding numbers and process them. Most of the time, for each input variable, instead of its exact known value, we only use a category to which this value belongs. For example, to decide where to eat, instead of the exact prices of different dishes, we usually base our decision on whether the restaurant is cheap, medium, expensive, or very expensive. Instead of taking into account all the menu details, we base our decision on whether this restaurant can be classified as Mexican, Chinese, etc. Similarly, when we select a hotel to

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_34,

stay during a conference, instead of taking into account all the possible features, we base our decision on how many stars this hotel has and whether it is walking distance, close, or far away from the conference site.

In all such cases, before we make decisions, we *cluster* possible situations, i.e., divide them into a few groups – and then make a decision based on the group to which the current situation belongs.

Clustering Is a Natural First Step to Learning the Rules. Humans are often good at making decisions. In many situations – such as face recognition – we are much better than the best of the known computer programs, in spite of the fact that computers process data much faster than we humans. To improve the ability of computers to solve problems, it is therefore reasonable to emulate the way we humans make the corresponding decisions. This means, in particular, that a reasonable way to come up with a set of good-quality decision rules is to first cluster possible situations, and then make a decision based on the cluster containing the current situation.

Clustering: Ideal Case. How shall we cluster? In order to cluster, we need to have a set of situations, i.e., vectors $x = (x_1, \ldots, x_n)$ consisting of the values of n known quantities that characterize each situation.

Let us first consider the case when we have so many examples that in the vicinity of each situation $x = (x_1, \ldots, x_n)$, we can meaningfully talk about the *density* $d(x)$ of situations in this vicinity – i.e., the number of situations per unit volume.

In the ideal case, when all situations belong to several clearly distinct clusters, there are no examples outside the clusters – so the density outside the clusters is 0. Within each cluster, the density $d(x)$ is positive. Different clusters can be distinguished from each other because each cluster is connected. So, in this ideal case, one we know the density $d(x)$ at each point x, we can find each cluster as the connected component of the set $\{x : d(x) > 0\}$.

Clustering: A More Realistic Case. In practice, in addition to objects and situations which clearly belong to different clusters, there are also "weird" situations that do not fall under any meaningful clusters. For example, when we make a medical decision, we classify all the patients into a few meaningful groups – e.g., coughing and sneezing patients can be classified into patients with cold, patients with allergy, patients with flu, etc. However, there may be some exotic diseases which also cause sneezing and coughing, diseases which are not present in the current sample in sufficient numbers.

Such not-easy-to-classify examples can occur for every x. Let d_a be the average density of such examples. In this case, if at some point, the observed density $d(x)$ is smaller than or equal to d_a, then most probably all examples with these parameters are not-easy-to-classify, so they do not belong to any of the clusters that we are trying to form. On the other hand, if for some point x, the observed density $d(x)$ is much larger than d_a, this means that all these examples cannot

come from not-easy-to-classify cases: some of these example come from one of the clusters that we are trying to form.

This idea leads us to the following clustering algorithm: we select a threshold t, and then find each cluster as the connected component of the set $\{x : d(x) \geq t\}$.

How to Estimate the Density $d(x)$. In practice, we only have a finite set of examples $x^{(1)}$, $x^{(2)}$, ..., $x^{(N)}$. In order to apply the above approach, we must use the observed values $x^{(j)}$ to estimate the density $d(x)$ at different values x.

One possible answer to this question comes from the fact that usually, dues to inevitable measurement inaccuracy, the measured values $x^{(j)} = (x_1^{(j)}, \ldots, x_n^{(j)})$ are not exactly equal to the actual (unknown) values $x_1^{(j),\mathrm{act}}, \ldots, x_n^{(j),\mathrm{act}}$ of the corresponding quantities; see, e.g., [7]. If we know the probability density function $\rho(\Delta x)$ describing the measurement errors $\Delta x \stackrel{\mathrm{def}}{=} x - x^{\mathrm{act}}$, then for each j, we know the probability density of the corresponding actual values: $\rho(x^{(j)} - x^{(j),\mathrm{act}})$.

So, if we only have one observation $x^{(1)}$, it is reasonable to estimate the density of different situations x as $d(x) = \rho(x^{(1)} - x)$. When we have N observations $x^{(1)}$, $x^{(2)}$, ..., $x^{(N)}$, it is reasonable to consider them all equally probable, i.e., to assume that each of these observations occurs with probability $p(x^{(j)}) = 1/N$. Thus, due to the formula of the full probability, the probability $d(x)$ of having the actual situation x can be computed as

$$d(x) = p(x^{(1)}) \cdot \rho(x^{(1)} - x) + \ldots + p(x^{(N)}) \cdot \rho(x^{(N)} - x) = \frac{1}{N} \cdot \sum_{j=1}^{N} \rho(x^{(j)} - x).$$

The above formula is known as the *Parzen window*; see, e.g., [8]. The corresponding function $\rho(x)$ is known as a *kernel*. As a result, we arrive at the following algorithm.

Resulting Clustering Algorithm. At first, we select a function $\rho(x)$. Then, based on the observed examples $x^{(1)}$, $x^{(2)}$, ..., $x^{(N)}$, we form a density function $d(x) = (1/N) \cdot \sum_{j=1}^{N} \rho(x^{(j)} - x)$. After that, we select a threshold t, and we find the clusters as the connected components of the set $\{x : d(x) \geq t\}$.

Algorithmic Comment. In practice, we can only handle a finite number of possible points x, so we perform computations only for finitely many x – e.g., for all points x from a dense grid. In the discrete case, the subdivision into connected components is equivalent to finding a transition closure of the direct neighborhood relation – and it is well known how to efficiently compute the transitive closure of a given relation; see, e.g., [1].

Discussion: Beyond Probabilities and Measurements. In the above text, we describe a statistical motivation for this algorithm. It turns out (see, e.g., [2,3]) that a fuzzy approach leads, in effect, to the same algorithm – in this case, instead of the probability density $\rho(x)$, we must take the membership function

describing the neighborhood relation, and the Parzen window formula for $d(x)$ describes not the total probability, but rather (modulo an irrelevant $1/N$ factor) the fuzzy cardinality of the fuzzy set of all examples in the neighborhood of the given point x. This fuzzy approach makes sense, e.g., when the sample values $x^{(j)}$ come not from measurements, but from expert observations.

Empirical Results. By testing different possible selections, we found out [2,3] that empirically:

- The best kernel is the Gaussian function $\rho(x) \sim \exp(-\text{const} \cdot x^2)$.
- To describe the best threshold, we must describe, for each possible threshold t, the interval formed by all the threshold values t' that lead to the same clustering of the original points $x^{(j)}$ as t. Then, we select the threshold t for which this interval is the widest – i.e., for which clustering is the most robust to the threshold selection.

1st Challenge: Explain the Above Empirical Results. Our 1st challenge is to provide a theoretical explanation for these empirical results.

2nd Challenge: Some Observations May Be Erroneous. In the above analysis, we assumed that all the situations that we observed and/or measured are exactly of the type in which we are interested. In other words, we assume that the only uncertainty is that the measurement values are imprecise. In reality, about some measurements, we are not sure whether the corresponding situations are of the desired type or not.

For example, when we analyze the animals that we observed in the wild, not only are our measurements not absolutely accurate, but in addition to this, in some cases, we are not sure whether we actually observed an animal or it was just a weird combination of shadows that made it look like an animal.

It is desirable to take this additional uncertainty into account during clustering.

Comment. This additional uncertainty was recently emphasized by L. Zadeh, when he promoted the idea of a *Z-number* [9], a number for which there are two types of uncertainty: an uncertainty in *value* – corresponding to the accuracy of the measuring instrument, and an uncertainty in whether we did measure anything meaningful at all – corresponding, e.g., to *reliability* of the measuring instrument.

3rd Challenge: Need for Fuzzy Clustering Results. The above algorithm provides a crisp (non-fuzzy) division into clusters. In real life, in some cases, we may indeed be certain that a given pair of objects belongs to the same cluster (or to different clusters); however, in many other cases, we are not 100% sure about it. It is desirable to modify the above clustering algorithm in such a way that it reflects this uncertainty. In other words, we *fuzzy* clusters, clusters in which some situations x are assigned to different clusters with different degrees of certainty.

The existing fuzzy-techniques-based clustering algorithms provide such classification (see, e.g., [4,6]); it is therefore reasonable to modify the above density-based fuzzy-motivated algorithm so that it will produce a similar fuzzy clustering.

4th Challenge: Need for Hierarchical Clustering. In practice, our classification is hierarchical. For example, to make a decision about how to behave when we see an animal in the forest, we first classify animals into dangerous and harmless ones. However, once we get more experience, we realize that different dangerous animals require different strategies, so we sub-classify them into subgroups with a similar behavior: snakes, bears, etc.

It is therefore desirable that our clustering algorithm have the ability to produce such a hierarchical clustering: once we subdivided the original situations into clusters, we should be able to apply the same clustering algorithm to all the situations within each cluster c and come up with relevant sub-clusters. Alas, this does not always happen in the above density-based clustering algorithm. Indeed, we select a threshold for which the corresponding interval is the widest. Thus, it is highly possible that within a cluster, we will have the same intervals – so the new sub-classification will be based on the same threshold and thus, it will return the exact same cluster. The 4th – and the last – challenge is to modify the above algorithm so that it will enable us to produce the desired hierarchical clustering.

In this paper, we propose possible solutions to all these challenges.

2 Solutions to Challenges

A Solution to the 1st Part of the 1st Challenge. We need to explain why empirically, the Gaussian membership functions – or, equivalently, the Gaussian kernels $\rho(\Delta x)$ – are empirically the best.

Case of Measurements. This empirical fact is reasonably easy to explain in the case when the values $x^{(j)}$ come from measurements, and the probability density $\rho(\Delta x)$ corresponds to the probability density of the measurement errors.

In this case, the empirical success of the Gaussian kernels can be easily explained by another (better known) empirical fact: that the Gaussian distribution of the measurement error is indeed frequently occurring in practice. This new empirical fact, in its turn, has a known explanation:

- a measurement error usually consists of a large number of small independent components, and,
- according to the Central Limit theorem, the distribution of the sum of a large number of small independent components is indeed close to Gaussian (see, e.g., [8]) – the more components, the closer the resulting distribution is to Gaussian.

Case of Expert Estimates. As we have mentioned, the values $x^{(j)}$ often come not from measurements, but from expert estimates. In this case, the inaccuracy of these estimates is also caused by a large number of relatively small independent factors. Thus, we can also safely assume that the corresponding estimation errors are (approximately) normally distributed.

Alternative Explanation. The whole idea of the above density estimates can be reformulated as follows: we start with the discrete distribution $d_N(x)$ in which we get N values $x^{(j)}$ with equal probability, and then we "smoothen" this original distribution – by taking a convolution between $d_N(x)$ and a kernel function $\rho(x)$: $d(x) = \int d_N(y) \cdot \rho(x - y)\, dy$.

In the previous text, we mentioned that the empirically best choice of the kernel function is a Gaussian function, but what we did not explicitly mention is that even after we fix the class of Gaussian functions, we still to find an appropriate parameter – the half-width of the corresponding Gaussian distribution. An appropriate selection of this parameter is important if we want to achieve a reasonable clustering:

- on the one hand, if we select a very narrow half-width, then each original point $x^{(j)}$ becomes its own cluster;
- alternative, if we select a very wide half-width, then all the density differences will be smoother out, and we will end up with a single cluster.

The choice of this half-width is usually performed empirically: we start with a small value of half-width and gradually increase it.

In principle, every time we slightly increase the half-width, we could go back to the original discrete distribution and apply the new slightly modified kernel function. However, since the kernel functions are close to each other, the resulting convolutions are also close to each other. So, it is more computationally efficient, instead of starting with the original discrete distribution, to apply a small modifying convolution to the previous convolution result.

In this approach, the resulting convolution is the result of applying a large number of minor convolutions, with modification kernel functions which change the function very slightly – i.e., which are close to the delta-function convolution with which does not change the original function at all. How can we describe the composition of such large number of convolutions?

From the mathematical viewpoint, each modification kernel function $K(x)$ can be viewed as a random variable whose probability density function proportional to $K(x)$. The delta-function kernel – that does not change anything – corresponds to the random variable which is equal to 0 with probability 1. A kernel corresponding to a small change thus corresponds to a random variable which is close to 0 – i.e., which is small.

It is well known that the probability distribution $\rho(X)$ of the sum $X = X_1 + X_2$ of two independent random variables is equal to the convolution of their probability density functions $\rho_1(X_1)$ and $\rho_2(X_2)$:

$$\rho(X) = \int \rho_1(X_1) \cdot \rho_2(X - X_1)\, dX_1.$$

Thus, applying several convolutions – corresponding to several small random variables – is equivalent to applying one convolution corresponding to the sum of these small random variables. Due to the Central Limit theorem, this sum is (almost) normally distributed. So, the corresponding probability density function is (almost) Gaussian, and the resulting convolution is (almost) the convolution with the Gaussian kernel.

A Solution to the 2nd Part of the 1st Challenge. The above clustering algorithm depends on the selection of an appropriate threshold t. It turns out that empirically, the following method of selecting this threshold works best: we select a threshold t for which the results of clustering the sample situations $x^{(j)}$ are the most robust with respect to this selection, i.e., for which the interval $\mathbf{t}(t)$ consisting of all threshold values t' that lead to the same clustering of the original situations as t is the widest.

How can we explain the empirical efficiency of this method?

Analysis of the Problem. The clustering of the sample situations is based on comparing the corresponding values $d(x^{(j)})$ with the threshold t. Thus, crudely speaking, the interval $\mathbf{t}(t)$ consists of all the values t' between the two sequential values $d(x^{(j)})$.

For simplicity, let us consider 1-D case. In this case, locally, the density function is monotonic, so the consequent values of the density $d(x^{(j)})$ are, most probably, attained at the two neighboring points $x^{(j)}$. (In multi-D case, if we use the local coordinates in which the gradient of the density is one of the directions, there are also additional dimensions that do not affect the density.)

In a sample of N points, the distance Δx to the next point can be found from the condition that there should be one point on this interval. By definition of the probability density, the probability to find the point on the intervals is equal to $d(x) \cdot \Delta x$. The total number of points is N, so the average number of points on the interval is $N \cdot d(x) \cdot \Delta x$. Thus, we have $N \cdot d(x) \cdot \Delta x \approx 1$ hence $\Delta x \approx 1/(N \cdot d(x))$. When we move from the original point x to the new point $x+\Delta x$, the density changes from $d(x)$ to $d(x)+\Delta x \cdot d'(x) \approx d(x)+d'(x)/(N \cdot d(x))$. Thus, the different between the two values if the threshold that lead to different clusterings – the desired gap – is proportional to the ratio $|d'(x)|/d(x)$. In multi-D case, we similarly have $\|\nabla d(x)\|_2/d(x)$, where $\nabla d \stackrel{\text{def}}{=} \left(\frac{\partial d}{\partial x_1}, \ldots, \frac{\partial d}{\partial x_n}\right)$ is the gradient vector, and for every vector $z = (z_1, \ldots, z_n)$, $\|z\|_2 \stackrel{\text{def}}{=} \sqrt{z_1^2 + \ldots + z_n^2}$ denotes its length.

After this reformulation, the question becomes: why, as an objective function, the ratio $\|\nabla d(x)\|_2/d(x)$ works the best? To answer this question, we will consider general reasonable optimality criteria which can be formulated in terms of the density function $d(x)$ and its gradient $\nabla d(x)$.

What We Need Is a Preference Relation. We do not necessarily need a numerical objective function that would enable us to compare two points x with two different values of $d(x)$ and $\nabla d(x)$. All we need is a preference relation $(d, z) \succeq (d', z')$

allowing us to compare two pairs (d, z) consisting of a real number d and an n-dimensional vector z. The meaning of this relation is that the pair (d, z) is better than (or of the same quality as) (d', z') – as a point whose value $d(x)$ is used as a threshold.

Let us enumerate natural properties of this relation, and then see which relations satisfy all these properties.

Natural Algebraic Properties of the Preference Relation. To be able to always make a decision, we must require that for every two pairs (d, z) and (d', z'), we have either $(d, z) \succeq (d', z')$ or $(d', z') \succeq (d, z)$. In mathematical terms, this means that the relation is *linear* or *total*.

Of course, since any pair (d, z) is of the same quality as itself, we must have $(d, z) \succeq (d, z)$ for all d and z. In mathematical terms, this means that the relation is *reflexive*.

This relation must also be *transitive*: indeed, if (d, z) is better than (or of the same quality as) (d', z'), and (d', z') is better than (or of the same quality as) (d'', z''), then (d, z) should better than (or of the same quality as) (d'', z'').

Closeness of the Preference Relation. Let us assume that $(d_n, z_n) \to (d, z)$, $(d'_n, z'_n) \to (d', z')$, and $(d_n, z_n) \succeq (d'_n, z'_n)$ for all n.

Since all the measurements are imprecise, this implies that for any given measurement error, for sufficiently large n, the pair (d, z) is indistinguishable from (d_n, z_n): $(d, z) \approx (d_n, z_n)$. Similarly, for sufficiently large n, the pair (d', z') is indistinguishable from the pair (d'_n, z'_n): $(d', z') \approx (d'_n, z'_n)$.

Thus, no matter how accurately we perform measurements, for the pairs (d, z) and (d', z'), there are indistinguishable pairs $(d_n, z_n) \approx (d, z)$ and $(d'_n, z'_n) \approx (d', z')$ for which $(d_n, z_n) \succeq (d'_n, z'_n)$. Hence, from the practical viewpoint, we will never be able to empirically conclude, based on measurement results, that $(d, z) \not\succeq (d', z')$. So, it is reasonable to conclude that $(d, z) \succeq (d', z')$.

In mathematical terms, this means that the relation $\succeq$ is *closed* in the topological sense.

Rotation Invariance. The components x_i of each situation $x = (x_1, \ldots, x_n)$ describe, e.g., spatial coordinates of some object, or components of the 3-D vector describing the velocity of this object. In all these cases, the specific numerical representation of the corresponding vector depends on the choice of the coordinate system. In most practical situations, the choice of a coordinate system is arbitrary: instead of the original system, we could select a new one which is obtained from the previous one by rotation. It is therefore reasonable to require that the preference relation not change if we simply rotate the coordinates. In other words, it is reasonable to require that if $(d, z) \succeq (d', z')$, and T is an arbitrary rotation in n-dimensional space, then $(d, T(z)) \succeq (d', T(z'))$.

First Result. From closeness and rotation invariance, we can already make an important conclusion about the preference relation. Let us formulate this first result in precise terms.

Definition 1

- *A relation $\succeq$ on a set A is called:*
  - linear *(or* total*) if for every two elements $a, a' \in A$, we have $a \succeq a'$ or $a' \succeq a$.*
  - reflexive *if $a \succeq a$ for all $a \in A$;*
  - transitive *if $a \succeq a'$ and $a' \succeq a''$ imply that $a \succeq a''$.*
- *Let $n \geq 1$ be an integer. By a* preference relation*, we mean a linear reflexive transitive relation $\succeq$ on the set of all pairs (d, z), where d is a non-negative real number and z is an n-dimensional vector.*
- *We say that a preference relation $\succeq$ is* closed *if for every two sequences $(d_n, z_n) \to (d, z)$ and $(d'_n, z'_n) \to (d', z')$ for which $(d_n, z_n) \succeq (d'_n, z'_n)$ for all n, we have $(d, z) \succeq (d', z')$.*
- *We say that a preference relation $\succeq$ is* rotation-invariant *if for every two pairs (d, z) and (d', z') and for every rotation T in n-dimensional space, $(d, z) \succeq (d', z')$ implies that $(d, T(z)) \succeq (d', T(z'))$.*

Proposition 1. *For every closed rotation-invariant preference relation $\succeq$, whether there is a relation $(d, z) \succeq (d', z')$ between the two pairs (d, z) and (d', z') depends only on the values d and d' and on the lengths $\|z\|_2$ and $\|z'\|_2$ of the vectors z and z', i.e., if $(d, z) \succeq (d', z')$, $\|z\|_2 = \|t\|_2$, and $\|z'\|_2 = \|t'\|_2$, then $(d, t) \succeq (d', t')$.*

Proof. Let us start with notations. Let us denote $a \equiv b$ if $a \succeq b$ and $b \succeq a$. This relation is clearly symmetric. Since the original relation $\succeq$ is reflexive and transitive, the new relation is also reflexive and transitive. In mathematical terms, reflexive symmetric transitive relations are called *equivalence relations*; thus, the above relation $\equiv$ is an equivalence relation.

One can easily check that $a \succeq b$ and $b \equiv c$ imply that $a \succeq c$, and that $a \equiv b$ and $b \succeq c$ also implies $a \succeq c$.

We plan to prove that for for every number d, for every vector z, and for every rotation T, we have $(d, z) \equiv (d, T(z))$.

Let us show that if we succeed in proving this, then the proposition will be proven. Indeed, since every two vectors of equal length can be transformed into each other by an appropriate rotation, this will mean that if $(d, z) \succeq (d', z')$, $\|z\|_2 = \|t\|_2$, and $\|z'\|_2 = \|t'\|_2$, then $(d, z) \equiv (d, t)$ and $(d', z') \equiv (d', t')$. From $(d, t) \equiv (d, z)$, $(d, z) \succeq (d', z')$, and $(d', z') \equiv (d', t')$, we will now be able to conclude that $(d, t) \succeq (d', t')$, i.e., exactly what we want to conclude in Proposition 1.

So, to complete our proof, it is sufficient to prove that for every axis ℓ and for every angle φ, the property $(d, z) \equiv (d, T_{\ell,\varphi}(z))$ holds, where $T_{\ell,\varphi}$ denoted a rotation by the angle φ around the axis ℓ.

We will first prove this statement for the case when $\varphi = 2\pi/k$ for some integer $k \geq 2$, i.e., when $k \cdot \varphi = 2\pi$.

Indeed, due to linearity of the preference relation $\succeq$, we have $(d, z) \succeq (d, T_{\ell,\varphi}(z))$ or $(d, T_{\ell,\varphi}(z)) \succeq (d, z)$. Without losing generality, let us consider the first case, when $(d, z) \succeq (d, T_{\ell,\varphi}(z))$.

In this case, rotation invariance implies that $(d, T_{\ell,\varphi}(z)) \succeq (d, T_{\ell,2\varphi}(z))$, that $(d, T_{\ell,2\varphi}(z)) \succeq (d, T_{\ell,3\varphi}(z))$, ..., and that $(d, T_{\ell,(k-1)\cdot\varphi}(z)) \succeq (d, T_{\ell,k\cdot\varphi}(z)) = (d, T_{\ell,2\pi}(z)) = (d, z)$.

Transitivity, when applied to $(d, T_{\ell,\varphi}(z)) \succeq (d, T_{\ell,2\varphi}(z)) \succeq \ldots \succeq (d, z)$, implies that $(d, T_{\ell,\varphi}(z)) \succeq (d, z)$. Since we already know that $(d, z) \succeq (d, T_{\ell,\varphi}(z))$, we conclude that $(d, z) \equiv (d, T_{\ell,\varphi}(z))$. The statement is proven.

Let us now prove the desired statement $(d, z) \equiv (d, T_{\ell,\varphi}(z))$ for the case when $\varphi = 2\pi \cdot (p/q)$ for some integers p and q.

Indeed, in this case, $\varphi = p \cdot \varphi(q)$, where we denoted $\varphi(q) \stackrel{\text{def}}{=} (2\pi)/q$. We already know, from Part 3.1 of this proof, that equivalence is preserved when we rotate by the angle $\varphi(q)$, i.e., that $(d, z) \equiv (d, T_{\ell,\varphi(q)}(z))$. Similarly, $(d, T_{\ell,\varphi(q)}(z)) \equiv (d, T_{\ell,2\varphi(q)}(z))$, ..., and, finally, that $(d, T_{\ell,(p-1)\cdot\varphi(q)}(z)) \equiv (d, T_{\ell,p\cdot\varphi(q)}(z)) = (d, T_{\ell,\varphi}(z))$. Thus, by transitivity of the equivalence relation, we conclude that indeed $(d, z) \equiv (d, T_{\ell,\varphi}(z))$. The statement is proven.

Let us now prove the desired statement $(d, z) \equiv (d, T_{\ell,\varphi}(z))$ for an arbitrary angle φ.

Indeed, every real number can be represented as a limit of rational numbers – e.g., its approximations of higher and higher accuracy. By applying this statement to the ratio $\varphi/(2\pi)$, we conclude that an arbitrary angle φ can be represented as a limit of the angles φ_n each of which has a form 2π times a rational number. For such angles, in Part 3.2 of our proof, we already proved that $(d, z) \succeq (d, T_{\ell,\varphi_n}(z))$ and $(d, T_{\ell,\varphi_n}(z)) \succeq (d, z)$. Due to closeness of the preference relation, we can now conclude that in the limit $\varphi_n \to \varphi$, we also have $(d, z) \succeq (d, T_{\ell,\varphi}(z))$ and $(d, T_{\ell,\varphi}(z)) \succeq (d, z)$, thus $(d, z) \equiv (d, T_{\ell,\varphi}(z))$.

The statement is proven, and so is the proposition.

Discussion. Based on Proposition 1, when we describe a preference relation, it is not necessarily to consider pairs consisting of a real number $d \geq 0$ and a vector z. Instead, it is sufficient to only consider two non-negative numbers: d and the length $l \stackrel{\text{def}}{=} \|z\|_2$ of the vector z. So now, we have a preference relation defined on the set of pairs of non-negative numbers d and l.

Monotonicity. If we have a homogeneous zone, i.e., a zone in which the density is constant and its gradient is 0, then this whole zone should belong to the same cluster. Selecting a threshold corresponding to this zone would mean cutting through this zone, which contradicts to the idea of clustering as bringing similar situations into the same cluster. From this viewpoint, it makes sense to dismiss pairs (d, l) for which $l = 0$: the optimal cut should never be at such pairs.

Similarly, points x with small gradient are probably not the best placed to cut. In other words, everything else being equal, situations with higher gradient (i.e., with larger values of l) are preferable as points used to determine a threshold.

With respect to density, as we have mentioned, the higher the density, the more probable it is that the corresponding values belong to the same cluster. Thus, everything else being equal, situations with lower density (i.e., with smaller values of d) are preferable to cut. From this viewpoint, it makes sense to dismiss

pairs (d, l) for which $d = 0$: if such ideal pairs are present, we have an ideal (no-noise) clustering (as we mentioned in the beginning of this paper), so there is no need for all these sophisticated methods. Thus, we arrive at the following definitions.

Definition 2

- *By a* non-zero preference relation*, we mean a linear reflexive transitive relation $\succeq$ on the set of all pairs (d, l) of positive real numbers.*
- *We say that a non-zero preference relation is* monotonic *if the following two conditions hold:*
  - *for every d and for every $l < l'$, $(d, l') \succeq (d, l)$ and $(d, l) \not\succeq (d, l')$;*
  - *for every l and for every $d < d'$, $(d, l) \succeq (d', l)$ and $(d', l) \not\succeq (d, l)$.*

Comment. The notion of closeness can be easily extended to this new definition.

Sub-samples. Instead of considering all possible situations, we may want to consider only part of them – this often happens in data processing when we want to decrease computation time. Of course, we need to select sub-populations in such a way that within each cluster, the relative density does not change. However, it is OK to select different fractions of sample in different clusters. For example, if some cluster contains a large number of different situations, it makes sense to select only some of them, while for another cluster which consists of a few situations, we cannot drastically decrease this number since otherwise, we will not have enough remaining elements to make statistically meaningful estimates (e.g., estimates of the probability density $d(x)$).

When we select only a portion of elements at a location x, the density $d(x)$ in the vicinity of this location is multiplied by the ratio λ of the following two proportions: the proportion of this cluster in the original sample, and the proportion of this cluster in the new sample. Since the values of x_j do not change, the gradient z – and hence, its length l – is also multiplied by the same constant λ. In the vicinity of another cluster, the corresponding values of d' and l' are similarly multiplied by a different constant λ'. It is reasonable to require that the relative quality of different possible thresholds does not change under this transition to a sub-sample. Thus, we arrive at the following definition.

Definition 3. *We say that a non-zero preference relation $\succeq$ is* sub-sampling invariant *if for every two pairs (d, l) and (d', l') and for every two positive real numbers $\lambda > 0$ and $\lambda' > 0$, $(d, l) \succeq (d', l')$ implies that*

$$(\lambda \cdot d, \lambda \cdot l) \succeq (\lambda' \cdot d', \lambda' \cdot l').$$

Proposition 2. *For every closed monotonic sub-sampling invariant non-zero preference relation $\succeq$, $(d, l) \succeq (d', l')$ if and only if $l/d \geq l'/d'$.*

Discussion. Thus, as a threshold, we should select a value of the density $d(x)$ corresponding to the point where the ratio $\frac{\|\nabla d\|}{d}$ attains its largest possible value. This result explains the above empirical rule.

Proof of Proposition 2. Since a preference relation is reflexive, we have $(d,l) \succeq (d,l)$ for every d and l. If we apply invariance with respect to sub-sampling with $\lambda = 1$ and $\lambda' = 1/d$, we get $(d,l) \succeq (1,l/d)$. If we apply invariance with respect to sub-sampling with $\lambda = 1/d$ and $\lambda' = 1$, we get $(1,l/d) \succeq (d,l)$. Thus, $(d,l) \equiv (1,l/d)$. Similarly, $(d',l') \equiv (1,l'/d')$. Thus, $(d,l) \succeq (d',l')$ if and only if $(1,l/d) \succeq (1,l'/d')$. For pairs $(1,l)$, due to monotonicity, $(1,l) \succeq (1,l')$ if and only if $l \geq l'$. Thus, indeed, $(d,l) \succeq (d',l')$ if and only if $l/d \geq l'/d'$. The proposition is proven.

A Solution to the 2nd Challenge. In the above algorithm, we implicitly assume that, while there is some inaccuracy in the measurement results corresponding to each observed situation $x^{(j)}$, each measurement indeed represents the situation of the type in which we are interested. In practice, we are not always sure that what we measured in necessarily one of such situations.

A natural way to describe this uncertainty is to assign, to each observed situation j, a probability p_j (most probably, subjective probability) that this situation is indeed of the desired type.

It is desirable to take these probabilities into account during clustering.

Idea. The main algorithm is based on the Parzen formula

$$d(x) = p(x^{(1)}) \cdot \rho(x^{(1)} - x) + \ldots + p(x^{(N)}) \cdot \rho(x^{(N)} - x) = \frac{1}{N} \cdot \sum_{j=1}^{N} \rho(x^{(j)} - x).$$

In deriving formula, we assumed that all observations $x^{(j)}$ are equally probable, i.e., that they have the same probability $p(x^{(j)})$ to be observed. Now that we know the probability p_j that each observation is real, these observations are not equally probable: the probability $p(x^{(j)})$ of the j-th observation is proportional to p_j: $p(x^{(j)}) = k \cdot p_j$ for some constant k.

This constant can be found from the condition that the overall probability is 1, i.e., that $\sum\limits_{j=1}^{N} p(x^{(j)}) = k \cdot \sum\limits_{j=1}^{N} p_j = 1$. Thus, we get $k = \frac{1}{\sum\limits_{j=1}^{N} p_j}$, and instead of the original Parzen formula, we get a new formula:

$$d(x) = p(x^{(1)}) \cdot \rho(x^{(1)} - x) + \ldots + p(x^{(N)}) \cdot \rho(x^{(N)} - x) = k \cdot d_0(x),$$

where $d_0(x) \stackrel{\text{def}}{=} \sum\limits_{j=1}^{N} p_j \cdot \rho(x^{(j)} - x)$.

Comment. Clusters are then determined based on the set of all the situations x that satisfy the inequality $d(x) \geq t$ for some threshold t. Since $d(x) = k \cdot d_0(x)$, this inequality is equivalent to the inequality $d_0(t) \geq t_0$, where $t_0 \stackrel{\text{def}}{=} \dfrac{t}{k}$.

Thus, instead of considering the actual density $d(x)$ and selecting an appropriate threshold t, we could as well consider a simpler function $d_0(x)$ and select an appropriate threshold t_0 for this simpler function. As a result, we arrive at the following modification of the above algorithm:

Resulting Algorithm. Based on the observations $x^{(j)}$ and on the probabilities p_j, we form an auxiliary function $d_0(x) \stackrel{\text{def}}{=} \sum\limits_{j=1}^{N} p_j \cdot \rho(x^{(j)} - x)$. Then, we select an appropriate threshold t_0 and find clusters as connected components of the set $\{x : d(x) \geq t_0\}$.

Comment. For selecting t_0 we can use the same algorithm as before since, as one can easily see, this algorithm does not change if we simply multiply all the values of $d(x)$ by the same constant $(1/k)$.

A Solution to the 3rd Challenge. In the above algorithm, we assign each situation to a definite cluster; crudely speaking, to the cluster which is most probable to contain this situation. Because of the probabilistic character of the assignment procedure, the resulting "most probable" assignment is not necessarily always the correct one – it is just the assignment which is correct in more cases than other possible assignments.

In reality, it is quite possible that each cluster also contains situations which were not assigned to it – and vice versa, that some situations that were assigned to this cluster actually belong to a different cluster. It is therefore desirable to estimate, for each current cluster c and for each situation x which is currently outside this cluster, the degree to which it is possible that x actually belongs to c.

An Idea on How to Solve This Challenge. In the above algorithm, the clusters were built based on the choice of a threshold t: each cluster c is a connected component of the set $\{x : d(x) \geq t\}$, where $d(x)$ is a probability density function based on the observed situations $x^{(j)}$.

If a situation x does not belong to the given cluster, this means that x cannot be connected to elements of c by points y for which $d(y) \geq t$. In other words, whatever connection we make between the point x and a point x_c from c (e.g., a curve connecting x and x_c), there will be a point y on this connection at which $d(y) < t$.

If for some situation x, there is a connection at which all these intermediate values $d(y)$ are close to t – e.g., exceed $t - \varepsilon$ for some small $\varepsilon > 0$ – this means that the corresponding situation y "almost" belongs to the cluster: it would belong to the cluster if we changed the threshold a little bit. In this case, if we assign degree of confidence 1 to situations originally assigned to the cluster c, it

makes sense to assign a degree close to 1 to this situation c. For example, we can simply take the ratio $\frac{t-\varepsilon}{t}$ as the desired degree of confidence that the situation x belongs to the cluster c.

On the other hand, if no matter how we connect x with some $x_c \in c$, we have to go through some points with very low probability density – e.g., density 0 – this means that no matter how much we decrease the threshold, this situation x will not end up in the cluster c. To such situations x, we should assign low degree of confidence that this situation x belongs to the given cluster c.

Thus, we arrive at the following natural definition.

Resulting Definition. For each situation x and for each cluster c, we estimate the degree $d_c(x)$ to which x belongs to c as the ratio $d_c(x) = \frac{t_c(x)}{t}$, where t is the threshold used for the original clustering, and $t_c(x)$ is the largest value $s \le t$ for which both the situation x and the original cluster c belong to the same connected component of the set $\{y : d(y) \ge s\}$.

Discussion. For elements x that were originally assigned to the cluster c, the degree $d_c(x)$ as defined above is equal to 1.

For elements x that can be connected to c by situations y for which $d(y) \ge t-\varepsilon$, the above-defined degree $d_c(x)$ is larger than or equal to $\frac{t-\varepsilon}{t}$.

Finally, if we have a situation x for which, no matter how we connect it to c, there will always be situations y on this connection for which $d(y) = 0$, then the above-defined degree $d_c(x)$ is equal to 0.

An Alternative Solution to the 3rd Challenge. In the above approach, all the situations x which have been originally assigned to a cluster c are automatically assigned degree $d_c(x) = 1$. An alternative approach is to assign different degrees $d_c(x)$ to different such situations x.

To assign such degrees, we can use the same idea that we used when we assigned degrees $d_c(x)$ to situations x which are *outside* the original cluster c. Namely, the original assignment of a situation x to different clusters is based on the value $d(x)$: situations with $d(x) \ge t$ were assigned to different clusters, while situations with $d(x) < t$ were not assigned to any clusters. If $d(x) = t$, then a minor change in $d(x)$ can move this situation outside the clusters. On the other hand, if $d(x) \gg t$, this means that even after a reasonable change in the value of $d(x)$, the situation x will still be assigned to a cluster. Thus, the larger the value $d(x)$, the larger our confidence that the situation x will be assigned to the cluster. It is therefore reasonable to take $d(x)$ as a degree of confidence that the situation x belongs to the cluster c.

Of course, this value needs to be normalized so that the largest degree will be 1. Thus, we arrive at the following alternative definition.

Alternative Definition. For each situation x and for each cluster c, we estimate the degree $d_c(x)$ to which x belongs to c as follows:

If $d(x) \geq t$, then, as $d_c(x)$, we take the ratio $\dfrac{d(x)}{d_{\max}}$, where $d_{\max} \stackrel{\text{def}}{=} \sup\limits_{y} d(y)$ is the largest possible value of the density $d(x)$.

If $d(x) < t$, then, as the desired degree, we take the ratio $d_c(x) = \dfrac{t_c(x)}{d_{\max}}$, where $t_c(x)$ is the largest value $s \leq t$ for which both the situation x and the original cluster c belong to the same connected component of the set $\{y : d(y) \geq s\}$.

A Solution to the 4th Challenge. Once the original clusters are established, then, for each cluster c, it is desirable to be able to apply the clustering algorithm only to the situations from this cluster – and come up with sub-clusters of the cluster c.

A possible solution is to use the *fuzzy clusters*, i.e., to produce the degrees of belonging (that we produced as a solution to the 3rd challenge), and then to use these degrees when clustering all the situations from the cluster c – as we did in our solution to the second challenge.

Because of the degree of belonging, the resulting density function is different from what we had based on the original sample. As a result, hopefully, we will not simply produce the original cluster (as in the original algorithm) – we will divide this cluster into reasonable sub-clusters.

Implementations. Most of the above solution have been implemented and applied to real-life problems [2,3]; the resulting clustering is indeed closer to the expert-generated clustering than the clustering performed by the usual fuzzy clustering algorithms.

Acknowledgments. This work was supported in part by NSF grants HRD-0734825 and HRD-1242122, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from NIH, by a grant N62909-12-1-7039 from ONR, and by a grant 111T273 from TUBITAK. This work was partially performed when Gözde Ulutagay was visiting the University of Texas at El Paso.

The authors are greatly thankful to Professor Zadeh for inspiring discussions and to the anonymous referees for valuable suggestions.

References

1. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stain, C.: Introduction to Algorithms. MIT Press, Cambridge (2009)
2. Nasibov, E.N., Ulutagay, G.: A new unsupervised approach for fuzzy clustering. Fuzzy Sets and Systems 158, 2118–2133 (2007)
3. Nasibov, E.N., Ulutagay, G.: Robustness of density-based clustering methods with various neighborhood relations. Fuzzy Sets and Systems 160, 3601–3615 (2009)
4. Pal, N.R., Bezdek, J.C.: On the cluster validity for the fuzzy c-means model. IEEE Transactions on Fuzzy Systems 3(3), 370–379 (1995)
5. Parzen, E.: On the estimation of a probability density function and the mode. Annals of Mathematical Statistics 33, 1065–1076 (1962)

6. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets. MIT Press, Cambridge (1998)
7. Rabinovich, S.: Measurement Errors and Uncertainties: Theory and Practice. Springer, New York (2005)
8. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall/CRC, Boca Raton (2007)
9. Zadeh, L.A.: A Note on Z-numbers. Information Sciences 181, 2923–2932 (2011)

Computing Covariance and Correlation in Optimally Privacy-Protected Statistical Databases: Feasible Algorithms

Joshua Day[1], Ali Jalal-Kamali[2], and Vladik Kreinovich[2]

[1] Department of Computer Science,
University of Wisconsin at Whitewater,
Whitewater, WI 53190, USA
dayja10@uww.edu
[2] Department of Computer Science,
University of Texas at El Paso,
El Paso, TX 79968, USA
ajalalkamali@miners.utep.edu, vladik@utep.edu

Abstract. In many real-life situations, e.g., in medicine, it is necessary to process data while preserving the patients' confidentiality. One of the most efficient methods of preserving privacy is to replace the exact values with intervals that contain these values. For example, instead of an exact age, a privacy-protected database only contains the information that the age is, e.g., between 10 and 20, or between 20 and 30, etc. Based on this data, it is important to compute correlation and covariance between different quantities. For privacy-protected data, different values from the intervals lead, in general, to different estimates for the desired statistical characteristic. Our objective is then to compute the range of possible values of these estimates.

Algorithms for effectively computing such ranges have been developed for situations when intervals come from the original surveys, e.g., when a person fills in whether his or her age is between 10 or 20, between 20 and 30, etc. These intervals, however, do not always lead to an optimal privacy protection; it turns out that more complex, computer-generated "intervalization" can lead to better privacy under the same accuracy, or, alternatively, to more accurate estimates of statistical characteristics under the same privacy constraints. In this paper, we extend the existing efficient algorithms for computing covariance and correlation based on privacy-protected data to this more general case of interval data.

Keywords: privacy protection, statistical database, computing covariance, computing correlation,interval uncertainty.

1 Formulation of the Problem

Need for Processing Data in Statistical Databases. Often, we collect data for the purpose of finding possible dependencies between different quantities. For example, we collect all possible information about the medical patients with the hope

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_35,

of finding out which factors affect different illnesses and which factors affect the success of different cures. The resulting collection of records $r_i = (r_{i1}, \ldots, r_{ip})$, $1 \le i \le n$, is known as a *statistical database* since typically, statistical methods are used for look for possible dependencies; see, e.g., [8]. These statistical methods are usually based on computing statistical characteristics such as mean $E_j = \frac{1}{n} \cdot \sum_{i=1}^{n} r_{ij}$, variance $V_j = \frac{1}{n} \cdot \sum_{i=1}^{n} (r_{ij} - E_j)^2$, standard deviation $\sigma_j = \sqrt{V_j}$, covariance $C_{jk} = \frac{1}{n} \cdot \sum_{i=1}^{n} (r_{ij} - E_j) \cdot (r_{ik} - E_k)$, and correlation $\rho_{jk} = \frac{C_{jk}}{\sigma_j \cdot \sigma_k}$.

Need for Privacy Protection. In many real-life situations, e.g., in medicine, it is necessary to process data while preserving the patients' confidentiality.

A similar need for privacy protection exists for analyzing data from social networks.

How to Protect Privacy in Statistical Databases: The Main Idea of an Interval Approach. One of the most efficient methods of preserving privacy is to replace the *exact* values with *intervals* that contain these values.

For example, instead of an exact age, a privacy-protected database only contains the information that the age is, e.g., between 10 and 20, or between 20 and 30, etc.

Interval Approach: A Threshold-Based Approach. In general, for each of p variables x_i, $1 \le i \le p$, we fix some thresholds $t_{i,1} < t_{i,2} < \ldots < t_{i,n_i}$ (e.g., 0, 10, 20, 30, ..., for age), and replace each original value x_i with the range $[t_{i,k}, t_{i,k+1}]$ that contains this value.

In the above example, the actual age of 19 will be replaced by the range $[10, 20]$.

Need to Process Corresponding Interval Data. Based on this interval data, it is important to compute the values of different statistical characteristics such as correlation and covariance between different quantities.

For privacy-protected data, for each statistical characteristic $C(v_1, \ldots, v_m)$, different values v_i from the given intervals $[\underline{v}_i, \overline{v}_i]$ lead, in general, to different estimates $C(v_1, \ldots, v_m)$. Thus, it is necessary to compute the range of possible values of these estimates:

$$C([\underline{v}_1, \overline{v}_1], \ldots, [\underline{v}_m, \overline{v}_m]) \stackrel{\text{def}}{=}$$

$$\{C(v_1, \ldots, v_m) : v_1 \in [\underline{v}_1, \overline{v}_1], \ldots, v_m \in [\underline{v}_m, \overline{v}_m]\}. \quad (1)$$

What Was Known Before. For most statistical characteristics, the problem of computing the range (1) under *general* interval uncertainty is computationally

intractable (NP-hard); see, e.g., [6][1]. However, for the above-described privacy-related case, feasible algorithms are possible for computing many statistical characteristics, in particular, covariance and correlation; see, e.g., [2–6].

Need to Go Beyond the Threshold-Based "Intervalization". In the above threshold-based "intervalization", we replace each data point $r = (r_1, \ldots, r_p)$ with a box

$$b = [\underline{b}_1, \overline{b}_1] \times \ldots \times [\underline{b}_p, \overline{b}_p] \tag{2}$$

formed by the corresponding threshold intervals $[\underline{b}_i, \overline{b}_i]$. The larger the boxes, the wider the resulting interval (1) – i.e., the less accurate our estimates of the corresponding statistical characteristics. From this viewpoint, the boxes b should be as narrow as possible. On the other hand, if they are too narrow, e.g., if some box contains only one record, then the privacy of this record is not well-protected. To properly protect privacy, we need to make sure for some sufficiently large integer K, each box b contains at least K records (this is called *K-anonymity*; see, e.g., [9]), and that for each variable x_i, there are at least ℓ different values of this variable coming from records within this box (this is called *ℓ-diversity*); see, e.g., [1].

Boxes do not have to come from thresholds. The only reasonable restriction is that they should form a *subdivision* in the sense that no two boxes should have a common interior point. Under the privacy-motivated restrictions of K-anonymity and ℓ-diversity, we must look for a subdivision into boxes which leads to the narrowest possible range $C([\underline{v}_1, \overline{v}_1], \ldots, [\underline{v}_p, \overline{v}_p])$ of the desired characteristic. It turns out (see, e.g., [10, 11]) that to attain this narrowest range, we need to use a general subdivision into boxes which is more complex than the above threshold-based one. Namely, in the above threshold-based subdivision into boxes, if two records $(r_1, r_2, \ldots)$ and $(r'_1, r'_2, \ldots)$ have the same value of r_1 (i.e., if $r'_1 = r_1$), then the corresponding boxes have the same x_1-interval $[\underline{b}_1, \overline{b}_1]$. In other words, the selection of the x_1-interval of the corresponding box depends only on the value r_1 and does not depend on the values of all other quantities $r_2, \ldots$

In contrast, in the optimal subdivision into boxes, the same value of r_1, depending on the values of other quantities $r_2, \ldots$, we may need boxes with different x_1-intervals. For example, if for some $r_2, \ldots$, there are more records around the

[1] For those readers who are not familiar with the notion of NP-hardness, here is a brief (somewhat informal) description; for details, see, e.g., [7]. We are interested in the possibility of solving problems by *feasible* algorithms – which is usually interpreted as *polynomial-time* algorithms, i.e., algorithms whose running time is bounded from above by a polynomial of the input length. The class of all the problems which can be solved in feasible (polynomial) time is usually denoted by P. In practice, we normally have problems for which we can feasibly *check* whether a given candidate for a solution is actually a solution; the class of such problems is denoted by NP. A problem $\mathcal{P}_0$ is called NP-hard if every problem from the class NP can be reduced to this problem. Thus, unless it turns out that P=NP – which most computer scientists believe to be impossible – no feasible (polynomial-time) algorithm can solve all the instances of the NP-hard problem $\mathcal{P}_0$.

point $(r_1, r_2, \ldots)$, then, in the optimal subdivision into boxes, these records are assigned to a narrower box, with narrower x_1-intervals. On the other hand, for the same value r_1 and different values $r'_2, \ldots$, there may be much fewer records around the point $(r_1, r'_2, \ldots)$. In this case, in the optimal subdivision into boxes, these new records records are assigned to a wider box, with a wider x_1-interval.

Resulting Problem and What We Do in This Paper. Since the optimal intervalization goes beyond a simple threshold-based one, it is necessary to extend algorithms for estimating covariance and correlation to such optimal intervalization. Such algorithms are presented in this paper.

2 Analysis of the Problem

First Comment: Computing the Upper Endpoint $\overline{C}_{jk}$ Can Be Reduced to Computing the Lower Endpoint $\underline{C}_{jk}$. One can easily check that if we replace each value r_{ik} with its opposite $r'_{ik} = -r_{ik}$, then the covariance C_{jk} changes sign: $C'_{jk} = -C_{jk}$. As a result, if we replace each original interval $[\underline{r}_{ik}, \overline{r}_{ik}]$ with its opposite $[-\overline{r}_{ik}, -\underline{r}_{ik}]$, then the resulting range is the opposite to the original range: $[\underline{C}'_{jk}, \overline{C}'_{jk}] = [-\overline{C}_{jk}, -\underline{C}_{jk}]$. This means, in particular, that $\underline{C}'_{jk} = -\overline{C}_{jk}$ and therefore, that $\overline{C}_{jk} = -\underline{C}'_{jk}$.

Thus, if we know how to compute lower endpoints, we can compute the lower endpoint $\underline{C}'_{jk}$ for the modified database, and then compute $\overline{C}_{jk}$ as $\overline{C}_{jk} = -\underline{C}'_{jk}$.

Because of this reduction, in the following text, we will only consider the problem of computing the lower endpoint $\underline{C}_{jk}$.

Known Facts from Calculus: Reminder. Each statistical characteristic $C(v_1, \ldots, v_m)$ is a continuous function of its variables. It is known that the range of a continuous function on a connected box $[\underline{v}_1, \overline{v}_1] \times \ldots \times [\underline{v}_m, \overline{v}_m]$ is an interval $[\underline{C}, \overline{C}]$ whose endpoints are the smallest possible value $\underline{C}$ of the function $C(v_1, \ldots, v_m)$ on the box and its largest value $\overline{C}$. It is also known that for each continuous function on a closed box, its minimum and its maximum are attained at some points.

When a function $C(v_1, \ldots, v_m)$ attains its minimum on the box at a point $(v_1^{\min}, \ldots, v_i^{\min}, \ldots, v_m^{\min})$, this means, in particular, that for every i, the one-variable function $f(v_i) \stackrel{\text{def}}{=} C(v_1^{\min}, \ldots, v_{i-1}^{\min}, v_i, v_{i+1}^{\min}, \ldots, v_m^{\min})$ attains its minimum on the interval $[\underline{v}_i, \overline{v}_i]$ at $v_i = v_i^{\min}$.

In general, a function $f(x)$ of one variable attains its minimum on an interval $[\underline{x}, \overline{x}]$ either inside this interval or at one of its endpoints $\underline{x}$ or $\overline{x}$. If the function $f(x)$ attains its minimum at an inside point, then its derivative at this point is known to be equal to 0: $f'(x^{\min}) = 0$. If $f(x)$ attains its minimum at $\underline{x}$, then we should have $f'(\underline{x}) \geq 0$ because otherwise, if we had $f'(\underline{x}) < 0$, then, for a small Δx, we would have $f(\underline{x} + \Delta x) < f(\underline{x})$, which contradicts to our assumption that the value $f(\underline{x})$ is the smallest. Similarly, if the function $f(x)$ attains its minimum at $\overline{x}$, we should have $f'(\overline{x}) \leq 0$.

Let Us Apply These Facts to Minimizing Covariance. For covariance, as one can easily check, $\dfrac{\partial C_{jk}}{\partial r_{ij}} = \dfrac{1}{n} \cdot (r_{ik} - E_k)$ and $\dfrac{\partial C_{jk}}{\partial r_{ik}} = \dfrac{1}{n} \cdot (r_{ij} - E_j)$. Thus, for the values $r_{ij}^{\min}$ and $r_{ik}^{\min}$ at which the minimum of C_{jk} is attained, we have one of the three options:

- either $\underline{r}_{ij} < r_{ij}^{\min} < \overline{r}_{ij}$ and $\dfrac{\partial C_{jk}}{\partial r_{ij}} = 0$, i.e., $r_{ik}^{\min} = E_k$;
- or $r_{ij}^{\min} = \underline{r}_{ij}$ and $r_{ik}^{\min} \geq E_k$;
- or $r_{ij}^{\min} = \overline{r}_{ij}$ and $r_{ik}^{\min} \leq E_k$.

Thus:

- if $r_{ik}^{\min} > E_k$, then the first and third cases are impossible, so we must have $r_{ij}^{\min} = \underline{r}_{ij}$;
- if $r_{ik}^{\min} < E_k$, then the first and second cases are impossible, so we must have $r_{ij}^{\min} = \overline{r}_{ij}$.

Therefore, if $E_k < \underline{r}_{ik}$, then, due to $\underline{r}_{ik} \leq r_{ik}^{\min}$, we get $E_k < r_{ik}^{\min}$ and therefore, $r_{ij}^{\min} = \underline{r}_{ij}$. Similarly, if $\overline{r}_{ik} < E_k$, then $r_{ij}^{\min} = \overline{r}_{ij}$.

Likewise, if $r_{ij}^{\min} > E_j$, then $r_{ik}^{\min} = \underline{r}_{ik}$, and if $r_{ij}^{\min} < E_j$, then $r_{ik}^{\min} = \overline{r}_{ik}$. So, if $E_j < \underline{r}_{ij}$, then $r_{ik}^{\min} = \underline{r}_{ik}$, and if $\overline{r}_{ij} < E_j$, then $r_{ik}^{\min} = \overline{r}_{ik}$.

Thus, if we know the location of E_j in comparison to the interval $[\underline{r}_{ij}, \overline{r}_{ij}]$ and we know the location of E_k in comparison with the interval $[\underline{r}_{ik}, \overline{r}_{ik}]$, then, with one exception, we can uniquely determine the minimizing values $r_{ij}^{\min}$ and $r_{ik}^{\min}$. For example, if $E_k < \underline{r}_{ik}$ and $E_j < \underline{r}_{ij}$, then $r_{ij}^{\min} = \overline{r}_{ij}$ and $r_{ik}^{\min} = \overline{r}_{ik}$. If $E_k < \underline{r}_{ik}$ and $\underline{r}_{ij} \leq E_j \leq \overline{r}_{ij}$, then $r_{ij}^{\min} = \overline{r}_{ij} \geq E_j$, hence $r_{ik}^{\min} = \overline{r}_{ik}$.

The only exception is when $E_j \in [\underline{r}_{ij}, \overline{r}_{ij}]$ and $E_k \in [\underline{r}_{ik}, \overline{r}_{ik}]$. In this case, minimizing over r_{ij}, we have three calculus-motivated options:

- the first option is $r_{ik}^{\min} = E_k$;
- the second option is $r_{ij}^{\min} = \underline{r}_{ij}$ and $r_{ik}^{\min} \geq E_k$;
- the third option is $r_{ij}^{\min} = \overline{r}_{ij}$ and $r_{ik}^{\min} \leq E_k$.

These conditions describe a set of possible pairs $(r_{ij}^{\min}, r_{ik}^{\min})$, a set formed by three line segments.

Similarly, minimizing over r_{ik}, we have three other calculus-motivated options:

- the first option is $r_{ij}^{\min} = E_j$;
- the second option is $r_{ik}^{\min} = \underline{r}_{ik}$ and $r_{ij}^{\min} \geq E_j$;
- the third option is $r_{ik}^{\min} = \overline{r}_{ik}$ and $r_{ij}^{\min} \leq E_j$,

which define a new three-segment set. The actual pair $(r_{ij}^{\min}, r_{ik}^{\min})$ belongs to both these sets and thus, belongs to their intersection. This intersection consists of three points: $(\underline{r}_{ij}, \overline{r}_{ik})$, $(\overline{r}_{ij}, \underline{r}_{ik})$, and (E_j, E_k).

Let us show that the minimum cannot be attained at a point (E_j, E_k). Indeed, let us show that if for some small $\Delta \neq 0$, we replace the value $r_{ij} = E_j$ with a new

value $r'_{ij} = E_j + \Delta$ and the value $r_{ik} = E_k$ with a new value $r'_{ik} = E_k - \Delta$, then the covariance will decrease – which shows that the minimum is not attained when $r_{ij} = E_j$ and $r_{ik} = E_k$. To show this, we will use a known equivalent expression for the covariance $C_{jk} = M - E_j \cdot E_k$, where $M \stackrel{\text{def}}{=} \frac{1}{n} \cdot \sum_{i=1}^{n} r_{ij} \cdot r_{ik}$. When we replace the values r_{ij} and r_{ik} with the new values r'_{ij} and r'_{ik}, then the mean E_j is replaced with $E'_j = E_j + \frac{\Delta}{n}$, the mean E_k is replaced with $E'_k = E_k - \frac{\Delta}{n}$. The product $r_{ij} \cdot r_{ik} = E_j \cdot E_k$ is replaced with

$$(E_j + \Delta)(E_k - \Delta) = E_j \cdot E_k - \Delta \cdot E_j + \Delta \cdot E_k - \Delta^2.$$

Thus, the quantity M is replaced with $M' = M - \frac{1}{n} \cdot \Delta \cdot E_j + \frac{1}{n} \cdot \Delta \cdot E_k - \frac{1}{n} \cdot \Delta^2$. Hence, the new expression for the covariance takes the form

$$C'_{jk} = M' - E'_j \cdot E'_k = M - \frac{1}{n} \cdot \Delta \cdot E_j + \frac{1}{n} \cdot \Delta \cdot E_k - \frac{1}{n} \cdot \Delta^2 - \left(E_j + \frac{\Delta}{n}\right) \cdot \left(E_k + \frac{\Delta}{n}\right).$$

After opening parentheses, we can see that the terms proportional to $\Delta \cdot E_j$ and $\Delta \cdot E_k$ cancel out, so we get $C'_{jk} = C_{jk} - \frac{1}{n} \cdot \Delta^2 + \frac{1}{n^2} \cdot \Delta^2 = C_{jk} - \frac{n-1}{n^2} \cdot \Delta^2 < C_{jk}$. This proves that when the box b contains the point (E_j, E_k), then we have only two options for the minimizing values of r_{ij} and r_{ik}.

Towards an Algorithm. In the privacy-protected database, boxes form a subdivision, so for each possible location of the pair (E_j, E_k), there is at most one box that contains this pair. This box contains several records; let us denote their number by n_b. In the minimizing selection, some of the pairs $(r_{ij}^{\min}, r_{ik}^{\min})$ are equal to $(\underline{r}_{ij}, \overline{r}_{ik})$ and some are equal to $(\overline{r}_{ij}, \underline{r}_{ik})$. Covariance does not change if we re-order the records; thus, when computing covariance, we only care about how many of n_b records are equal to $(\underline{r}_{ij}, \overline{r}_{ik})$; let us denote this number by m_b. One can easily check that M, E_j, and E_k are linear functions of m_b; thus, the covariance $C_{jk} = M - E_j \cdot E_k$ is a quadratic function of m_b: $C_{jk} = C_2 \cdot m_b^2 + C_1 \cdot m_b + C_0$, for known values C_i.

To find the smallest possible value of C_{jk}, we want to find a value $m_b = 0, 1, \ldots, n_b$ for which this expression is the smallest possible. This can be done by using the known properties of a quadratic function $C_2 \cdot m_b^2 + C_1 \cdot m_b + C_0$:

- when $C_2 > 0$, it decreases when $m_b \le -\frac{C_1}{2C_2}$ and increases after that;
- when $C_2 < 0$, it increases when $m_b \le -\frac{C_1}{2C_2}$ and decreases after that;
- when $C_2 = 0$, it increases if $C_1 > 0$ and decreases if $C_1 < 0$.

On the interval where this expression is increasing, we take the smallest possible value of m_b; on the interval where this expression is decreasing, we take the largest possible value of m_b.

Towards an Algorithm: Final Touch. What is important is where the values E_j and E_k are in comparison with the endpoints of the corresponding intervals $[\underline{r}_{ij}, \overline{r}_{ij}]$ and $[\underline{r}_{ik}, \overline{r}_{ik}]$. Thus, to find possible ranges of E_j, we can sort all the endpoints $\underline{r}_{ij}$ and $\overline{r}_{ij}$ of the x_j-intervals of different boxes into an increasing sequence $T_{j,1} < T_{j,2} < \ldots$, and consider all possible "small boxes" $b = [T_{j,i_j}, T_{j,i_j+1}] \times [T_{k,i_k}, T_{j,i_k+1}]$. Thus, we arrive at the following algorithm for computing the lower endpoint $\underline{C}_{jk}$ of the range of covariance.

3 Algorithm for Computing Covariance

What Is Given. We are given a finite collection of B boxes $b_a = [\underline{b}_{a1}, \overline{b}_{a1}] \times \ldots \times [\underline{b}_{ap}, \overline{b}_{ap}]$, $1 \le a \le B$. These boxes form a subdivision, i.e., no two boxes have a common interior point. For each of these boxes, we are given the number n_a of records corresponding to this box. We are also given the indices j and k for which we want to find the range of covariance values.

Algorithm. First, we sort all $2B$ j-endpoints $\underline{b}_{aj}$ and $\overline{b}_{aj}$ of all B boxes into an increasing sequence $T_{j,1} < T_{j,2} < \ldots$, and form $\le 2B$ "small" j-intervals $[T_{j,i_j}, T_{j,i_j+1}]$.

Then, we similarly sort all $2B$ k-endpoints $\underline{b}_{ak}$ and $\overline{b}_{ak}$ of all B boxes into an increasing sequence $T_{k,1} < T_{k,2} < \ldots$, and form $\le 2B$ "small" k-intervals $[T_{k,i_k}, T_{k,i_k+1}]$. After that, we form "small boxes" by considering all possible pairs $b = [T_{j,i_j}, T_{j,i_j+1}] \times [T_{k,i_k}, T_{j,i_k+1}]$ of a small j-interval and a small k-interval. In our algorithms, we will analyze these small boxes one by one.

Let us now consider computations corresponding to a fixed small box b. As we have shown, once the small box $b = [\underline{b}_j, \overline{b}_j] \times [\underline{b}_k, \overline{b}_k]$ is fixed, then for almost all original boxes (except for the original box b_{a_0} that contains b), we can uniquely determine the minimizing values $r_{ij}^{\min}$ and $r_{ik}^{\min}$:

- if $\overline{b}_j \le \underline{b}_{aj}$ and $\overline{b}_k \le \underline{b}_{ak}$, then $r_{ij}^{\min} = \underline{b}_{aj}$ and $r_{ik}^{\min} = \underline{b}_{ak}$;
- if $\overline{b}_j \le \underline{b}_{aj}$ and $\underline{b}_{ak} \le \underline{b}_k \le \overline{b}_k \le \overline{b}_{ak}$, then $r_{ij}^{\min} = \overline{b}_{aj}$ and $r_{ik}^{\min} = \underline{b}_{ak}$;
- if $\overline{b}_j \le \underline{b}_{aj}$ and $\overline{b}_{ak} \le \underline{b}_k$, then $r_{ij}^{\min} = \overline{b}_{aj}$ and $r_{ik}^{\min} = \underline{b}_{ak}$;
- if $\overline{b}_{aj} \le \underline{b}_j$ and $\overline{b}_k \le \underline{b}_{ak}$, then $r_{ij}^{\min} = \underline{b}_{aj}$ and $r_{ik}^{\min} = \overline{b}_{ak}$;
- if $\overline{b}_{aj} \le \underline{b}_j$ and $\underline{b}_{aj} \le \underline{b}_k \le \overline{b}_k \le \overline{b}_{ak}$, then $r_{ij}^{\min} = \underline{b}_{aj}$ and $r_{ik}^{\min} = \overline{b}_{ak}$;
- if $\overline{b}_{aj} \le \underline{b}_j$ and $\overline{b}_{ak} \le \underline{b}_k$, then $r_{ij}^{\min} = \overline{b}_{aj}$ and $r_{ik}^{\min} = \overline{b}_{ak}$;
- if $\underline{b}_{aj} \le \underline{b}_j \le \overline{b}_j \le \overline{b}_{aj}$ and $\overline{b}_k \le \underline{b}_{ak}$, then $r_{ij}^{\min} = \underline{b}_{aj}$ and $r_{ik}^{\min} = \overline{b}_{ak}$;
- if $\underline{b}_{aj} \le \underline{b}_j \le \overline{b}_j \le \overline{b}_{aj}$ and $\overline{b}_{ak} \le \underline{b}_k$, then $r_{ij}^{\min} = \overline{b}_{aj}$ and $r_{ik}^{\min} = \underline{b}_{ak}$.

This way, for each of the boxes b_a ($a \ne a_0$), we can compute this box's contributions to the expressions M, E_j, and E_k as, correspondingly,

$$\frac{n_a}{n} \cdot r_{ij}^{\min} \cdot r_{ik}^{\min}, \quad \frac{n_a}{n} \cdot r_{ij}^{\min}, \text{ and } \frac{n_a}{n} \cdot r_{ik}^{\min}.$$

For the box $b_{a_0} = [\underline{b}_{a_0 j}, \overline{b}_{a_0 j}] \times [\underline{b}_{a_0 k}, \overline{b}_{a_0 k}]$, the corresponding contributions take the form

$$\frac{m_{a_0}}{n} \cdot \underline{b}_{a_0 j} \cdot \overline{b}_{a_0 k} + \frac{n_{a_0} - m_{a_0}}{n} \cdot \overline{b}_{a_0 j} \cdot \underline{b}_{a_0 k},$$

$$\frac{m_{a_0}}{n} \cdot \underline{b}_{a_0 j} + \frac{n_{a_0} - m_{a_0}}{n} \cdot \overline{b}_{a_0 j}, \text{ and } \frac{m_{a_0}}{n} \cdot \overline{b}_{a_0 k} + \frac{n_{a_0} - m_{a_0}}{n} \cdot \underline{b}_{a_0 k},$$

with an unknown m_{a_0}. By adding the contributions corresponding to different boxes and forming $C_{jk} = M - E_j \cdot E_k$, we get an expression for C_{jk} which is quadratic in m_{a_0}. By using techniques described in the previous section, we can compute the minimum of this expression over all possible integer values m_{a_0} from 0 to n_{a_0}. This minimum $C_{jk}(b)$ is the smallest possible value of the covariance under the assumption that the pair (E_j, E_k) belongs to the small box b.

To find the desired value $\underline{C}_{jk}$, we can then compute the smallest of the values $C_{jk}(b)$ corresponding to all possible small boxes b.

Computational Time for This Algorithm. Sorting takes time $O(B \cdot \log(B))$. After sorting, we get $\leq 2B$ j-intervals and $\leq 2B$ k-intervals, so we get $O(B^2)$ small boxes – pairs of such intervals.

In the main part of the algorithm, for each of $O(B^2)$ small boxes b and for each of B original boxes b_a, we need finitely many computational steps. Thus, the total number of computational steps for the main part is bounded by $O(B^2) \cdot B \cdot \text{const} = O(B^3)$. The total computation time is thus equal to $O(B \cdot \log(B)) + O(B^3)$, i.e., to $O(B^3)$. This algorithm requires cubic time and is, therefore, feasible.

Comment. According to [10], in some cases, better estimates for covariance come from weighted estimates $C^w_{jk} = \sum\limits_{i=1}^{n} w_i \cdot (r_{ij} - E^w_j) \cdot (r_{ik} - E^w_k)$, where

$$E^w_j = \sum_{i=1}^{n} w_i \cdot r_{ij}, \quad E^w_k = \sum_{i=1}^{n} w_i \cdot r_{ik},$$

and w_i are appropriate weights for which $w_i \geq 0$ and $\sum\limits_{i=1}^{n} w_i = 1$. The weight w_i of a record depends only on the box b_a that contains this record. In other words, for some values W_a, $w_i = W_a$ for all the records r_i from the box b_a. In these terms, the equality $\sum\limits_{i=1}^{n} w_i = 1$ means that $\sum\limits_{a} n_a \cdot W_a = 1$. The formula for C^w_{jk} can be represented in an equivalent form, as $C^w_{jk} = M^w - E^w_j \cdot E^w_k$, where $M^w_{jk} = \sum\limits_{i=1}^{n} w_i \cdot r_{ij} \cdot r_{ik}$.

An analysis similar to the one from Section 2 shows that, in effect, the algorithm from Section 3 can be applied for computing the range of this characteristic as well; the only difference is that after selecting the values $r^{\min}_{ij}$ and $r^{\min}_{ik}$, we need to use the weighted expressions M^w, E^w_j, and E^w_k instead of original equal-weight expressions for M, E_j, and E_k.

4 Algorithms for Computing Correlation

Correlation: Reminder. The Pearson's correlation coefficient ρ describes the degree of dependence between the inputs: if the coefficient ρ is close to 1 or to -1, this means that there is a strong dependence; if this coefficient is close to 0, this means that most probably, there is no dependence.

Correlation under Interval Uncertainty: Practical Meaning of Lower and Upper Bounds. Under interval uncertainty, instead of a single value ρ, we get an interval $\left[\underline{\rho}, \overline{\rho}\right]$ of possible values. For positive values ρ, the upper endpoint $\overline{\rho}$ describes to what extent it is *possible* that there is a dependence between the inputs, while the lower endpoint $\underline{\rho}$ describes to what extent, based on the available data, we can *guarantee* that there is a dependence. Similarly, for negative values ρ, the lower endpoint $\underline{\rho}$ describes to what extent it is *possible* that there is a dependence between the inputs, while the upper endpoint $\overline{\rho}$ describes to what extent, based on the available data, we can *guarantee* that there is a dependence.

Which Endpoints Are Most Important for Statistical Databases. As we have mentioned, one of the main purposes of statistical databases is to discover possible new dependencies – dependencies which can then be checked and utilized. From this viewpoint, the most important endpoints are: the upper endpoint for the positive correlation, and the lower endpoint for the negative correlation.

Computing Correlation: What Is Known. The relative importance of different bounds is good news: while in general, computing correlation under interval uncertainty is NP-hard (see, e.g., [6]), a feasible (i.e., polynomial-time) algorithm is possible for computing the upper endpoint $\overline{\rho}$ for positive correlations and the lower endpoint $\underline{\rho}$ for negative correlations; see, e.g., [2].

The Known Algorithm Is Rather Slow. This algorithm is polynomial-time: for inputs consisting of n records, its computation time is bounded by $O(n^5)$.

However, from the practical viewpoint, even for a small database with $n = 1000$ records, this means 10^{15} arithmetic operations: two weeks on a Gigaflop machine; for $n = 10^4$ records, this already means an unrealistic amount of 10^{20} operations.

For Statistical Databases with Privacy-Motivated Boxes, the Known Algorithm Can Be Made Somewhat Faster. In the algorithm from [2], we consider possible quadruples (pairs of pairs) of vertices. In the privacy-motivated case, we have $\le 4B$ vertices, where B is the number of different boxes. Thus, the total number of quadruples of vertices is $O(B^4)$.

According to [2], once the quadruple is fixed, then, within each box b_a, we select the same optimizing values $r_{ij}^{\max}$ and $r_{ik}^{\max}$ (or $r_{ij}^{\min}$ and $r_{ik}^{\min}$) for all the records from this box. Thus, once the quadruple is fixed, we need to perform only finitely many computations within each box – and then, as we did for covariance, multiply the results by n_a. For each of $O(B^4)$ quadruples, we therefore need $O(B)$ computational steps, to the total of $O(B^4) \cdot O(B) = O(B^5)$.

This number of steps is still large, but since the number of boxes is much smaller than the number of records, this number of steps is much smaller than $O(n^5)$ – and thus, more realistic.

Acknowledgments. This work was supported by The National Center for Border Security and Immigration (NCBSI) at the University of Texas at El Paso, a Department of Homeland Security (DHS) Center of Excellence, by Grants 1 T36 GM078000-01 and 1R43TR000173-01 from the National Institutes of Health, by the National Science Foundation grants HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence) and DUE-0926721, and by a grant N62909-12-1-7039 from the Office of Naval Research.

The authors are thankful to the anonymous referees for valuable suggestions.

References

1. Ghinita, G., Karras, P., Kalnis, P., Mamoulis, N.: A Framework for Efficient Data Anonymization under Privacy and Accuracy Constraints. ACM Transactions on Database Systems 34(2), Article 9 (2009)
2. Jalal-Kamali, A., Kreinovich, V.: Estimating Correlation under Interval Uncertainty. Mechanical Systems and Signal Processing 37, 43–53 (2013)
3. Jalal-Kamali, A., Kreinovich, V., Longpré, L.: Estimating Covariance for Privacy Case under Interval (and Fuzzy) Uncertainty. In: Yager, R.R., Reformat, M., Shahbazova, S., Ovchinnikov, S. (eds.) Proceedings of the World Conference on Soft Computing, San Francisco, CA, May 23-26 (2011)
4. Kreinovich, V., Longpré, L., Starks, S.A., Xiang, G., Beck, J., Kandathi, R., Nayak, A., Ferson, S., Hajagos, J.: Interval Versions of Statistical Techniques, with Applications to Environmental Analysis, Bioinformatics, and Privacy in Statistical Databases. Journal of Computational and Applied Mathematics 199(2), 418–423 (2007)
5. Kreinovich, V., Xiang, G., Starks, S.A., Longpré, L., Ceberio, M., Araiza, R., Beck, J., Kandathi, R., Nayak, A., Torres, R., Hajagos, J.: Towards combining probabilistic and interval uncertainty in engineering calculations: algorithms for computing statistics under interval uncertainty, and their computational complexity. Reliable Computing 12(6), 471–501 (2006)
6. Nguyen, H.T., Kreinovich, V., Wu, B., Xiang, G.: Computing Statistics under Interval and Fuzzy Uncertainty. SCI, vol. 393. Springer, Heidelberg (2012)
7. Papadimitriou, C.H.: Computational Complexity. Addison Wesley, San Diego (1994)
8. Sheskin, D.J.: Handbook of Parametric and Nonparametric Statistical Procedures. Chapman & Hall/CRC, Boca Raton (2011)
9. Sweeney, L.: k-anonymity: a model for protecting privacy. International Journal on Uncertainty, Fuzziness and Knowledge-Based System 10(5), 557–570 (2002)
10. Xiang, G., Ferson, S., Ginzburg, L., Longpré, L., Mayorga, E., Kosheleva, O.: Data Anonymization that Leads to the Most Accurate Estimates of Statistical Characteristics: Fuzzy-Motivated Approach. In: Proceedings of the Joint World Congress of the International Fuzzy Systems Association and Annual Conference of the North American Fuzzy Information Processing Society IFSA/NAFIPS 2013, Edmonton, Canada, June 24-28, pp. 611–616 (2013)
11. Xiang, G., Kreinovich, V.: Data Anonymization that Leads to the Most Accurate Estimates of Statistical Characteristics. In: Proceedings of the IEEE Symposium on Computational Intelligence for Engineering Solutions CIES 2013, Singapore, April 16-19, pp. 163–170 (2013)

Feature Selection with Fuzzy Entropy to Find Similar Cases

József Mezei[1], Juan Antonio Morente-Molinera[2], and Christer Carlsson[1]

[1]IAMSR
Åbo Akademi University
Turku, Finland
{jmezei,Christer.Carlsson}@abo.fi
[2]Department of Computer Science and Artificial Intelligence
University of Granada
Granada, Spain
jamorena@decsai.ugr.es

Abstract. Process interruptions are carried out either automatically by monitoring and control systems that react to deviations from standards or by operators reacting to anomalies or incidents. Process interruptions in (very) large production systems are difficult to trace and to deal with; an extended stop is also very costly and solutions are sought to find an effective support technology to minimize the number of involuntary process interruptions. Feature selection is intended to reduce the complexity of handling the interactions of numerous factors in large process systems and to help find the best ways to handle process interruptions. We show that feature selection can be carried out with fuzzy entropy and interval-valued fuzzy sets.

1 Introduction

In the modern process industry we typically have to deal with very large and very complex systems that should be operated at close to maximum capacity most of the time with as few interruptions as possible. In this way the return on assets (ROA) can be kept at levels that will give sustainable competitive advantages for the company running the process(es).

Interruptions are made by monitoring and control systems that react to deviations from predefined standard settings or by operators reacting to anomalies or to incidents that deviate from process operations that are considered normal. The incidents that cause interruptions by monitoring and control systems are easily identified (at least in an initial round) as they are deviations from preset standards but the incidents to which operators react can be more challenging (also if the initial diagnostics of the monitoring and control systems turn out to be false). The more demanding incidents require an intelligent or smart diagnostics process to find out *what* is taking place, *when* it got started, *how* the incident has happened and *how* it will work out to what consequences. This then typically initiates reasoning and decision processes to work out (i) *what* should

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_36,

be done, (ii) *when* and (iii) *how* it should be done. Quite frequently we want this to be done in some optimal ways.

In the process industry this needs to happen in almost real time as we do not want many interruptions, and if they happen they should be as short as possible. We work on building ontology of the imprecise as a platform for handling incidents for the process industry when diagnostics should be fast and sufficiently correct; this is part of a research program called D2I (Data to Intelligence) funded by the process industry and Tekes (The Finnish Funding Agency for Technology and Innovation).

The cases we are considering are characterized as "big data", i.e. there is a very large set of documents that cover the incidents and what decisions were made and what the outcomes were (actually - how good the decisions were).

In our work with the process industry we found that it makes sense to first work out the core logic of the activities and events that create the incidents; this will reduce the complexity of the problem and in many cases also the dimensionality of the data sets. Second, carry out a feature selection to find the variables that will contribute to the problem solving; then use, for instance, principal component analysis to find the variables that contribute to the data volume but not to the problem solving (if some variables show up in both phases the analysis needs some further steps).

The purpose of feature selection is to reduce the number of input variables (features) describing the logic of the incidents. By selecting an appropriate set of features and excluding the ones that are irrelevant the problem solving process can be improved not only in terms of computational complexity but also in terms of the interpretability of the methods.

The feature selection set is used to identify similar incident cases that can be used in case-based reasoning implementations to significantly speed up the problem-solving processes. In this paper we will focus on the feature selection and work out how to do it with fuzzy entropy and interval-valued fuzzy sets.

The paper is structured as follows: in section 2 preliminary definitions of entropy and interval-valued fuzzy sets are given; section 3 shows the feature selection method which is illustrated with an example in section 4; conclusions are given and future research is outlined in section 5.

2 Preliminaries

In order to present the feature selection method, we will present the necessary definitions concerning fuzzy entropy, type-2 fuzzy sets and feature selection.

2.1 Fuzzy Entropy

In [12], Zadeh defined for the first time a fuzzy entropy on a fuzzy set A for a finite set $X = \{x_1, x_2, ..., x_n\}$ with respect to a probability distribution $P = \{p_1, p_2, ..., p_n\}$ as

$$H(X) = -\sum_{i=1}^{n} \mu_A(x_i) p(x_i) \log p(x_i),$$

where μ_A is the membership function of the fuzzy set A. A fuzzy set A of X is characterized by its membership function $\mu_A : X \to [0,1]$ where $\mu_A(x)$ is interpreted as the degree of membership of element x in fuzzy set A for each $x \in X$. If the membership value in the fuzzy entropy takes the form

$$\mu_A(x) = \begin{cases} 1, \text{ if } x \in X \\ 0, \text{otherwise} \end{cases}$$

then we obtain the definition of entropy introduced by Shannon [9].

Since the first proposal, many different approaches were proposed to quantify the "fuzziness" of a fuzzy set using entropy measures. Although there are many different definitions, the common basis for a well-defined entropy measure (since the proposal of Kosko [5]) is the four De Luca-Termini axioms [3]. They include the following:

1. $H(A) = 0$ if and only if $A \in X$ is a crisp set.
2. $H(A) = 1$ if and only if $\mu_A(x_i) = 0.5$ for every i.
3. $H(A) \leq H(B)$ if A is less fuzzy than B, i.e., if $\mu_A(x) \leq \mu_B(x)$ when $\mu_B(x) \leq 0.5$ and $\mu_A(x) \geq \mu_B(x)$ when $\mu_B(x) \geq 0.5$.
4. $H(A) = H(A^C)$ where $A^C = 1 - A$ denotes the complement of A.

Different definitions of fuzzy entropy include:

– De Luca and Termini [3]

$$H_{LT}(A) = -\sum_{i=1}^{n} [\mu_A(x_i) \log(\mu_A(x_i)) + (1 - \mu_A(x_i)) \log(1 - \mu_A(x_i))]$$

– Kosko [5]

$$H_K(A) = \frac{\sum_{i=1}^{n} \min(\mu_A(x_i), \mu_{A^C}(x_i))}{\sum_{i=1}^{n} \max(\mu_A(x_i), \mu_{A^C}(x_i))}$$

Fuzzy entropy has been applied successfully in different decision making problems [13] and image processing [1]. One of the most important applications of fuzzy entropy is feature selection as a basis for classification and segmentation. Lee et al. [6] employ fuzzy entropy measure to partition the feature space into decision regions and to select relevant features with good separability for the classification task. Pasi [8] combines similarity-based classification procedure with feature selection process based on fuzzy entropy and obtains satisfactory results with a smaller number of selected features than previous studies. In this paper we will follow the approach proposed by Shie and Chen [10], and extend the proposal by incorporating interval-valued fuzzy sets and fuzzy entropy of interval-valued fuzzy sets in the feature selection process. In the following we recall the main concepts from [10].

Let X be a universal set with n elements distributed in a pattern space: $X = \{x_1, x_2, ..., x_n\}$. Let A be a fuzzy set defined on an interval of pattern space which contains k elements. The mapped membership degree of the element

x_i with the fuzzy set A is $\mu_A(x_i)$. The n elements are divided into m classes: $C_1, C_2, ..., C_m$. The match degree D_j with the fuzzy set A for the elements of the class C_j on an interval is defined as

$$D_j = \frac{\sum_{x_i \in C_j} \mu_A(x_i)}{\sum_{x_i \in X} \mu_A(x_i)}.$$

The fuzzy entropy $H_j(A)$ of the elements of class C_j in an interval is defined as

$$H_j(A) = -D_j \log D_j.$$

The fuzzy entropy $H(A)$ on the universal set X for the elements within an interval is defined as

$$H(A) = \sum_{j=1}^{m} H_j(A).$$

2.2 Type-2 Fuzzy Sets

A *fuzzy number* A is a fuzzy set in $\mathbb{R}$ with a normal, fuzzy convex and continuous membership function of bounded support. The family of fuzzy numbers is denoted by $\mathcal{F}$. Fuzzy numbers can be considered as possibility distributions.

Definition 1 ([4]). *An interval-valued fuzzy set A defined on X is given by*

$$A = \left\{ (x, [\mu_A^L(x), \mu_A^U(x)]) \right\}, x \in X,$$

where $\mu_A^L(x), \mu_A^U(x) : X \to [0,1]; \forall x \in X, \mu_A^L(x) \leq \mu_A^U(x)$, and the ordinary fuzzy sets $\mu_A^L(x)$ and $\mu_A^U(x)$ are called lower fuzzy set and upper fuzzy set about A, respectively.

The most used types of fuzzy sets in different applications are triangular and trapezoidal fuzzy numbers. The membership function of a triangular fuzzy number A can be written as,

$$A(x) = \begin{cases} 1 - \dfrac{a - x}{\alpha} & \text{if } a - \alpha \leq x \leq a \\ 1 - \dfrac{x - a}{\beta} & \text{if } a \leq x \leq a + \beta \\ 0 & \text{otherwise} \end{cases}$$

and we use the notation $A = (a, \alpha, \beta)$. An interval-valued fuzzy set $A = (A^U, A^L)$ of triangular form can be represented by six parameters $A = (a, \alpha_1, \beta_1, \alpha_2, \beta_2)$ where $A^L = (a, \alpha_1, \beta_1)$ stands for its lower fuzzy number and $A^U = (a, \alpha_2, \beta_2)$ denotes its upper fuzzy number.

It is important to mention, that there exist different definitions of entropy for Type-2 fuzzy sets, in this paper we will consider the one introduced by Szmidt and Kacprzyk [11]:

$$H_{SK}(A) = \frac{1}{n} \sum_{i=1}^{n} \frac{1 - \max(1 - \mu_A^U(x_i), \mu_A^L(x_i))}{1 - \min(1 - \mu_A^U(x_i), \mu_A^L(x_i))}.$$

3 Feature Selection with Interval-Valued Fuzzy Sets

The first step in employing entropy for feature subset selection is to define the optimal number of intervals (classes) and the corresponding membership functions for every dimension. The fuzzy entropy of a feature decreases when the number of intervals increases. Too many clusters can result in over-fitting problem and reduce the classification accuracy rates. For every feature in the model, on way to determine the optimal number of fuzzy sets to describe the variable is to perform cluster analysis on the feature values, determine the cluster centers and create the membership functions. After calculating the fuzzy entropy of each fuzzy set A_i of the feature F_j the fuzzy entropy of the feature itself, if the entropy of a new clustering is not improved above the predefined threshold value, we can choose the lastly created clustering as the final one.

If we have l number of features and 1 outcome variable (with m classes), the universal set is a subset of $\mathbb{R}^{l+1}$. For the final number of fuzzy subsets for F_j we will use the notation k_j. The upper fuzzy membership values for F_j can be defined in the following way:

– For the left-most cluster center, c_1:

$$A_1^U(x) = \begin{cases} 0.5 + \dfrac{0.5(x - x_{min})}{c_1 - x_{min}}, & \text{if } x_{min} \le x \le c_1 \\ \dfrac{c_2 - x}{c_2 - c_1}, & \text{if } c_1 \le x \le c_2 \\ 0, & \text{otherwise} \end{cases}$$

– For the right-most cluster center, c_{k_j}:

$$A_{k_j}^U(x) = \begin{cases} \dfrac{x - c_{k-1}}{c_k - c_{k-1}}, & \text{if } c_{k_j - 1} \le x \le c_{k_j} \\ 0.5 + \dfrac{0.5(x_{max} - x)}{x_{max} - c_k}, & \text{if } c_{k_j} \le x \le x_{max} \\ 0, & \text{otherwise} \end{cases}$$

– For an internal cluster center, c_i: $A^U = (c_i, c_i - c_{i-1}, c_{i+1} - c_i)$

The lower fuzzy numbers can be defined in a similar way with the center the same as the center of the corresponding upper fuzzy number and the support is contained in the support of the upper fuzzy number.

To improve the model presented in [10], we propose two different solutions to incorporate interval-valued fuzzy sets. A straightforward extension is to define the upper and lower match degrees in the following way:

$$D_j^U = \frac{\sum_{x_i \in C_j} \mu_{A^U}(x_i)}{\sum_{x_i \in X} \mu_{A^U}(x_i)},\ D_j^L = \frac{\sum_{x_i \in C_j} \mu_{A^L}(x_i)}{\sum_{x_i \in X} \mu_{A^L}(x_i)}.$$

The overall match degree can be calculated as the function of the upper and lower match degrees (in the simplest case as linear combination), and the entropy is calculated using this overall match degree, D_j as

$$H_j(A) = -D_j \log D_j.$$

A different modification of the original model is to use a fuzzy entropy defined for interval-valued fuzzy sets. In this paper, we use the one proposed by Szmidt and Kacprzyk [11]. According to this measure, the match degree can be calculated as

$$D_j = \sum_{i=1}^{n} \frac{1 - \max(1 - \mu_A^U(x_i), \mu_A^L(x_i))}{1 - \min(1 - \mu_A^U(x_i), \mu_A^L(x_i))}.$$

The fuzzy entropy then calculated as the sum of the fuzzy entropy of the k_j interval-valued fuzzy sets describing a feature:

$$H(A) = \sum_{j=1}^{m} H_j(A).$$

As a result of choosing the optimal number of fuzzy subsets for every feature, we will also obtain the overall entropy of the features with respect to the outcome variable. The feature with the optimal entropy value will be selected as the best describing variable of the model and used in the consequent steps. In the algorithm, in every step one new feature is added to the set of selected features: the choice is determined by calculating the joint entropy of a subset of features with respect to the outcome variable. In the following we describe how the joint entropy is calculated when choosing the second feature.

Let us suppose that F_1 was chosen as the best feature. For every $h = 2, ..., l$, we need to calculate the entropy of the pair $(A_s^1, A_t^h), s = 1, ..., k_1, t = 1, ..., k_h$ in the following way. First we need to calculate the match degree: both proposed methods can be applied. According to the first one, we calculate the upper and lower match degrees as

$$D_j^{U_{1,h}}(s,t) = \frac{\sum_{x_i^{l+1} \in C_j} \max(\mu_{A_s^{U_1}}(x_i^1), \mu_{A_t^{U_h}}(x_i^h))}{\sum_{x_i \in X} \max(\mu_{A_s^{U_1}}(x_i^1), \mu_{A_t^{U_h}}(x_i^h))},$$

$$D_j^{L_{1,h}}(s,t) = \frac{\sum_{x_i^{l+1} \in C_j} \max(\mu_{A_s^{L_1}}(x_i^1), \mu_{A_t^{L_h}}(x_i^h))}{\sum_{x_i \in X} \max(\mu_{A_s^{L_1}}(x_i^1), \mu_{A_t^{L_h}}(x_i^h))}.$$

The overall match degree can be calculated as the linear combination of the upper and lower match degrees. According to the second approach, we calculate the match degree as

$$D_j^{1,h}(s,t) = \sum_{i=1}^{n} \frac{\min\left\{1 - \max(1 - \mu_{A_s^{U_1}}(x_i), \mu_{A_s^{L_1}}(x_i)), 1 - \max(1 - \mu_{A_t^{U_h}}(x_i), \mu_{A_t^{L_h}}(x_i))\right\}}{\max\left\{1 - \max(1 - \mu_{A_s^{U_1}}(x_i), \mu_{A_s^{L_1}}(x_i)), 1 - \max(1 - \mu_{A_t^{U_h}}(x_i), \mu_{A_t^{L_h}}(x_i))\right\}}.$$

The entropy of the pair (A_s^1, A_t^h) is

$$H(A_s^1, A_t^h) = \sum_{j=1}^{m} -D_j^{1,h}(s,t) \log D_j^{1,h}(s,t),$$

and finally the overall entropy

$$H(F_1, F_h) = \sum_{s=1}^{k_1} \sum_{t=1}^{k_h} H(A_s^1, A_t^h).$$

At the end of this step, the feature pair with the optimal entropy will be selected as the best describing pair. If the improvement in the entropy is above a given threshold, we continue the algorithm by fixing the best pair and calculate the joint entropy with all the remaining features one by one.

4 Illustrative Example

To illustrate the feature selection method with interval-valued fuzzy entropy, we will use the data set from [2]. The data set contains the description of 4898 wines in terms of 11 different properties (acidity, pH level, alcohol etc). The outcome variable is the quality of wine measured on a discrete scale between 1 and 10. We used MATLAB software [7] for the implementation of the feature selection algorithm. In this example we focus on reducing the number of variables in a model, not on the classification of the wines.

Table 1 compares the features that were selected as the best variables based on 3 different approaches: by using fuzzy entropy and by using the two different approaches described in this paper based on the entropy of interval-valued fuzzy sets. As we can see from the table, Alcohol is always chosen as the best feature as it minimizes the entropy in all 3 methods. The original fuzzy entropy method and the extension with type-2 fuzzy sets result in the same subset of features, the difference is that by using a threshold value, according to the original fuzzy entropy all three features would be selected while based on interval-valued fuzzy sets, Alcohol and Chlorides provide sufficient information, there is no significant improvement by selecting also CitAcid. Using the upper and lower match degrees, we obtained different set of features in the second and third steps: Density and VolAcid.

Table 1. Comparison of the results of the different approaches

Step	Fuzzy entropy	Upper and lower match degrees	Type-2 entropy
1	Alcohol	Alcohol	Alcohol
2	Chlorides	Density	Chlorides
3	CitAcid	VolAcid	CitAcid

5 Conclusions and Future Research

In this paper, we presented an algorithm for feature selection employing interval-valued fuzzy sets and fuzzy entropy. The membership functions on the fuzzy sets are defined individually for every variable by creating a partition of the observation with minimal fuzzy entropy. The selection of the features is performed by

increasing the selected subset with the feature that minimizes the joint entropy with respect to an output variable. In future research, the main application of the algorithm will be the identification of similar cases in an event database in case of an incident. Using the presented method, the number of variables to be considered can be significantly reduced while minimizing the information loss, and the identified features can be used to find similar cases and consequently to identify the possible causes of an incident. In future research we will test the classification performance of the algorithm by comparing the performance with type-1 fuzzy entropy measures.

Acknowledgments. This research has been funded through the TEKES strategic research project Data to Intelligence [D2I], project number: 340/12.

References

1. Cheng, H.D., Chen, Y.-H., Sun, Y.: A novel fuzzy entropy approach to image enhancement and thresholding. Signal Processing 75(3), 277–301 (1999)
2. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems 47(4), 547–553 (2009)
3. De Luca, A., Termini, S.: A definition of non-probabilistic entropy in the setting of fuzzy sets theory. Information and Computation 20, 301–312 (1972)
4. Gorzalczany, M.B.: A method of inference in approximate reasoning based on interval-valued fuzzy sets. Fuzzy sets and systems 21(1), 1–17 (1987)
5. Kosko, B.: Fuzzy entropy and conditioning. Information Sciences 40(2), 165–174 (1986)
6. Lee, H.M., Chen, C.M., Chen, J.M., Jou, Y.L.: An efficient fuzzy classifier with feature selection based on fuzzy entropy. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 31(3), 426–432 (2001)
7. MATLAB Release, The MathWorks, Inc., Natick, Massachusetts, United States (2012a)
8. Luukka, P.: Feature selection using fuzzy entropy measures with similarity classifier. Expert Systems with Applications 38(4), 4600–4607 (2011)
9. Shannon, C.E.: A mathematical theory of communication. Bell System Technical Journal 27(3), 379–423 (1948)
10. Shie, J.D., Chen, S.M.: Feature subset selection based on fuzzy entropy measures for handling classification problems. Applied Intelligence 28(1), 69–82 (2008)
11. Szmidt, E., Kacprzyk, J.: Entropy for intuitionistic fuzzy sets. Fuzzy Sets and Systems 118, 467–477 (2001)
12. Zadeh, L.A.: Fuzzy sets. Information and Control 8, 338–353 (1965)
13. Wu, J.-Z., Zhang, Q.: Multicriteria decision making method based on intuitionistic fuzzy weighted entropy. Expert Systems with Applications 38(1), 916–922 (2011)

Computing Intensive Definition of Products

László Horváth and Imre J. Rudas

Óbuda University, John von Neumann Faculty of Informatics,
Institute of Applied Mathematics
H-034 Budapest, Bécsi u. 96/b
horvath.laszlo@nik.uni-obuda.hu, rudas@uni-obuda.hu

Abstract. By their widespread application, engineering virtual spaces together with computation methods play key role in product information management in order to assist product definition and the relevant decisions. Research in order to include hard and soft computing methods in well organized product model of increasing intelligence is a main objective of product model development efforts. This paper is a contribution to these efforts by a possible method for including a higher level knowledge based modeling as extension to the currently applied product models. After an introduction of relevant characteristics of engineering problem solving, role of soft computing is discussed. Following this, extending of knowledge integration in product model by the proposed method is explained and new features for the extended product model are introduced. Finally, the proposed new context structure is outlined and its implementation in product lifecycle management (PLM) systems is conceptualized.

1 Introduction

During the past decades, vast methodology has been developed for the support of engineering activities during lifecycle of products. In order to achieve consistent solutions, integrated product models are applied by extensive utilization of novel approaches such as feature principle, context based definition of features, and representation of corporate knowledge in product model.

By now, feature principle is applied for all engineering objects in a product model for PLM. Feature modifies product model in a well defined way. Its position and activity can be changed in the product model. Some parameters of a feature are defined in the context some of the other features while some of the other features are defined in the context of given parameters of this feature.

Knowledge representation in current product models is not suitable for real connection of human request with model feature generation process. In order to achieve better connection, the current knowledge representation capabilities must be extended. Some engineering activities need hard, while others need soft computing based problem solving for numerical analysis and imprecise approximation, respectively. The two needs may encounter in the same product definition process. Inadequate knowledge representation limits the application of hard and soft computing methods.

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_37,

The Laboratory of Intelligent Engineering System (LIES) which is the laboratory of the Institute of Applied Mathematics, John von Neumann Faculty of Informatics, Óbuda University does fundamental and applied researches in new ideas and methods for representative leading product lifecycle management (PLM). As one of the results, a new integration of knowledge in product model is proposed to improve communication of human request with product feature generation processes in this paper.

Integration of engineering information in product model started with the standard ISO 10303 by International Organization for Standardization (ISO) during eighties and nineties. Current advanced engineering practice is concentrated in Product Lifecycle Management (PLM) systems. This demands new considerations at product modeling. Main research issue in product model is bridging the gap between human and feature object generation processes. Soft computing is anticipated to have key role in this work in the future.

In this paper, considering the need for the above integration, a method is introduced for contextual representation of human request, product behavior, and problem solving knowledge in product model with particular attention to the integration and utilization of the strength of the soft computing. An introduction to relevant characteristics of engineering problem solving is followed by a discussion on role of soft computing. As the main contribution, extending of knowledge integration in product model is explained including new product model features for this purpose. The last sector of the paper proposes a new context structure and concept of implementation in PLM systems in close connection with this structure.

2 Management of Product Information in Increasingly Intelligent PLM Systems

Situation driven definition of product features has become a critical issue in PLM systems. In order to achieve this, self modification capability of product model must be developed for the case of changed circumstances. Active knowledge in product model initiates propagation of these changes in the product model along contextual chains of product features.

A simplified schema of main elements and their connections in a PLM system answers the question that what are purposes and functions of a PLM system (Fig. 1). Communication of engineers with the PLM system is done through engineer community management. Engineers access PLM system using this functionality. Product definition and simulation process management organizes PLM activities and places process definition features in the product model. Engineers access modeling procedures through this functionality. Capture management and production equipment definition and control management is a connection with the production world. Product structure and data management handles PLM Model of product.

Product and related knowledge definition management is highlighted in Fig.1 because this is the area of research where some recent results are reported in this paper. Human interactions are done for direct product feature definitions and knowledge definitions. Knowledge features in the product model and the PLM system are applied

for adaptive drive of product feature generation. This is the indirect way of product feature definition. The purpose of the proposed modeling is contributing to the development of a more automated and intelligent way of indirect product model feature definition in advanced PLM modeling systems.

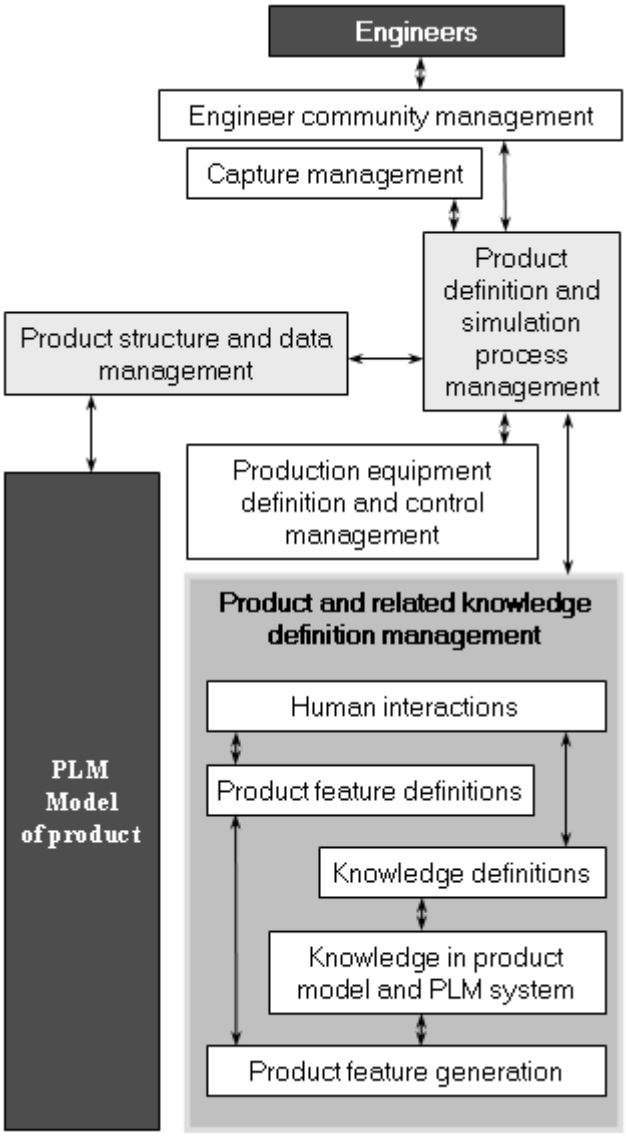


Fig. 1. Product definition in the course of product modeling

3 Engineering Problem Solving and Role of Soft Computing

Lotfi A. Zadeh who is recognized as father of Fuzzy, summarized soft computing in [1] as a set of methodologies that exploits the tolerance for imprecision and uncertainty. Key methodologies include fuzzy logic, neuro-computing, and probabilistic reasoning. The awaited results are tractability, robustness, and low cost at solutions. Inevitable, soft computing must have key importance in engineering problem solving. While soft computing has gained key role in product control processes in the form of embedded intelligence, product model based engineering problem solving in industrial PLM systems still can not utilize its strength. The door is open for new research results because advanced PLM systems can accept new object and procedure definitions for this purpose at their open surfaces.

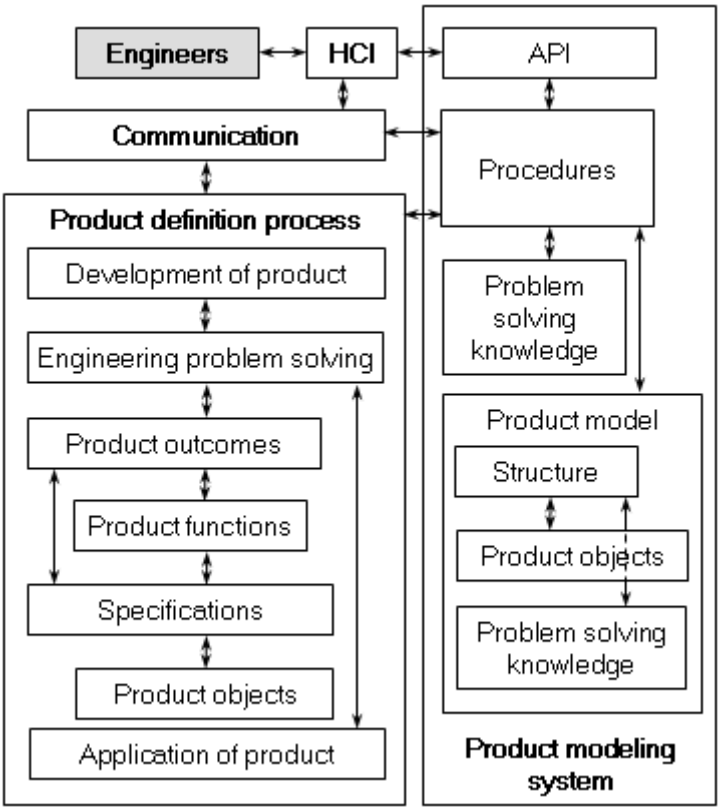


Fig. 2. Product definition in the course of product modeling

Paper [2] discusses the question how soft computing capabilities can serve engineering outcomes by the applications of fuzzy logic, genetic algorithms, and neural networks. The modeling extension explained below gives a possible answer by the definition of the outcomes in product model and the method to fulfill of outcomes in

the course of product definition. Among others, it relies upon the recognition that the product development must be supported by definition of product functions in the product model. A functional and physical representation is proposed in [3] for a better support of conceptual design than in the conventional physical and geometric representation based product modeling is possible. This concept is important towards better recognition of the non-geometrical characteristics of product. Authors of [4] recognize that conceptual design is only partially covered in PLM systems. A discontinuity can be experienced in the customer and functional requirements, product characteristics, and product design parameters information flow. Integration of quality function deployment, failure mode and effects analysis, and axiomatic design are proposed. While this can be a right way, experiences tell that much more contexts should be defined in order to achieve consistent model and modeling process.

Analyzing representative product definition in current industrial practice, main functions and system elements were concluded (Fig. 2) providing starting point for the work in this paper. Engineers communicate with model based product definition processes using suitable implementation of human computer interaction (HCI) methods. They can do required development of modeling by using of application programming interface (API) or other surfaces for this purpose. Product definition processes serve engineering problem solving for development and application of product. Engineering problem solving first defines product outcomes. Using product outcomes, product functions and specifications are defined then used at the definition of known product objects. An object is available in the applied PLM product or it must be defined in application or third party environment by using of API or purposeful communication surfaces. The answer to the question that how can this model be realized in the current advanced industrial product modeling tells the subsequent needs for research in product modeling. The role of fuzzy logic, genetic algorithm, and artificial neural network in subsequent research is inevitable.

As it is discussed in [5], Fuzzy logic should be considered as a useful tool at decision making. In engineering, human assessment includes subjectivity which cannot be expressed by numeric values. In [5], fuzzy logic is applied to handle linguistic terms. Fuzzy membership functions are used to convert linguistically expressed preferences into fuzzy numbers. This allows for application of Fuzzy operators.

A well known role of genetic algorithms in engineering is development of results by replacing a population representing an engineering result with a new generation. As an example for the application of genetic algorithm, multi-stage reverse logistics network problem is solved in [6] by using of weight mapping crossover which includes priority-based encoding method and a new crossover operator. Artificial neural network is mainly useful in learning of knowledge. Although current product modeling systems have tools to accept soft computing methods, lack of product level integration of problem solving does not allow for organized and coordinated application of soft computing.

4 Extended Integration of Knowledge in Product Model

The development of PLM systems proceeded the way of including knowledge in full feature and active context driven product model during the past decade. The word active means that including or modification of a knowledge entity automatically changes any other entity in the product model any parameter of which is in represented contextual connection with parameters of the changed entity. Because professional PLM systems are undergone quick development, correct connection of the proposed extended knowledge integration demands the definition of the concept currently prevailing modeling (CPM) in Fig. 3.

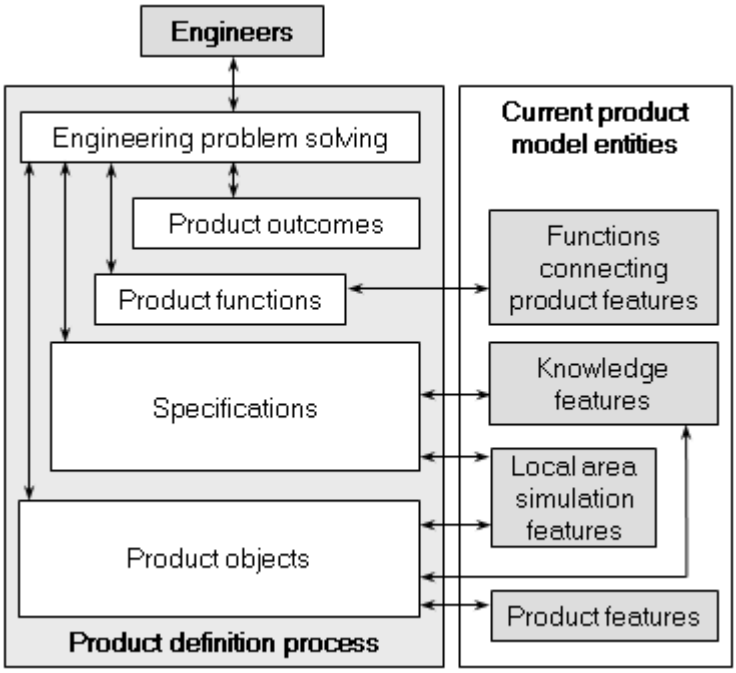


Fig. 3. Current product definition

Product definition process and product model entities for the case of CPM are introduced in Fig. 3. During the reported work, it was recognized that this model has not representation for the product function driven organization of knowledge. In order to replace the missing organization, more connections are needed in the engineering problem solving than in Fig 2. Because product outcome representation is not available in the product model to drive definition of product functions and specifications, engineer makes direct definition. Nevertheless, procedures may be available for this purpose under the supervision of engineering problem solving. Product functions are directly defined by engineer and they are included in the product model only in a structure where they connect product features. At the same time, substantial local area

analyses are available in advanced CMP products. Specifications are placed in knowledge features and local area simulation features. Product objects are represented as product features and are defined by using of knowledge and local area simulation features. Role of direct utilization of specifications by engineers during product object definition is still covers large percentage of decisions. This situation results product model which has not capability for a reveal of the applied way to decision. At the same time, thousands of product model modification interactions would require original intent and method of decisions to be changed during lifecycle of a product.

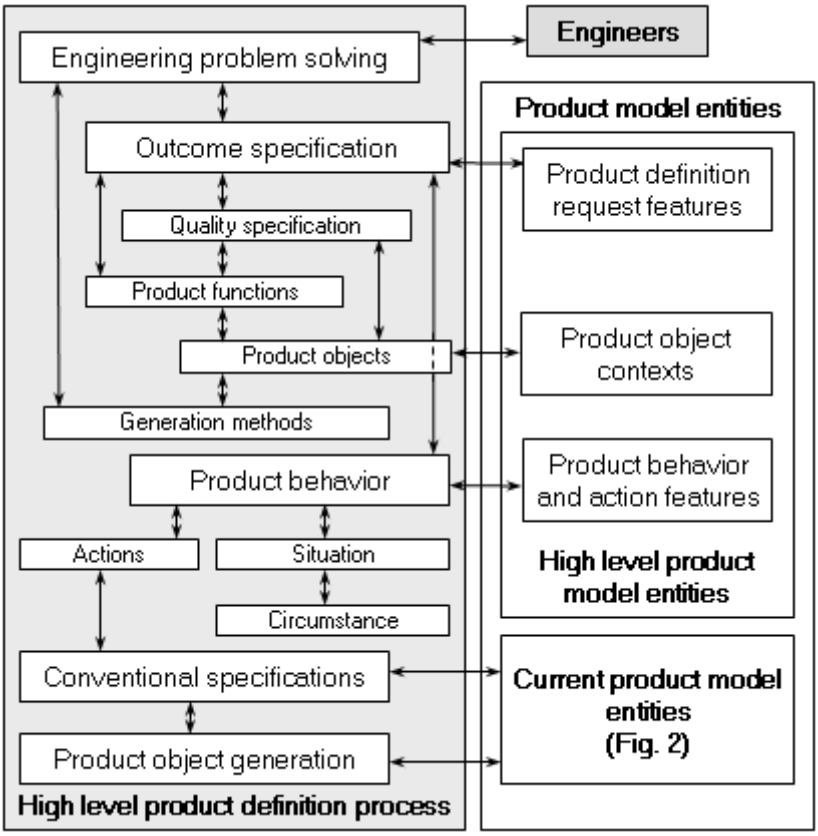


Fig. 4. High level product definition

The method which is proposed as a contribution to improving CPM modeling by enhancing its representation capabilities is called as high level product definition process (Fig. 4) in this paper. High level refers to level of decisions by using of higher level product definition knowledge in the product model. One of the new product features for this purpose must be defined for the representation of the outcome specification. In the proposed modeling, this feature has parameters in order to define contextual connections for quality specification, product function, product objects, and

generation method features. Moreover, outcomes are in contextual connection within the product model in order to gain survey on engineering problem solving for engineers in this level.

Concept of function is considered as it was introduced in [7]. Knowledge is integrated by parameters of generation method features. Product object is represented here only for its fitting to a contextual structure and refers to the relevant product feature in the CPM. The new high level product model features are representations for product definition requests, product object contexts, and product behavior and action features. High level product model entities are defined as product definition request features, product object contexts, and product behavior and action features in order to represent engineer intent, organized contextual connections of product objects, and product behaviors, respectively (Fig. 5). Actions on the current product model entities are organized to behavior definitions so that behavior controls product object definition. Main contextual connections of the high level product model entities are visualized in Fig. 5.

Product definition request is controlled by product function. Although quality specification is associated with product function, quality is defined by human for function it is not in contextual connection with product function in Fig. 5. However, PLM systems offer free definition of contexts so that the above connections can be changed depending on product type or human intent. Knowledge for the generation of product objects and their contextual connections can be included in the generation method. A traceable and identified way to solution is to be recorded. Product objects as features can be defined directly or they are selected for product function or quality specification already defined.

Product object contexts are defined in order to complete the current product model (CPM) entities by organized contextual connections for better survey able product model. One of the possible contextual structures of the product object context related features is shown in Fig. 5. Product features and context features are created in the context of context structure and parameter features. At the same time, product features are defined in the context of context features along contextual chains.

Product behavior feature is connected with other connecting behavior features in order to gain a contextual integration of behaviors those are defined for a product. Behavior is defined by situation and circumstances as it is usual in situation driven modeling mainly in automation. In other to achieve this, situation and circumstances are defined in the context of behavior. In an extended behavior definition, circumstance sets are mapped to situation variants. A circumstance set represents all information needed by the definition of the relevant feature definition actions in order to generate actions needed by the control of product object definition to fulfill behavior. Product object definition requirements are transferred by feature definition actions to the current product model entity generation processes. Product definition request features, product object contexts, and product behavior and action features include features some of them are in contextual connections. Context structure is driven by product features as they are mapped in the contextual groups of product definition request features. At the same time, product is defined in the context of product function using other features in the contextual group of product definition request features.

The proposed method applies knowledge feature representations those are available in PLM systems for modeling of experience and expertise as active features. Situation driven active product feature parameter definition is done by rules and rule sets. Rules can refer to any mathematical formula. For the purpose of connection parameters of product features, definition of arbitrary parameter is available. Event based product feature definition is available, for example, for making a feature active or inactive. The proposed modeling can be connected with the knowledge definition in CPM. At the same time, higher level object definition processes can be integrated in product model in this way.

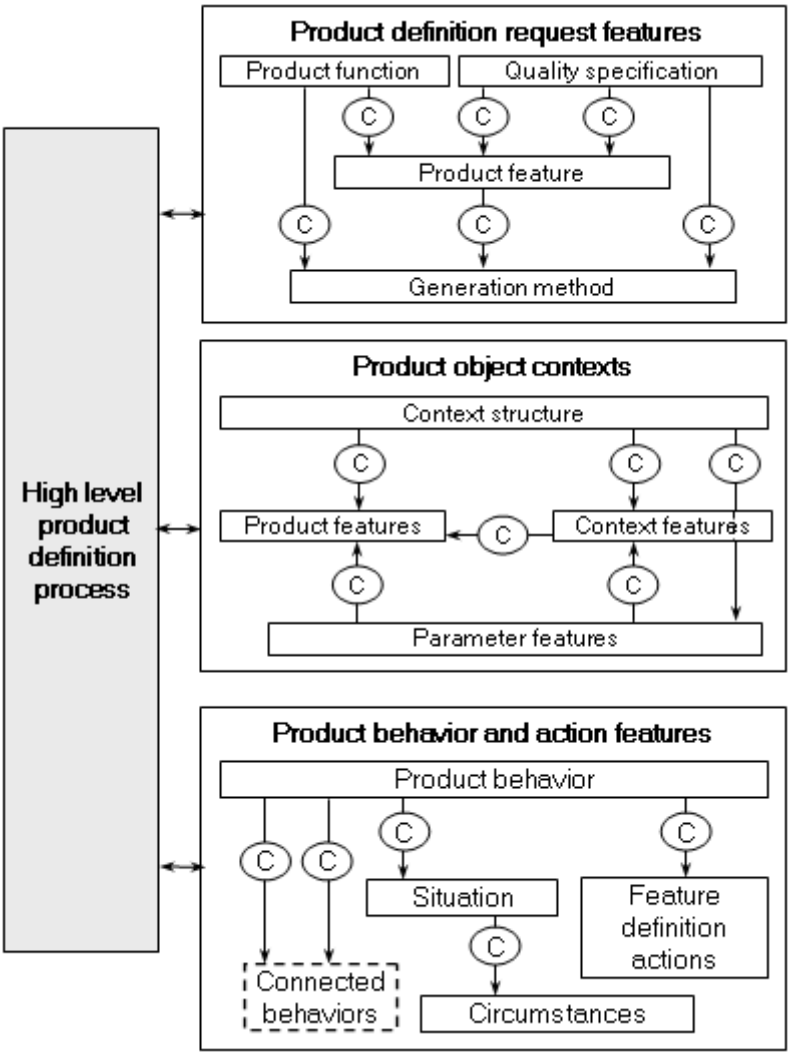


Fig. 5. High level product model entities

5 Context Structure and Implementation in PLM systems

Product behavior [8] is in the centre of the proposed modeling in order to connect outcome specification and product object generation. Previous decisions are considered as circumstances for situation. Behavior includes actions for product definition requests. Action includes knowledge and specifications for the product object generation. Conventional specifications are embedded as active knowledge in the CPM and are applied for driving product objects as it was introduced in [9].

The currently prevailing product modeling methods ensure free definition of features, feature parameters, and contexts in accordance with the actual engineering task and the decisions of engineers involved in that task. Application of context definitions in the proposed modeling extension is explained in Fig. 6. Product function PF feature has parameters $P1_{PF}$-Pn_{PF}. Parameter Pi_{PF} is the product object PO feature. When engineer defines the feature PF, first creates or selects the PO feature. When this selection is included in the model, it is defined as context *C*. Context *C* refers to relevant method for the selection. In this way, feature generation methods are included in organized system of context definitions. *C* is a context definition for extension feature. The organized system of contexts is connected to CPM modeling and serves the generation of product features and relationships amongst them.

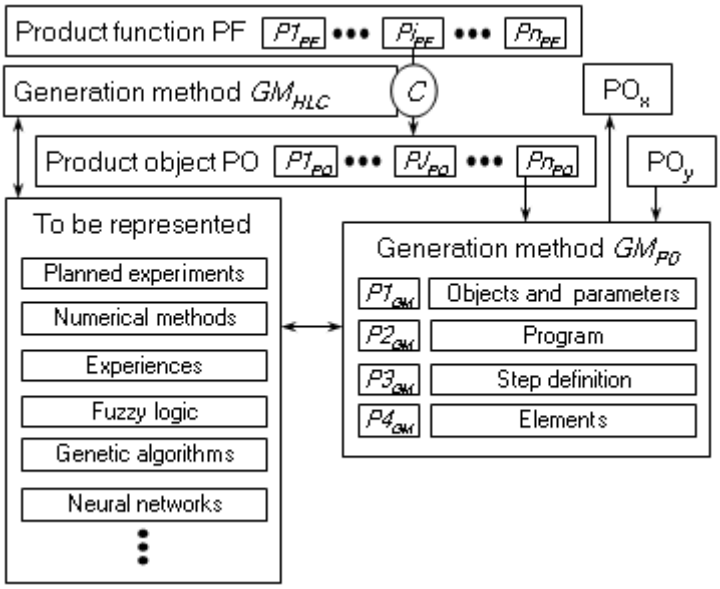


Fig. 6. Contextual connection definition in product model

Generation methods carry knowledge for the selection of features and definition of their parameters. Parameters of a generation method include product objects and their parameters, generation program consisting of contextual steps, steps, and elements for steps stored in the pool of elements. In example of Fig. 6, generation method GM_{HLC}

serves selection of product object PO for product function PF, while generation method GM_{PO} defines relationships for given parameters of product objects PO, PO_X, and PO_Y. Context is defined according to knowledge to be represented such as numerical methods, fuzzy logic, neural networks, etc. (Fig. 6).

As it is obvious from this paper, the proposed modeling supposes implementation in advanced PLM system environment. The classical solution is application programming interface (API) which ensures access to CPM objects in the current product modeling systems in order to define new objects and model object handling procedures. Recent systems are developed towards offering dedicated communication surfaces similar to those applied at model construction.

The proposed modeling has been gained more importance by the development of PLM modeling towards human definition of product function and behavior. The proposed method offers new possibility for the content based connection of product object generation and product function and behavior request.

Future research will reveal characteristics of the new extension features, their parameters, and the relevant contextual connections. The next task will be integration of knowledge in contexts as representation of product object generation methods. The main objective is integration of deeper knowledge in contextual connections in the form of soft computing procedures in order to enhance intelligence of self adaptive product definition.

6 Conclusions

Knowledge representation capabilities in current product models must be extended because they are not suitable for real connection of human request with model feature generation process. This is outstandingly true for the connection of function, specification, and generation requests with the more analyzable behavior in order to intelligent automation of product object definition. The proposed method is a contribution to efforts in application of organically integrated knowledge instead of the closed and separated knowledge based procedures.

For the above purpose, a new integration of knowledge in product model is proposed in this paper. This integration is aimed to establish a new request driven product feature generation in close communication with current product model features serving as extension to the well proven CPM product modeling in PLM systems. In the extension, product model representation is solved for engineering outcome specification. This representation is used to define behavior with circumstance defined situation. Behavior drives actions for the definition of product and conventional specification features in order to generate CPM product model features. The main contextual chain in the proposed modeling extension includes product definition request, product object context, and product behavior and action features. This chain establishes connection between specification supported model object requests, organized definitions of feature related contextual definitions and product object driving enable definition and application of product behaviors.

Acknowledgments. The authors gratefully acknowledge the financial support by the Óbuda University research fund.

References

1. Zadeh, L.A.: Soft computing and fuzzy logic. In Software 11(6), 48–56 (1994)
2. Saridakis, K.M., Dentsoras, A.J.: Soft computing in engineering design. A review. Advanced Engineering Informatics 22(2), 202–221 (2008)
3. Tay, F.E.H., Gu, J.: Product modeling for conceptual design support. Computers in Industry 48(2), 143–155 (2002)
4. Torres, V.H., Ríos, J., Vizán, A., Pérez, J.M.: Integration of Design Tools and Knowledge Capture into a CAD System: A Case Study. Concurrent Engineering 18(4), 311–324 (2010)
5. Sharon, M.: Ordoobadi: Development of a supplier selection model using fuzzy logic. Supply Chain Management: An International Journal 14(4), 314–327 (2009)
6. Leea, J.-E., Gena, M., Rhee, K.-G.: Network model and optimization of reverse logistics by hybrid genetic algorithm. Computers & Industrial Engineering 56(3), 951–964 (2009)
7. Horváth, L., Rudas I.J.: Requested Behavior Driven Control of Product Definition. In: Proc. of the 38th Annual Conference on IEEE Industrial Electronics Society, Montreal, Canada, pp. 2821–2826 (2012)
8. Horváth, L., Rudas, I.J.: Decision Support at a New Global Level Definition of Products in PLM Systems. In: Precup, R.-E., Kovács, S., Preitl, S., Petriu, E.M. (eds.) Applied Computational Intelligence in Engineering. TIEI, vol. 1, pp. 301–319. Springer, Heidelberg (2012)
9. Horváth, L., Rudas, I.J.: Active Knowledge for the Situation-driven Control of Product Definition. Acta Polytechnica Hungarica 10(2), 217–234 (2013)

PSO Optimal Tracking Control for State-Dependent Coefficient Nonlinear Systems

Fernando Ornelas-Tellez[1], Mario Graff[1], Edgar N. Sanchez[2], and Alma Y. Alanis[3]

[1] Facultad de Ingenieria Electrica
Universidad Michoacana de San Nicolas de Hidalgo
Cd. Universitaria, Morelia, 58030, Mexico
{fornelast,mgraffg}@gmail.com
[2] Centro de Investigacion y de Estudios Avanzados del IPN
Unidad Guadalajara
Av. del Bosque 1145, Zapopan, 45019, Mexico
sanchez@gdl.cinvestav.mx
[3] CUCEI, Universidad de Guadalajara
Apartado Postal 51-71, Col. Las Aguilas, Zapopan 45080, Jalisco, Mexico
almayalanis@gmail.com

Abstract. This contribution presents an infinite-horizon optimal tracking controller for nonlinear systems based on the state-dependent Riccati equation approach. The synthesized control law comes from solving the Hamilton-Jacobi-Bellman equation for state-dependent coefficient factorized (SDCF) nonlinear systems. The proposed controller minimizes a quadratic performance index, whose entries are determined by the particle swarm optimization (PSO) algorithm in order to improve the performance of the control system by fulfilling with design specifications such as bound of the control input expenditure, steady-state tracking error and rise time. The effectiveness of the proposed PSO optimal tracking controller is applied via simulation to the Van der Pol Oscillator.

1 Introduction

The optimal control approach has the aim of achieving an adequate performance of a control system by minimizing a meaningful cost functional, which can be established to evaluate the response of the state variables and the control input expenditure. A solution to the optimal control problem is obtained by solving the associated HamiltonJacobiBellman (HJB) equation, whose solution in general is rather complicated for nonlinear systems [1, 2], but if this solution exists, the control law results in an state feedback controller, which generates optimal trajectories from every initial condition [1, 4]. Indeed, in [1, 3, 5], optimal control theory is introduced as a synthesis tool to guarantee stability margins, which are basic robustness properties that a control system must possess. Moreover, the optimal control solution based on the HJB equation is more suitable for feedback design over infinite-time intervals [1, 5].

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_38,

Recently, an optimal control scheme for state-dependent coefficient factorized nonlinear systems have been proposed based on the State-Dependent Riccati Equation (SDRE) approach [6–9]. The SDRE strategy have emerged as appropriate methodology to obtain nonlinear optimal controllers, observers and filters [10]. Different successful simulations, experimental and practical applications of SDRE control have demonstrated the effectiveness of the control methodology. This control approach provides an effective algorithm for synthesizing nonlinear feedback controllers. In essence, the SDRE technique is a systematic way for synthesizing nonlinear feedback controllers, which mimic the controller synthesis as done for the linear case.

It is worth pointing out that the cost functional in the optimal control scheme depends on parameters which are usually selected by a trial-and-error procedure such that an adequate performance of the system is achieved. However, this procedure can take several iterations until the desired performance is obtained. This fact motivates to develop a mechanism (using PSO) such that the parameters of the cost functional are determined in an optimized way.

Additionally, although there exist many important results on optimal control based on the SDRE to achieve stabilization for nonlinear systems, the optimal tracking for nonlinear systems have been seldom analyzed, in spite of that for different control applications, it is required that the output of the system tracks a desired trajectory.

This paper proposes an optimal tracking control scheme for state-dependent coefficient factorized nonlinear systems, based on the HJB equation solution, which results in the solution of the SDRE. A parameters-dependent cost functional is minimized by the optimal controller, which achieves trajectory tracking to a desired reference for the controlled variables. As a contribution of this paper, the parameters for the cost functional are determined through a PSO algorithm in order to improve the performance of the control scheme. The proposed fitness function to be optimized considers the tracking error, the control input expenditure and the rise time of the controlled variables. A simulation illustrates the applicability of the proposed PSO optimal tracking control scheme.

2 Optimal Control for Nonlinear Systems

This section describes the solution to the optimal tracking control for nonlinear systems, which can be presented as a state dependent coefficient factorization. Also, concepts on controllability and observability for this class of nonlinear systems is discussed.

2.1 State-Dependent Coefficient Factorized Nonlinear Systems

Let us consider the nonlinear system

$$\dot{x} = f(x) + B(x)\,u, \qquad x(t_0) = x_0 \tag{1}$$

$$y = h(x) \tag{2}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input and $y \in \mathbb{R}^p$ is the system output; the functions $f(x)$, $B(x)$ and $h(x)$ are smooth maps of appropriate dimensions, with $f(0) = 0$ and $h(0) = 0$.

Consider that function $f(x)$ in (1) and $h(x)$ in (2) can be transformed in the SDCF representation as $f(x) = A(x)\,x$ and $h(x) = C(x)\,x$, respectively [11, 12], then system (1)-(2) results in

$$\dot{x} = A(x)\,x + B(x)\,u \tag{3}$$

$$y = C(x)\,x. \tag{4}$$

In order to obtain well-defined optimal control schemes, appropriate factorization for these representations should be determined such that controllability and observability properties are fulfilled for system (3) and output (4).

In [11, 10], the generalization of the rank test for the state-dependent factorized controllability matrix of system (3) is defined as

$$rank\,\{\mathcal{C}(x)\} = n\ \forall x \tag{5}$$

where

$$\mathcal{C}(x) = \left[B(x),\ A(x)\,B(x),\ \cdots,\ A^{n-1}(x)\,B(x)\right] \tag{6}$$

whereas the state-dependent observability matrix is defined as [10]

$$\mathcal{O}(x) = \left[C(x),\ C(x)\,A(x),\ \cdots,\ C(x)\,A^{n-1}(x)\right]^T. \tag{7}$$

Hence, factorization $A(x)\,x$ must be determined such that $\mathcal{C}(x)$ and $\mathcal{O}(x)$ have full rank, or to use duality between controllability and observability [13].

2.2 Optimal Tracking Control for Nonlinear Systems

For many applications, as aerospace, electrical machines, robotics, among others, it is important for control purposes to track a desired trajectory for the closed-loop system, then it is required that an output of the system track a desired trajectory as close as possible in an optimal sense and with minimum control effort expenditure [14, 5].

In order to deal with optimal tracking in dynamical systems, let us define the tracking error as

$$\begin{aligned} e &= r - y \\ &= r - C(x)\,x \end{aligned} \tag{8}$$

where r is a desired reference (constant or time-varying) to be tracked by the system output y.

The quadratic cost functional J to be minimized, associated to system (3), is defined in function of the tracking error as

$$J = \frac{1}{2}\int_{t_0}^{\infty} \left(e^T Q(e)\,e + u^T R(e)\,u\right)\,dt \tag{9}$$

where Q is a symmetric and nonnegative definite matrix, and R is a symmetric and positive definite matrix [14]. Matrices $Q(e)$ and $R(e)$ are design parameters that are proposed such that a desired performance or the fulfillment of design specifications are guaranteed, as will be described in Section 3.

The optimal control solution is related to determine the control $u(t),\ t \in [t_0, \infty)$, such that the criterion (9) is minimized. In this case, the solution for optimal trajectory tracking control is established as the following theorem.

Theorem 1. *Assume that system* (3)-(4) *is state-dependent controllable and state-dependent observable. Then the optimal control law*

$$u^*(x) = -R^{-1}B^T(x)\ (P\,x - z) \tag{10}$$

achieves trajectory tracking for system (3) *along a desired trajectory* r*, where* P *is the solution to the symmetric matrix differential equation*

$$\begin{aligned}\dot{P} = &-C^T\,(x)\ Q\,C\,(x) + P\,B(x)\,R^{-1}B^T(x)\,P \\ &-A^T(x)\,P - P\,A(x)\end{aligned} \tag{11}$$

and z *is the solution to the vector differential equation*

$$\dot{z} = -\left[A(x) - B(x)\,R^{-1}B^T(x)\,P\right]^T z - C^T\,(x)\ Q\,r \tag{12}$$

with boundary conditions $P(\infty) = 0$ *and* $z(\infty) = 0$*, respectively. Control law* (10) *is optimal in the sense that it minimizes the cost functional* (9)*, which has an optimal value function given as*

$$J^* = \frac{1}{2}x^T(t_0)\,P(t_0)\,x(t_0) - z^T(t_0)\,x(t_0) + \varphi(t_0) \tag{13}$$

where φ *is the solution to the scalar differentiable function*

$$\dot{\varphi} = -\frac{1}{2}r^T Q\,r + \frac{1}{2}z^T\,B(x)\,R^{-1}B^T(x)\,z \tag{14}$$

with $\varphi(\infty) = 0$.

Proof. For details of the proof see [6]. □

3 PSO Optimal Tracking Control

As described in Section 2.2, the cost functional (9) depends on the parameters $Q(e)$ and $R(e)$. Usually, these parameters are determined by a trial-and-error tuning method by considering the transient and steady-state performance of the system variables, which is a time-consuming process often taking a few iterations before a desirable response can be obtained.

This paper proposes to determine the entries of $Q(e)$ and $R(e)$ for (9) through the PSO algorithm such that the following measures are considered:

1. An amount of steady-state tracking error;
2. An amount of the control input expenditure;
3. Rise time (T_r): the time required for the controlled variable to be settled from the 10% to the 90% of its final value.

3.1 PSO Algorithm

The PSO algorithm was developed in [17] as an optimization algorithm based on social simulation models to determine the minimum of nonlinear functions by iteratively improving a candidate solution with regard to a given fitness function (measure of performance) [17, 18]. In the PSO algorithm, the swarm is defined as a set:

$$S = \{s_1, s_2, \ldots, s_N\} \tag{15}$$

of N particles (candidate solutions), defined as:

$$s_j = (s_{j1}, s_{j2}, \ldots, s_{jn})^T \in A, \quad j = 1, 2, \ldots, N \tag{16}$$

where n is the number of particle components. The objective function, $F(s)$, is assumed to be available for all points in A. The particles are assumed to move within the search space, A, in an iterative way, which is possible by adjusting their position using a proper shift, called velocity, and denoted as:

$$v_j = (v_{j1}, v_{j2}, \ldots, v_{jn})^T, \quad j = 1, 2, \ldots, N. \tag{17}$$

Velocity is also adapted iteratively to render particles capable of potentially visiting any region of A. Velocity is updated based on information obtained in previous steps of the algorithm. This is implemented in terms of a memory, where each particle can store the best position it has ever visited during its search. For this purpose, besides the swarm, S, which contains the current positions of the particles, PSO maintains also a memory set:

$$P_s = \{p_1, p_2, \ldots, p_N\} \tag{18}$$

which contains the best positions:

$$p_j = (p_{j1}, p_{j2}, \ldots, p_{jn})^T \in A, \quad j = 1, 2, \ldots, N \tag{19}$$

ever visited by each particle. These positions are defined as:

$$p_j(r) = \arg\min_r F_j(r). \tag{20}$$

The algorithm approximates the global minimizer with the best position ever visited by all particles. Let * be the index of the best position with the lowest function value in P at a given iteration r

$$p^*(r) = \arg\min_r F(p_j(r)). \tag{21}$$

Then, the PSO is defined by the following equations [17]:

$$v_{jl}(r+1) \;=\; v_{jl}(r) + c_1\gamma_1(p_{jl}(k) - s_{jl}(r)) + c_2\gamma_2(p_j^*(r) - s_{jl}(r)), \tag{22}$$

$$s_{jl}(r+1) = s_{jl}(r) + v_{jl}(r+1) \tag{23}$$

$$j = 1, 2, \ldots, N, \qquad l = 1, 2, \ldots, n$$

where r denotes the iteration counter; γ_1 and γ_2 are random variables uniformly distributed within $[0, 1]$; and c_1, c_2, are weighting factors, also called the cognitive and social parameter, respectively [18].

The last two terms on the right side in (22) enable each particle to perform a local search around its individual best position p_{jl} and the swarm best position p_j^*. The first term on the right side in (22) allows each particle to perform a global search by exploring a new search space.

In this paper, the PSO algorithm is used for the controller (10)–(12) by defining the components of a particle as the entries of matrices $Q(e)$ and $R(e)$.

4 PSO Optimal Tracking Application to the Van der Pol Oscillator

Consider the model of the Van der Pol Oscillator [19]

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + \frac{1}{2}\left(1 - x_1^2\right) x_2 + (2 + \cos(0.1\, x_1))\, u. \end{aligned} \tag{24}$$

In order to illustrate the optimal control tracking scheme, let us define the nonlinear outputs for (24) as $y_1 = x_1$ and $y_2 = x_1\, x_2$.

Note that the system (24) can be presented as

$$\begin{aligned} \dot{x} &= A(x)\, x + B(x) u \\ y &= C(x)\, x \end{aligned}$$

where $x = [x_1,\, x_2]^T$, with the matrices defined as

$$A(x) = \begin{bmatrix} 0 & 1 \\ -1 & \frac{1}{2}\left(1 - x_1^2\right) \end{bmatrix}; \quad B(x) = \begin{bmatrix} 0 \\ 2 + \cos(0.1\, x_1) \end{bmatrix}; \quad C(x) = \begin{bmatrix} 1 & 0 \\ 0 & x_1 \end{bmatrix}.$$

Consider the references for $y = [y_1,\, y_2]^T$ as $r = [r_1,\, r_2]^T = [10,\, 0]^T$, and the error defined as $e = r - y$. The control scheme to achieve optimal tracking is stated by (10)–(12), and the entries of $Q(e)$, are defined as $Q(e) = \begin{bmatrix} \frac{\alpha}{\beta \|y_1 - r_1\| + \gamma} & 0 \\ 0 & 1 \end{bmatrix}$ and $R(e) = \ell$, where α, β, γ and ℓ are positive constants.

The performance specifications for the Van der Pol oscillator are: 1) less than 1% steady-state tracking error; 2) less than 100 for the control input value and 3) less than 3 seconds for the rise time T_r.

The PSO algorithm calculates the optimized values for α, β, γ and ℓ, subject to the fitness function

$$F(s) = 330\,(r_1 - y_1)^2 + x_2^2 + u^2 + 5\, T_r$$

such that the performance specifications are guaranteed. The obtained optimized values are $\alpha = 93.1213$, $\beta = 0.0017$, $\gamma = 0.1327$ and $\ell = 1.2685$. In is worth mentioning the PSO algorithm facilitates the determination of these parameters, if not, a trial-and-error methodology should be used to select the parameters, which is a time consuming and inefficient task.

Considering the optimized parameters for $Q(e)$ and $R(e)$, the optimal trajectory tracking for x_1 is displayed in Fig. 1, for which the desired reference is $r_1 = 10$. Additionally, the time-response for x_2 and the optimal control law evolution are displayed in this figure.

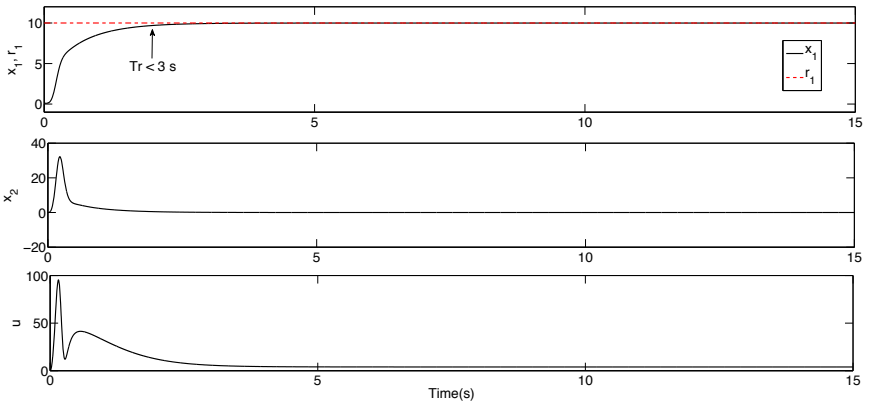


Fig. 1. Time response for x_1, x_2 and control law u

5 Conclusions

This paper has established the optimal control for tracking of state-dependent coefficient factorized nonlinear systems and minimizing a parameters-dependent cost functional, whose parameters are determined by using a PSO algorithm in order to improve the performance of the control scheme. The proposed controller results in a state feedback optimal control law plus a time-varying term. The applicability of the proposed approach is illustrated via simulations, achieving an optimized tracking for the Van der Pol Oscillator.

References

1. Sepulchre, R., Jankovic, M., Kokotović, P.V.: Constructive Nonlinear Control. Springer, Berlin (1997)
2. Krstić, M., Tsiotras, P.: Inverse optimal stabilization of a rigid spacecraft. IEEE Transactions on Automatic Control 44(5), 1042–1049 (1999)
3. Freeman, R.A., Kokotović, P.V.: Robust Nonlinear Control Design: State-Space and Lyapunov Techniques. Birkhauser Boston Inc., Cambridge (1996)

4. Primbs, J.A., Nevistic, V., Doyle, J.C.: Nonlinear optimal control: A control Lyapunov function and receding horizon perspective. Asian Journal of Control 1, 14–24 (1999)
5. Anderson, B.D.O., Moore, J.B.: Optimal Control: Linear Quadratic Methods. Prentice-Hall, Englewood Cliffs (1990)
6. Ornelas-Tellez, F., Rico, J.J., Ruiz-Cruz, R.: Optimal Tracking for State-Dependent Coefficient Factorized Nonlinear Systems. Asian Journal of Control, n/a–n/a (2013), doi:10.1002/asjc.761
7. Ornelas-Tellez, F., Rico, J.J., Rincon- Pasaye, J.J.: Optimal control for non-polynomial systems. Journal of the Franklin Institute 350(4), 853–870 (2013)
8. Erdem, E.B.: Analysis and real-time implementation of state-dependent Riccati equation controlled systems. Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA (2001)
9. Cimen, T.: State-dependent Riccati equation (SDRE) control: A survey. In: Proceedings of the 17th World Congress, The International Feredation of Automatic Control, Seoul, Korea, pp. 3761–3775 (2008)
10. Banks, H.T., Lewis, B.M., Tan, H.T.: Nonlinear feedback controllers and compensators: a state-dependent Riccati equation approach. Computational Optimization and Applications 37(2), 177–218 (2007)
11. Hammett, K.D., Hall, C.D., Ridgely, D.B.: Controllability issues in nonlinear state dependent Riccati equation control. Journal of Guidance, Control and Dynamics 21(5), 767–773 (1998)
12. Cloutier, J.R., D'Sousa, C.N., Mracek, C.P.: Nonlinear regulation and nonlinear H_∞ control via the state-dependet Riccati equation technique: Part 1, theory. In: Proceedings of the First Internation Conference on Nonlinear Problems in Aviation and Aerospace, Daytoba Beach, FL, USA (1996)
13. Hermann, R., Krener, A.: Nonlinear controllability and observability. IEEE Transactions on Automatic Control 22(5), 728–740 (1977)
14. Athans, M., Falb, P.L.: Optimal Control: An Introduction to the Theory And Its Applications. McGraw Hill, New York (1966)
15. Kirk, D.E.: Optimal Control Theory: An Introduction. Prentice-Hall, Englewood Cliffs (1970)
16. Kwakernaak, H., Sivan, R.: Linear Optimal Control Systems. Wiley and Sons, Inc. (1972)
17. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, Washington, DC, USA, vol. 4, pp. 1942–1948 (1995)
18. Parsopoulos, K.E., Vrahatis, M.N.: Particle Swarm Optimization and Intelligence: Advances and Applications. Information Science Reference, Hershey, NY, USA (2010)
19. Nevistić, V., Primbs, J.A.: Constrained nonlinear optimal control: a converse HJB approach. Tech. Rep. CIT-CDS 96-021. California Institute of Technology. Pasadena, CA, USA (1996)

Delphi-Neural Approach to Clinical Decision Making: A Preliminary Study

Ki-Young Song[1] and Madan M. Gupta[2]

[1] Department of Mechanical Engineering
The University of Tokyo
Tokyo, Japan
sky8071@gmail.com
[2] Intelligent Systems Research Laboratory
College of Engineering
University of Saskatchewan
Saskatoon, Canada
Madan.gupta@usask.ca

Abstract. In clinical practice, making diagnostically crisp decisions is critical to successful treatment outcomes. However, there is no agreement on the operational methodology that is best suited to convert imprecise symptomatic information into crisp clinical treatment decision making. In this paper, a new computational decision making tool, Delphi-Neural Decision Making Processor (D-NDMP), is introduced as a preliminary study to apply to clinical practices for more successful and efficient operational decisions. A case study in a dental clinical decision involving a deep decay tooth is presented as an example to perform D-NDMP. The results yield a more reliable and confident opinion on the practical application of treatment decision in uncertain cases in a clinical decision making process.

1 Introduction

In our daily life, we make frequent decisions based on our past experiences which we are consciously or unconsciously subject to. Decision making may solve a problem or make it worse, but that is only revealed once an activity has been applied. In professional fields, however, such as health care, we rely on experts to make the best possible decisions and thereby lessen the chance of an unfortunate outcome following the treatment. Therefore, it is natural to develop a method to capture the best expert knowledge to ensure as much as possible that we have successful and effective clinical practices in diagnosis, monitoring and interventions.

Clinical decision making is conventionally a cognitive heuristic process: assessment (through data gathering, assimilation, and analysis), judgment (through evaluation and choice), and operational decision. Heuristics may result in a feasible solution but may cause a substantial decision error due to bias [1]. Also, clinical decisions involve uncertainties and trade-offs. The uncertainties may be

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_39,

from the accuracy of diagnostic tests, the natural history of a disease, the effect of treatment in an individual patient, or the effects of an intervention in a group or population as a whole [2]. Considering the complexity, clinical decision making becomes a crucial and difficult procedure.

In this paper, we present a new computational decision making model, named Delphi-Neural Decision Making Processor (D-NDMP), to assist clinicians to make more successful and efficient decisions. In this model, we propose a significant modification to the neural inputs and synaptic weights by employing the Delphi technique. This Delphi-neural approach increases the reliability and the confidentiality of clinical decision making processes by quantitatively adapting opinions of professional clinicians and practitioners. As a preliminary study, a dental decision making process is introduced as an example in this paper. From the result, D-NDMP shows very promising assistance to the clinical decision making process.

2 Methods

In a decision making process, the qualitative (descriptive) terms, such as 'very', 'normal', 'so so', and others, which are common in clinical encounters, should be transformed into numerical scores to reflect the subjective qualitative evaluations of experts (professional clinicians and practitioners). The descriptive terms are rather fuzzy (not crisp) making it difficult to make a crisp decision. It is more precise to use numbers to represent an opinion. In this study, we use scores between -5 and 5 to evaluate clinical factors (categories) for operational decisions. In this way, the limitation of point scales (such as losing information by few points and cognitive overload by too many points [3]) can be overcome. For example, if a clinician has an 'extremely negative' opinion on a category for a decision, the clinician would score between -5 and -4 on the category.

Table 1 describes the starting position in such a basic clinical decision making process. At the top of the table, a problem is defined for an operational decision. Regarding the definition of the problem, n questionnaires are formulated as categories (clinical factors) with m experts (professional clinicians). The clinicians score the categories between -5 and 5 in the shaded area of the table, **C**, considering the definition of the problem. The average, **A**, of each category is calculated as

$$A_{mn} = \frac{1}{m}\left(\sum_{i=1}^{m} C_{in}\right), n \in N \tag{1}$$

Then, D-average, **D**, is computed by adapting the Delphi technique to improve the reliability of the opinions from the experts. D represents the value of agreeing opinions from experts. The Delphi technique (or Delphi method) is a structured communication technique with a panel of experts. It was originally developed in the 1950s by the Rand Corporation, Santa Monica, California, for use in operation research. The schematic diagram of the Delphi technique is shown in Fig. 1. Experts are asked to give their opinions on questionnaires in the first round.

Table 1. Delphi-neural approach to clinical decision making

Definition of problem				(score: $-5 \sim 5$)
Categories / Clinitians	Category#1	Category#2	$\cdots$	Category#n
Clinician#1	C_{11}	C_{12}	$\cdots$	C_{1n}
Clinician#2	C_{21}	C_{22}	$\cdots$	C_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
Clinician#m	C_{m1}	C_{m2}	$\cdots$	C_{mn}
Average (**A**)	A_1	A_2	$\cdots$	A_n
D-average (**D**)	D_1	D_2	$\cdots$	D_n

In the next round, the experts receive other opinions anonymously as a feedback and then give new opinions under the influence of their colleagues' opinions. This expert survey is repeated several times until a consensus occurs. This technique delivers quantitative as well as qualitative results by using explorative, predictive and normative opinions from experts. Thus, this technique is a relatively stronger process for use with naturally unsure and incomplete information [4]. As aforementioned, the core of the Delphi technique is the 'anonymous' circulation of the expert opinions with proper reasoning as internal feedbacks. The anonymous circulation can be expressed as

$$C_{mn}(k+1) = C_{pn}(k) \tag{2}$$

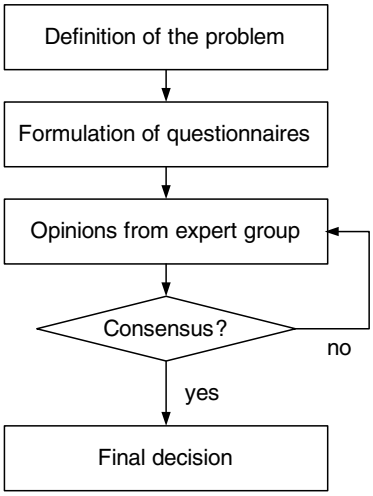


Fig. 1. Schematic diagram of Delphi technique

where k is number of round, p is randomly permuted number ($p \in 1, 2, \ldots, m$). Then, average opinions, $\mathbf{D}$, can be updated by the feedback as

$$D_n(k+1) = \frac{1}{m}\left(\sum_{l=1}^{m} \frac{C_{ln}(k) + C_{ln}(k+1)}{2}\right) \tag{3}$$

The feedback process can be represented by averaging the opinion from an expert and an anonymously circulated opinion. The reliability of averaged opinions, R, can be calculated by using variance of the opinions as

$$R_n(k) = \frac{1}{var\left(C_{mn}(k)\right)} \tag{4}$$

where $var(\bullet)$ represents variance function. As the variance of C_{mn} decreases, the reliability of the average opinion of C_{mn} increases. The circulation should stop in the condition as

$$\forall R_j > r_d, \qquad j = 1, 2, \cdots, n \tag{5}$$

where r_d is a desired reliability defined by a user. However, it is more convenient to appoint a desired variance as an allowable variance. Thus, a consensus of opinion for each category can be achieved with final value of $\mathbf{D}$.

The next process of clinical decision making is to analyze the consensus by a classification method. Neural networks are one of the most powerful tools for classification. Neural networks were inspired by the study of biological neurons. By employing synaptic operation and somatic operation, neural networks provide a superiority of identification and classification and have therefore been widely used in research fields [5]. In this paper, we do not consider the dynamic feedback (such as back-propagation algorithm) in neural networks but apply static neural networks to the clinical decision making process. The schematics of the proposed process is illustrated in Fig. 2

A neural unit (single artificial neuron) is composed of synaptic operation, v, and somatic operation, y_N, and they are expressed as

$$v = \sum \mathbf{XW} \tag{6}$$

$$y_N = \Phi[v] \tag{7}$$

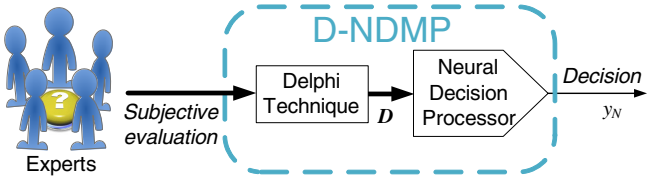


Fig. 2. Schematics of the process flow of D-NDMP

where **X** is input vector, **W** is weight vector, and $\Phi[\bullet]$ is a mapping function. Conventionally, a sigmoid function is chosen as a mapping function due to its special characteristics exhibiting a progression from small beginning to accelerated end as natural processes [6]. In clinical decision making, D-average **D** replaces **X**, and the value of **W** is assigned by the professional clinicians (experts) between 0 and 1 with agreement. The Delphi technique is applied to evaluate the importance of the clinical factors (categories) for the given problem (the more important, the higher number). Thus, the synaptic operation for the Delphi-neural approach can be rewritten as

$$v = \sum \mathbf{DW} \tag{8}$$

3 Case Study: Operational Decision in a Dental Clinic Practice

Decision making in dental clinics is not easy for dentists or patients. One of the most common complex dilemmas is an operational decision on deep tooth decay. In this case, mainly two choices are available: root canal treatment or implant (see Fig. 3). The former operation saves the natural tooth with a permanent filling after removing the damaged pulp (which includes the nerve and blood vessels) of the tooth. It is necessary to remove all of the pulp in the tooth to avoid further infection. Sometimes, a cap on the treated tooth is necessary to protect the tooth from a heavy load. However, sometimes the removing process cannot be perfectly performed due to the shape of the root canals. The latter operation extracts the damaged tooth and replaces it with an artificial tooth. Dental implants are titanium tooth roots that anchor an artificial tooth to the dental bone. Replacing a tooth takes time and planning. Other factors such as gum disease and bad bite may cause an implant to fail prematurely. Among professional dental clinicians and practitioners, there are no absolutes applicable to this complex decision making process [7,8].

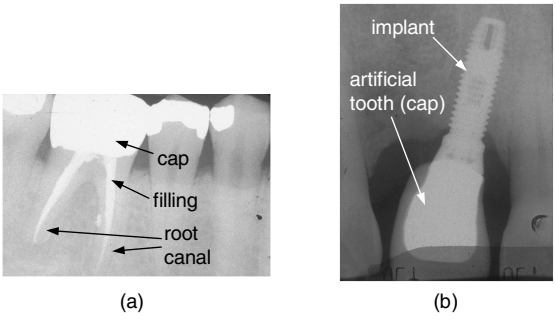


Fig. 3. Dental treatment for deep tooth decay: (a) root canal treatment and (b) implant [8]

Now we apply the proposed Delphi-neural decision making processor (D-NDMP) to this clinical decision making. In order to perform the process by D-NDMP, we first need to define the clinical problem in order to formulate questionnaires (categories). An example is presented for a case study below.

> A healthy man had a very sensitive molar. A few days later, he felt pain around the molar and went to see a dentist. The dentist (d1) found that the root of the molar was infected by a virus. The dentist (d1) decided to perform a root canal treatment on the tooth after discussing the situation with the patient. While the dentist (d1) was drilling a hole on the tooth, he found a crack on the tooth. The crack has not shown on the X-Ray picture taken before the process. The dentist (d1) informed the patient about the crack and decided to continue the root canal treatment on the next visit. The patient visited another dentist (d2) for a second opinion on the tooth. The second dentist (d2) examined the tooth carefully and discovered that the crack continued into the gum. The second dentist (d2) then suggested an implant process to the patient since in his opinion the crack probably continued to the deep end of the tooth. The second dentist (d2) explained to the patient that in this case root canal treatment would not be sufficient to treat the tooth permanently. The patient visited a third dentist (d3) who concluded that root canal treatment would be a suitable treatment for the tooth.

The problem in the given example is to select the better treatment for the damaged tooth: root canal treatment or implant. In order to select one alternative using D-NDMP, root canal treatment is scored in positive numbers ($1 \sim 5$, the higher number being a stronger recommendation). The implant option is scored in negative numbers ($-5 \sim -1$, the lower number being a stronger recommendation). The neutral opinion is scored as 0. Three experts are associated with different opinions on the treatment. The clinical factors (categories, $\mathbf{F}$) for the operational decision and the weights ($\mathbf{W}$) of the factors would be:

- F1: importance of infection in the tooth for the treatments (W1: 1)
- F2: importance of crack on the tooth for the treatments (W2: 1)
- F3: importance of mobility (shakiness) of the tooth for the treatments (W3: 0.1)
- F4: importance of the patients health for the treatments (W4: 0.8)
- F5: predictability of the success of the treatments (W5: 1)
- F6: risk of the infection after the treatments (W6: 0.8)
- F7: finances for the treatments (W7: 0.6)
- F8: aesthetic after the treatments (W8: 0.8)

The clinical factors and weights would vary in a real practical case. The given factors and numbers are reasonably selected for this case study. The weights ($\mathbf{W}$) of the factors vary between 0 and 1 which would be defined after agreement of the experts opinions. F3 has very low weight since the damaged tooth is rarely mobile (shaking). The complete table of decision making for this example is shown in Table 2.

Table 2. D-NDMP to decision making in dental operational decision

Root canal treatment $(1 \sim 5)$ **or Implant** $(-5 \sim -1)$ (score: $-5 \sim 5$)				
Categories / Clinicians	F1 (W1: 1)	F2 (W1: 1)	F3 (W1: 0.1)	F4 (W1: 0.8)
d1	5	2	1	5
d2	−5	−5	−1	−4
d3	2	−2	0	2
Average (**A**)	0.67	−1.67	0	1
D-average (**D**)	0.04	−0.42	0	0.06

Root canal treatment $(1 \sim 5)$ **or Implant** $(-5 \sim -1)$ (score: $-5 \sim 5$)				
Categories / Clinicians	F5 (W1: 1)	F6 (W1: 0.8)	F7 (W1: 0.6)	F8 (W1: 0.8)
d1	4.5	3	5	3
d2	−4	−2	−4	−5
d3	2	0	3	0
Average (**A**)	0.67 0.83	0.33	1.33	−0.67
D-average (**D**)	0.05	0.02	0.08	−0.04

3.1 Conventional Approach (Averaging)

Conventionally, the average value, **A**, would be applied to make a decision. For the calculation, **A** was applied as neural inputs with given weights and sigmoid function. As the result with **A**, the root canal treatment is highly recommended for this patient. The result is shown in Fig. 4. The blue line represents the mapping function between −5 and 5. The red dot is the final neural output (y_N), and the output represents how the final decision is confident with the two treatment, implant (−1) and root canal treatment (1), respectively. In this example, the value of synaptic operation (v) is 1.17, and y_N is 0.82 with average, **A**, as neural input, which means that it is highly recommended (about 82%) that the patient should take the root canal treatment as operational decision. The decision may be acceptable; however, the reliability of the decision is still questionable.

3.2 D-NDMP Approach

Now we perform D-NDMP to make a decision for the same example. MATLAB was used to develop D-NDMP. In order to increase the reliability of the processor, the tolerance for matrix **C** was initially set to 0.1. That means that the circulation round for the Delphi technique continues until the variance in each category becomes less than the tolerance. D-average was achieved with little variances (< 0.1) for each category, which yielded $v = -0.24$ and $y_N = -0.24$. This result recommended that the patient might take the implant operation, which is a different opinion from that of the conventional decision making. Furthermore, the decision may be reconsidered since the value of the neural output

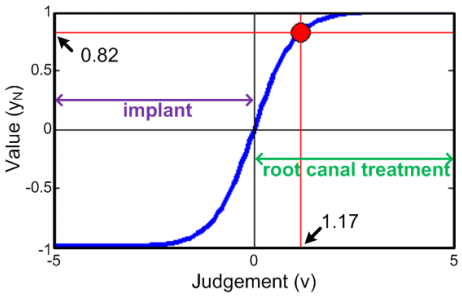


Fig. 4. Result of the decision making by a conventional method, averaging the scores. From the result, the root canal treatment is certainly recommended (82%).

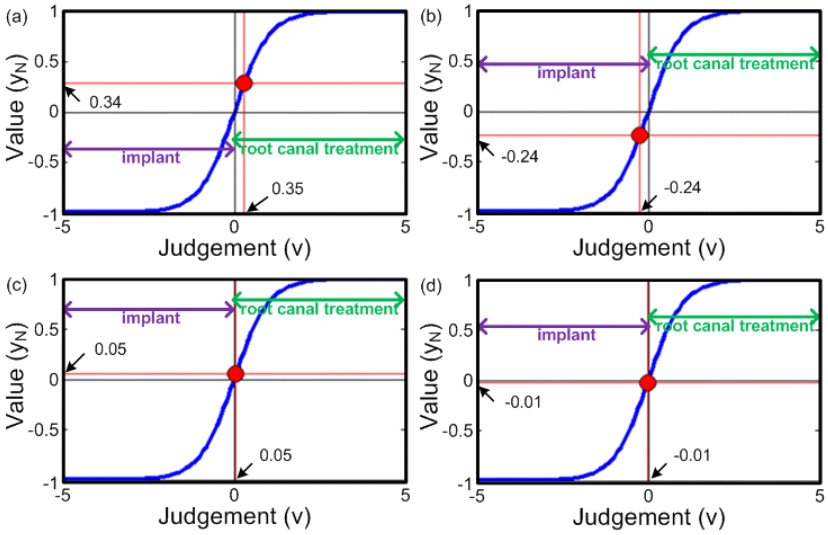


Fig. 5. Results of D-NDMP by employing different values of allowable variances. (a) $var(\bullet) < 1$, (b) $var(\bullet) < 0.1$, (c) $var(\bullet) < 0.01$, and (d) $var(\bullet) < 0.001$. ($var(\bullet)$ represents variance of $\mathbf{C}$ after the Delphi technique).

is not convincing (< 0.4). It was also observed that as the tolerance changes, the decision of D-NDMP are altering (see Fig. 5). That indicates that operational decision is still debatable, which indicates that the conventional decision may not be a right choice.

4 Conclusions

In this paper, we propose a new computational decision making method by developing a Delphi-neural decision making processor (D-NDMP) as a preliminary

study. The Delphi technique was adapted to increase the reliability of the decision, and neural networks were employed for classification. D-NDMP provides an adaptive method on consensus opinions to improve quantitative as well as qualitative decision making in clinical decision making processes. An operational decision making in a clinical dental practice was considered as a case study in this paper. The result of the case study shows that as the reliability of the information increased, the decision became more neutral, which indicates that both treatments are still in debate with current experts and categories. The model developed herein will next to be tested with actual clinical data.

References

1. Hall, M., Noble, A., Smith, S.: A Foundation for Neonatal Care: A Multi-disciplinary Guide. Radcliffe Publishing Ltd. (2009)
2. Hunink, M.G.M.: Decision Making in Health and Medicine: Integrating Evidence and Values. Cambridge University Press (2001)
3. Groves, R.M., Fowler, F.J., Couper, M.F., Lepkowski, J.M., Singer, E.: Survey Methodology, 2nd edn. Wiley (2009)
4. United Nations Industrial Development Organization: Delphi Method, http://www.unido.org/fileadmin/import/16959_DelphiMethod.pdf
5. Hou, Z.-G., Song, K.-Y., Gupta, M.M., Tan, M.: Neural Units with Higher-Order Synaptic Operations for Robotic Image Processing Applications. Soft Computing – A Fusion of Foundations, Methodologies and Applications 11, 221–228 (2007)
6. Bukovsky, I., Hou, Z.-G., Bila, J., Gupta, M.M.: Foundations of Nonconventional Neural Units and Their Classification. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) 2, 29–43 (2008)
7. Leader, D.: Root Canal Treatment or Dental Implant – Which is Best? (2007), http://www.associatedcontent.com/article/345411/root_canal_treatment_or_dental_implant_pg3.html?cat=5
8. Bader, H.I.: Treatment planning for implants versus root canal therapy: A contemporary dilemma. Implant Dentistry 11, 217–223 (2002)

Contextual Bipolar Queries

Sławomir Zadrożny[1], Janusz Kacprzyk[1,2], Mateusz Dziedzic[2], and Guy De Tré[3]

[1] Systems Research Institute, Polish Academy of Sciences
Janusz.Kacprzyk@ibspan.waw.pl
[2] Dept. of Electrical and Computer Engineering, Cracow University of Technology
Mateusz.Dziedzic@ibspan.waw.pl
[3] Dept. of Telecommunications and Information Processing, Ghent University
Guy.DeTre@UGent.be

Abstract. A widespread and growing use of information systems, notably databases, calls for formalisms to specify user preferences richer than classical query languages. The concept of bipolarity of preferences is recently considered crucial in this respect. Its essence consists in considering separately positive and negative evaluations provided by the user which are not necessarily a complement of each other. In our previous work we have proposed an approach based on a specific interpretation of the positive and negative evaluations and their combination using a non-standard logical connective referred to as "and possibly". Here we propose a novel extension of that approach. We present the concepts, possible interpretations and some analyses.

Keywords: flexible database query, bipolar query, context in querying.

1 Introduction

We consider here database querying as an action, or process, going beyond a simple retrieval of data satisfying a standard condition(s) as, e.g., a price lower than a threshold, location in one of favorite districts, etc. To better present the essence of our new approach, we relate our concepts and analyses to an intuitive example of querying a real estate database. In such a context the user's interests, intentions and preferences are usually much more sophisticated and their adequate representation using classical query languages is not easy, if possible at all. One step towards such a better representation is the concept of the flexible fuzzy queries [1] which make it possible to use linguistic terms in queries with their semantics provided by appropriate fuzzy sets. The user may then express in a more natural way his or her preferences as to, e.g., the price of the property sought, just stating that the property should be *cheap* instead of artificially defining a threshold distinguishing acceptable prices from those to be rejected. Recently, a further step towards even more human consistent representation of the user preferences has been proposed in the form of *bipolar queries*, termed so by Dubois and Prade [2]. The motivation for that type of queries has been an attempt to grasp the negative and positive aspects which may be independently provided by the user to characterize data he or she is looking for. The bipolar

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_40,

queries may be considered in their most general form as consisting of two conditions separately accounting for these negative and positive assessments. However in the literature a specific interpretation of bipolar queries seems to prevail so far in which the negative and positive condition correspond to a *constraint* and a *wish*, respectively. There are different possible interpretations of such a constraint and wish. We have earlier [3,4] developed an interpretation referred to as the *required/desired semantics* which is a fuzzy extension of the seminal work of Lacroix and Lavency [5].

In this paper, we propose a further extension of this type of bipolar queries and we show its relation to various approaches known in the literature.

The structure of the paper is as follows. In the next section we briefly remind the essence of the the concept of a bipolar query with an emphasis on its special version based on the required/desired semantics. Section 3 comprises the main contribution of this paper introducing the concept of contextual bipolar queries. Finally, we conclude summarizing the content of the paper and envisaging the lines of possible further research.

2 Bipolar Queries

Some studies [6] show that while expressing his or her preferences the human being is contemplating separately *positive* and *negative* features of alternative courses of action, objects, etc. Applying this paradigm to database queries one should consider two types of conditions: the satisfaction of one of them makes a tuple desired while the satisfaction of the second makes it to be rejected. This is the most general form of what we will refer to as a *bipolar query*. Moreover, we will assume that both conditions are expressed using fuzzy predicates and thus may be satisfied to a degree, from 0 to 1. In the most general form mentioned above one obtains two satisfaction degrees, of the positive and negative condition, respectively, and then needs some means to compare such pairs of degrees to rank tuples in an answer to the query; cf., e.g. [7].

We will consider here a special class of bipolar queries where the positive and negative condition are interpreted in a specific way as that the data items sought definitely have to satisfy the complement of the latter condition, while the former condition are somehow less stringent. For example, a hotel sought by the user may be required to be cheap and – but only *if possible* – (desired to be) comfortable. The negative condition is here "not cheap" while the positive condition is "comfortable". We specify the complement of the negative condition (denoted C) which may be interpreted as a *required* condition. On the other hand, the positive condition is directly expressed and may be referred to as a *desired* condition (denoted P); the whole query is denoted (C, P). Both C and P are generally specified as fuzzy sets, with the membership functions μ_C and μ_P, respectively. Following Lacroix and Lavency [5], Yager [8] and Bordogna and Pasi [9], we model the interaction between the required and desired condition using the "and possibly" operator casting the whole bipolar query condition in the following form:

$$C \text{ and possibly } P \tag{1}$$

which may be exemplified, referring to the previous example, by "cheap and possibly comfortable".

The "possibility" referred to in the "and possibly" operator is meant here as the *consistency*[1] Namely, in the case of crisp conditions C and P it is assumed that if there is a tuple which satisfies both conditions, then and only then it is actually *possible* to satisfy both of them and each tuple has to meet both of them. In such a case (C, P) reduces to the usual conjunction $C \wedge P$. On the other hand, if there is no such a tuple, then it is *not possible* to satisfy both conditions and the desired one can be disregarded. In such a case (C, P) reduces to C alone. If C and/or P are fuzzy, both conditions may be simultaneously satisfied to some degree lower than 1. Then, the matching degree of the (C, P) query against a tuple is between its matching degrees of $C \wedge P$ and C.

Formally, for the crisp case [5]:

$$C(t) \textit{ and possibly } P(t) \equiv C(t) \wedge (\exists s(C(s) \wedge P(s)) \Rightarrow P(t)) \tag{2}$$

and for the fuzzy case [12,4]:

$$\text{truth}(C(t) \textit{ and possibly } P(t)) = \min(\mu_C(t), \max(1 - \max_{s \in R} \min(\mu_C(s), \mu_P(s)), \mu_P(t))) \tag{3}$$

where R denotes the whole dataset (relation) queried. The value of

$$\max_{s \in R} \min(\mu_C(s), \mu_P(s)), \text{ denoted as } \exists CP \tag{4}$$

which expresses the truth of $\exists s(C(s) \wedge P(s))$, may be treated as a measure of the interference of P with C.

The formula (3) is derived from (2) using the classic fuzzy interpretation of the logical connectives via the maximum and minimum operators. In Zadrożny and Kacprzyk [4] we analyze the properties of the counterparts of (3) obtained by using a broader class of the operators modeling logical connectives.

3 New Type of Bipolar Queries

Our starting point are bipolar queries in the sense of the required/desired semantics described in the previous section. According to (1) the query "C and possibly P" is satisfied (to a high degree) by a tuple t only if either of the two conditions holds:

1. it satisfies (to a high degree) both conditions C and P, or
2. it satisfies C and there is no tuple in the whole database which satisfies both conditions. (5)

[1] For other interpretations of the "and possibly" operator cf., e.g. [10,11].

Here we will discuss the concept of another "and possibly" operator based on the weakening of the condition (5). Namely, the notion of the "possibility" of satisfying both C and P will now be meant in a certain *context* of a given tuple.

Let us consider an example. We would like to prepare recommendations for the customers concerning hotels in particular regions of interest and to list cheap and comfortable hotels. But we want all regions covered and if there are no cheap and comfortable hotels in a given region, then a cheap hotel will do.

This implies the concept of a *contextual bipolar query*:

> Find *cheap* and possibly – *with respect to the hotels located in the same region* – *comfortable* hotels. (6)

meant to be satisfied by a hotel if:

1. it is cheap (to a high degree) and is comfortable (to a high degree), or
2. it is cheap (to a high degree) and there is no another hotel located in the same region which is both cheap and comfortable.

The new "contextual and possibly" operator may be formalized as follows. The context is identified here with a part of the database defined by an additional binary predicate W, i.e.,

$$Context(t) = \{s \in R : W(t, s)\} \tag{7}$$

where R denotes the whole database (relation). The "contextual and possibly" operator is no more a binary operator as it admits three arguments:

$$C \text{ and possibly } P \text{ with respect to } W \tag{8}$$

where predicates C and P should be interpreted as representing the required and desired condition, respectively, while the predicate W denotes the context.

Then, the formula (8) is interpreted as:

$$C(t) \textit{ and possibly } P(t) \text{ with respect to } W \equiv C(t) \wedge \exists s(W(t, s) \wedge C(s) \wedge P(s)) \Rightarrow P(t) \tag{9}$$

Referring to our example, C and P represent the properties of being cheap and comfortable, respectively, while W denotes the relation of being located in the same region, i.e., $W(t, s)$ is true if both tuples represent hotels located in the same region. Predicate W may be fuzzy as, e.g., if the location of a hotel represented by a tuple t is Southern Texas, then another hotel located in the (fuzzy) border of the Southern and Northern Texas, and represented by a tuple s, may be in relation W (being located in the same region) to a degree, i.e., $W(t, s) \in [0, 1)$. Below, we further discuss some possible interpretations of W.

A Partition/Similarity Based Interpretation. A relation expressed with the predicate W may define a partition of the set of tuples. In our running example concerning a hotel database, the hotels may be grouped according to their location. Such a grouping may be crisp as, e.g., in the case when the hotel location is identified with its city address. Then, the relation represented by W should be an equivalence relation, i.e., a reflexive, symmetric and transitive relation; the equivalence classes of it correspond to particular cities "present" in the database of hotels. However, such a grouping may be also fuzzy as, e.g., when the hotel location is identified with the region such as the Southern Texas or the Northern Texas. The belongingness to a given region may be not clear-cut and a hotel may be seen as located in two (or even more) neighbouring regions, with a different degree to each of them.

Most often, it will be not practical, if at all possible, for the user to directly define the relation W, i.e., to explicitly assign a membership degree for each pair of tuples. A reasonable approach, illustrated by the examples of the above paragraph, will be to define such a relation on a domain of a given attribute, obviously, for a domain of a small cardinality. For example, if we identify the address of a hotel with the city and the number of cities represented in our hotel database is reasonably small, then the user may directly define any relation on the set of the cities and adopt it to the set of the tuples in an obvious way:

$$W(t,s) \equiv P(t(A_i), s(A_i)) \tag{10}$$

where A_i is one of the attributes belonging to the scheme of the database[2] R, $t(A_i)$ denotes the value of attribute A_i for the tuple t (and, analogously $s(A_i)$), and P is a relation defined on $D(A_i)$, i.e., on the domain of the attribute A_i. Obviously, the relation W may also be defined in terms of a set of relations on a number of domains as well as in terms of a relation defined on a Cartesian product of the domains of several attributes. The latter scenario is less practical as the cardinality of the set resulting from the Cartesian product grows quickly.

A more practical scenario, in particular for "quantitative" numerical attributes, is to use an expression referring to the values of this attribute to define a *similarity* between the tuples. For example, if we wish to group the hotels into classes corresponding to various levels of an average room price, then the relation W can be defined as:

$$W(t,s) = f(\mid t(\texttt{AvgRoomPrice}) - s(\texttt{AvgRoomProced}) \mid)$$

where `AvgRoomPrice` is an attribute representing the average room price and f is a function which relates the difference of the prices of the rooms to the membership of both tuples to the same price class/level.

Actually, in some cases a more natural way to define the relation W may be via a direct definition of a partition. Such a partition is easily definable in

[2] We often identify a database with a relation which what does not restrict the generality of our consideration but makes it possible to avoid potential ambiguity of which relation is actually referred to: that being a part of a relational database or that defined on the set of the tuples of such a relation and denoted by the predicate W.

terms of the values of a selected attribute. In case of our hotels database, such a partition may be based on the value of the attribute `State`, making it possible to group the hotels according to the US state in which they are located. In the case of a crisp relation W, due to the well known one-to-one correspondence between partitions of a given set and equivalence relations defined on the same set, the user may define W in a way more comfortable to him or her, i.e., either to specify a relation or partition. In the fuzzy case the situation is a bit more complex but following Schmechel [13] it is also possible, under some conditions, to preserve the validity of such a correspondence. Namely [13], let us assume that a *fuzzy equivalence relation*, or $\tau-$*equivalence relation* is a fuzzy relation $W : R \times R \rightarrow [0, 1]$ satisfying the following conditions:

reflexivity $W(t, t) = 1, \qquad \forall t \in R,$
symmetry $W(t, s) = W(s, t), \qquad \forall t, s \in R,$
τ-transitivity $\tau(W(t, s), W(s, u)) \leq W(t, u) \qquad \forall t, s, u \in R.$

where τ is a t-norm. Moreover, let us assume that a subset $\mathcal{P} \subseteq \mathcal{F}(R)$ of the space of all fuzzy sets defined on R is a *fuzzy partition*, or a τ-partition of R if it satisfies the following conditions:

1. $\forall_{X \in \mathcal{P}} \exists_{t \in R} \mu_X(t) = 1,$
2. $\forall_{t \in R} \exists_{X \in \mathcal{P}} \mu_X(t) = 1,$
3. $\forall_{X \in \mathcal{P}} \forall_{Y \in \mathcal{P}} \forall_{t \in R} \forall_{s \in R} \mu_X(t) = 1 \Rightarrow \tau(\mu_Y(t), \mu_Y(s)) \leq \mu_X(s).$

Then, there is a one-to-one correspondence between fuzzy equivalence relations on R and fuzzy partitions of R. As the assumed properties of fuzzy equivalence relations and fuzzy partitions are fairly intuitive, such a correspondence provides for the same freedom to express the relation W as in the crisp case.

An Ordering Based Interpretation. Above we showed the usefulness of W when it is an equivalence relation. Consider now the following example.

> Find *a spacious* and possibly – *with respect to the rooms not much more expensive* – *located on a low floor* room.

Thus, of interest is now a room which:

1. is spacious (to a high degree) and located on a low floor (a fuzzy predicate "low" should be satisfied to a high degree), or
2. is spacious (to a high degree) and there is no another room not much more expensive which is both spacious and located on a low floor.

The relation W now may be expressed as $W = \neg Z$ where Z represents the concept of "much more expensive", i.e., is a (fuzzy) order relation.

Similarly interesting queries may be built referring to a temporal relation between the data searched. For example:

> Find a hotel in *Southern Texas* and possibly – *with respect to its previous record for this year* – *offering low prices.*

assuming a database storing information on the historic prices offered by hotels in different regions. Thus, of interest is now a room which:

1. is spacious (to a high degree) and located on a low floor (a fuzzy predicate "low" should be satisfied to a high degree), or
2. is spacious (to a high degree) and there is no another room not much more expensive which is both spacious and located on a low floor.

A Modal View. The predicate W has an immediate interpretation when our new type of a bipolar query is considered from the modal logic perspective [14]. Namely, let us adopt the language of the standard propositional modal logic [15] with the modality symbol $\Diamond$. Let the Kripke frame (Ω, W) be defined as:

- the set of worlds Ω comprises tuples of the relation T, i.e., for each $t \in T$ we have an $\omega_t \in \Omega$,
- W is now the accessibility relation defined on $\Omega \times \Omega$.

Then, we have the following equivalence:

$$C(t) \textit{ and possibly } P(t) \text{ with respect to } W \equiv C \wedge (\Diamond(C \wedge P) \Rightarrow P) \tag{11}$$

Thus, the satisfaction (degree) of the query is identified with the truth (degree) of the modal formula on the right hand side of (11). The latter formula is true (to a degree) at a world ω_t (for a tuple t) if and only if the formula representing the required condition C is true at the world ω_t (tuple t satisfies C) and, moreover, if the conjunction of the formulas representing C and P is true at a world ω_s accessible from ω_t, then also the formula representing the desired condition P have to be satisfied at the world ω_t.

The modal view of W provides for an obvious generalization of the interpretations of W as discussed earlier.

4 Concluding Remarks

In this paper we proposed a novel concept of a contextual bipolar query involving a ternary condition. It makes it possible to form queries with a combination of positive and negative conditions, aggregated using the "and possibly" operator in which the possibility is meant in a separately defined context. We discuss various possible interpretations of such a context.

The proposed concept of an extended bipolar query needs further research, notably related to other relevant approaches, in particular cast in the framework of *queries with preferences* in the sense of Chomicki [16,17]. There is also a certain similarity to the skyline operator based queries which are also covered by the general approach of Chomicki. Another important aspect requiring a further study is an effective and efficient execution of such queries. This will be the subject of our intensive further research. A starting point may be again the approach of Chomicki who studies several algorithms suitable for executing queries with preferences depending on the structure of the preference relation.

Acknowledgment. Mateusz Dziedzic contribution is supported by the Foundation for Polish Science under International PhD Projects in Intelligent Computing. Project financed from The European Union within the Innovative Economy Operational Programme (2007-2013) and European Regional Development Fund.

References

1. Kacprzyk, J., Zadrożny, S., Ziółkowski, A.: FQUERY III+: a "human consistent" database querying system based on fuzzy logic with linguistic quantifiers. Information Systems 14(6), 443–453 (1989)
2. Dubois, D., Prade, H.: Bipolarity in flexible querying. In: Andreasen, T., Motro, A., Christiansen, H., Larsen, H.L. (eds.) FQAS 2002. LNCS (LNAI), vol. 2522, pp. 174–182. Springer, Heidelberg (2002)
3. Zadrożny, S., Kacprzyk, J.: Bipolar queries and queries with preferences. In: Proc. 17th Int. Conf. on Datab. and Exp. Syst. Appl., DEXA 2006, pp. 415–419 (2006)
4. Zadrożny, S., Kacprzyk, J.: Bipolar queries: An aggregation operator focused perspective. Fuzzy Sets and Systems 196, 69–81 (2012)
5. Lacroix, M., Lavency, P.: Preferences: Putting more knowledge into queries. In: Proc. of the 13 Int. Conf. on Very Large Databases, pp. 217–225 (1987)
6. Dubois, D., Prade, H.: Handling bipolar queries in fuzzy information processing. In: Galindo, J. (ed.) Handbook of Research on Fuzzy Information Processing in Databases. Information Science Reference, New York, USA, pp. 97–114 (2008)
7. Matthé, T., De Tré, G., Zadrożny, S., Kacprzyk, J., Bronselaer, A.: Bipolar database querying using bipolar satisfaction degrees. Int. J. Intell. Syst. 26(10), 890–910 (2011)
8. Yager, R.: Higher structures in multi-criteria decision making. International Journal of Man-Machine Studies 36, 553–570 (1992)
9. Bordogna, G., Pasi, G.: Linguistic aggregation operators of selection criteria in fuzzy information retrieval. Int. J. of Intell. Systems 10(2), 233–248 (1995)
10. Lietard, L., Tamani, N., Rocacher, D.: Fuzzy bipolar conditions of type "or else". In: FUZZ-IEEE, pp. 2546–2551. IEEE (2011)
11. Dujmović, J.: Partial absorption function. Journal of the University of Belgrade, EE Dept. 659, 156–163 (1979)
12. Zadrożny, S.: Bipolar queries revisited. In: Torra, V., Narukawa, Y., Miyamoto, S. (eds.) MDAI 2005. LNCS (LNAI), vol. 3558, pp. 387–398. Springer, Heidelberg (2005)
13. Schmechel, N.: On lattice-isomorphism between fuzzy equivalence relations and fuzzy partitions. In: ISMVL, pp. 146–151 (1995)
14. Kacprzyk, J., Zadrożny, S.: Bipolar queries, and intention and preference modeling: synergy and cross-fertilization. In: Proc. World Conf. on Soft Computing (2011)
15. Blackburn, P., Van Benthem, J., Wolter, F. (eds.): Handbook of Modal Logic. Elsevier (2007)
16. Chomicki, J.: Querying with intrinsic preferences. In: Jensen, C.S., Jeffery, K., Pokorný, J., Šaltenis, S., Bertino, E., Böhm, K., Jarke, M. (eds.) EDBT 2002. LNCS, vol. 2287, pp. 34–51. Springer, Heidelberg (2002)
17. Chomicki, J.: Preference formulas in relational queries. ACM Transactions on Database Systems 28(4), 427–466 (2003)

Landing of a Quadcopter on a Mobile Base Using Fuzzy Logic

Patrick J. Benavidez, Josue Lambert, Aldo Jaimes, and Mo Jamshidi

Department of Electrical and Computer Engineering
University of Texas at San Antonio
San Antonio, USA
{patrick.benavidez,aldo.jaimes}@gmail.com,
josueilambert@yahoo.com, moj@wacong.org

Abstract. In this paper, we present control systems for an unmanned aerial vehicle (UAV) which provides aerial support for an unmanned ground vehicle (UGV). The UGV acts as a mobile launching pad for the UAV. The UAV provides additional environmental image feedback to the UGV. Our UAV of choice is a Parrot ArDrone 2.0 quadcopter, a small four rotored aerial vehicle, picked for its agile flight and video feedback capabilities. This paper presents design and simulation of fuzzy logic controllers for performing landing, hovering, and altitude control. Image processing and Mamdani-type inference are used for converting sensor input into control signals used to control the UAV.

1 Introduction

1.1 Background on Quadrotors

Quadcopters are a class of four-rotored aerial vehicles. They have been shown to provide stable acrobatic flight as demonstrated in many hobbyist, research and commercial grade products. To provide the flight characteristics that quadcopters are prized for, large quantities of energy must be consumed for each of the four high-speed motors. For many small quadcopters, the battery life is typically limited to minutes of flight time due to the weight of the batteries and power draw of the motors. Limited time of use for quadrotors creates a problem for researchers to solve.

1.2 Targeted Landing and Landing on a Mobile Base

Researchers are trying to find ways to improve the effectiveness of quadcopter given current battery systems and motors. One method many researchers are trying is limiting the flight time of quadrotors and providing a base station to act as either a landing pad for battery conservation [1], hot swapping [2,3] or battery recharging purposes [4]. In [2], researchers created a fixed mechanical base station for a quadcopter to proceed with a hot-swap of a battery. In [1] researchers performed landing maneuvers on a mobile base using a low-cost

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_41,

camera sensor to sense infrared Light Emitting Diodes (LEDs) acting as beacons on the mobile base.

Researchers from the University of Waterloo, Canada simulated coordinated landing of a quadcopter using nonlinear control methods. With their methods, they designed a joint decentralized controller that attracts the two linearized systems together via coupled state information [3].

In [4] researchers developed a system where an Adept Mobile Robotics P3AT unmanned ground vehicle (UGV) provided services for mobile landing and target identification for visual inspection by an ArDrone quadcopter [4]. For control of the quadcopter, the researchers utilized classical controllers for controlling the altitude, pitch, roll and yaw. Vision input was used by the researchers on both platforms for navigation and landing control. Actuators onboard the UGV platform performed error correction post-landing by shifting the ArDrone to the optimal landing position.

1.3 Paper Topic and Structure

In this paper we utilize fuzzy logic controllers to control heading, altitude, approach, and hovering for an ArDrone 2.0 using visual and distance feedback. We draw inspiration from the automated UGV/UAV inspection system in [4] for testing our system. Visual feedback is provided by identification of visual tags, similar to the ones used in [4], using open source software to calculate the tag identication match and orientation. Compared to [4], we utilize fuzzy logic controllers instead of classical controllers to simplify the design of quadcopter manuevers. Also, we use a single multi-resolution visual tag that changes by distance, instead of using multiple spatially separated tags. We also do not rely on more than the coordinate systems relative to the cameras. This is a major difference from the work of others that have constructed a fully mapped and controlled test environment to provide high resolution location services. The rest of the paper is organized as follows: Section 2 details the control problems handled by the fuzzy controllers. Section 3 details the software and hardware experimentation with the controllers. Section 4 provides results from hardware and software tests. Section 5 provides conclusions on the system and future work with the UAV/UGV system. The rest of the paper is organized as follows: Section 2 details the control problems handled by the fuzzy controllers. Section 3 details the software and hardware experimentation with the controllers. Section 5 provides results from hardware and software tests. Section 6 provides conclusions on the system and future work with the UAV/UGV system.

2 Control System

2.1 Control Problems

Figure 1 shows a depiction of the system controller. Figure 2 shows a depiction of the control problems.

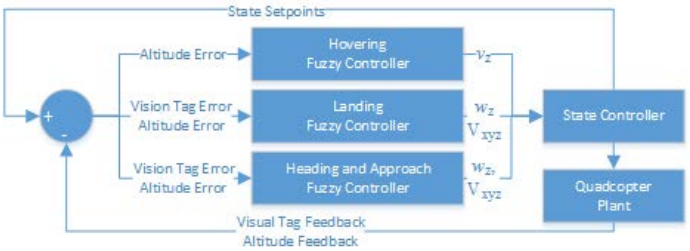


Fig. 1. Depiction of UAV System Controller

(a) Heading Control (b) Hovering Control (c) Landing Control

Fig. 2. Depiction of UAV/UGV Landing Control Problems

2.2 Altitude Control

The altitude control problem is to control the quadcopter to reach and maintain a set altitude with minimal deviation from the setpoint. To do so with a quadcopter, one needs to vary the power provided to all four rotors to produce the necessary lift to rise to the setpoint or downward force to reduce altitude.

2.3 Heading and Landing Control

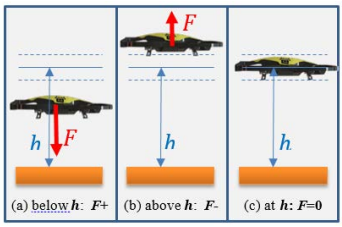
The heading control problem is to control the quadcopter to reach a desired orientation angle with the mobile landing base. To do this, the quadcopter needs to translational velocity along its local y-axis, while rotating along its local z-axis to reduce the orientation angle. Overall, the goal is to have the same orientation on both the UAV and UGV. With the same orientation, the UAV can then approach the UGV for landing operations. To approach the UGV once oriented, the quadcopter must translate along its x-axis to get close enough to use its bottom cameras to detect the landing pad. Once detected, the landing control is used to lower the craft. The landing control problem is to control the quadcopter to maintain a mininum orientation angle and positional error on descent towards a visual marker. To do this, the quadcopter needs to translate across both the x and y axes to reduce positional error, rotate along the z-axis to reduce orientation error and while descending along the z-axis to land.

2.4 Controller Variables

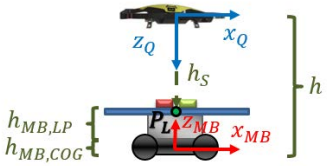
We define the following variables for the controllers:

- $h_{MB,LP}$ - height of the landing pad
- $h_{MB,COG}$ - height of the landing pad
- h_S - sensed height
- h - flying height of quadcopter
- (x_q, y_q, z_q) - pose relative to the quadcopter local frame
- (x_{MB}, y_{MB}, z_{MB}) - pose relative to the mobile landing pad local frame
- P_L - Landing position as a 3D point

Figure 3 shows the usage of the control variables in the system.



(a) Altitude control



(b) Landing control

Fig. 3. Control variable usage

3 Simulation and Hardware Experimentation

3.1 Software Packages

To develop the fuzzy controllers, we used a combination of Robot Operating System (ROS) [5], ROS Gazebo [6] with the TUM ArDrone Simulator [7], ROS package ar_track_alvar [8], and tools from FuzzyLite [9]. ROS is a software package that allows for the transport of sensor and control data via "topics" using the publisher/subscriber message passing model. The TUM ArDrone Simulator is a package for ROS Gazebo that allows for simulation of the ArDrone in 3D environments. FuzzyLite is an open source Fuzzy Logic Controller library, written in C++, which has a Graphical User Interface (GUI), called QtFuzzyLite, for designing fuzzy logic controllers. The ROS package al_track_alvar is used for unique tag identification.

3.2 Detailed Controller Design Workflow

Listed below is the workflow of the design of the fuzzy logic controller using both software and hardware:

1. Modify membership functions and/or rules in FuzzyLite GUI
2. Export fuzzy logic controller C++ code using FuzzyLite GUI
3. Insert generated code into simulated controller in ROS and compile
4. Run controller ROS node with TUM simulator
5. Repeat 1-4 if controller is not ready for hardware test, or go to step 5
6. Run controller ROS node with ArDrone autonomy drivers and ArDrone hardware
7. Repeat 1-6 if hardware test of controller exhibits unwanted behavior

3.3 ArDrone Controller and Driver ROS Nodes

The following ROS topics are inputs to 'ardrone_autonomy' which are used to control both the simulated and hardware ArDrone:

- $/ardrone/navdata/altd$ – estimated altitude in millimeters
- $/cmd\_vel/twist/linear/x$ – controls movement along local x-axis
- $/cmd\_vel/twist/linear/y$ – controls movement along local y-axis
- $/cmd\_vel/twist/linear/z$ – controls movement along local z-axis
- $/cmd\_vel/twist/angular/z$ – controls movement along local z-axis

The following ROS topics are tag recognition inputs to the fuzzy controllers from the package 'al_track_alvar':

- $/visualization\_marker/id$ - unique marker id linked to known tag
- $/visualization\_marker/pose/position/x$ - position of tag along x-axis relative to center
- $/visualization\_marker/pose/position/y$ - Position of tag along y-axis relative to center
- $/visualization\_marker/pose/position/z$ - Position of tag along z-axis relative to center of reference tag image. This is also $h - h_{MB,COG}$ when positioned above the tag
- $/visualization\_marker/pose/orientation/z$ - Quaternion angle around z-axis used for orientation

The following fuzzy system state controller actions are performed by the ROS package 'ace_ardrone_fuzzy':

1. If not directed to search for landing vehicle, perform mission
2. If tag not found, search for tag
3. If tag found, apply fuzzy alignment of quadcopter with mobile base
4. If tag found, quadcopter aligned and landing directed, apply fuzzy landing controller

4 Simulation Results

4.1 Fuzzy Inference Systems

For altitude control, we used the altitude input variable h_S as input to the controller. Figure 4 shows the fuzzy controller designed for altitude control during evaluation in FuzzyLite. The three input fuzzy controller designed for landing control is shown below in Figure 5 during evaluation in FuzzyLite.

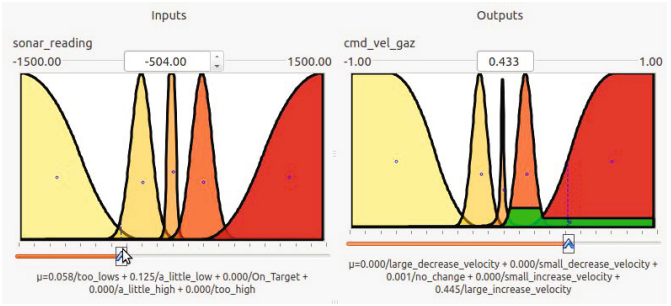


Fig. 4. Altitude controller fuzzy inference system

Rules for the altitude controller are as follows:

- if sonar_reading is too_low then cmd_vel_gaz is large_increase_velocity
- if sonar_reading is a_little_low then cmd_vel_gaz is small_increase_velocity
- if sonar_reading is On_Target then cmd_vel_gaz is no_change
- if sonar_reading is a_little_high then cmd_vel_gaz is small_decrease_velocity
- if sonar_reading is too_high then cmd_vel_gaz is large_decrease_velocity

Rules for the landing controller are as follows:

- if orientation_x is way_left then cmd_vel_z_rot is large_turn_right
- if orientation_x is a_little_left then cmd_vel_z_rot is small_turn_right
- if orientation_x is On_Target then cmd_vel_z_rot is hold_direction
- if orientation_x is a_little_right then cmd_vel_z_rot is small_turn_left
- if orientation_x is way_right then cmd_vel_z_rot is large_turn_left
- if displacement_x is way_left then cmd_vel_y_linear is large_move_right
- if displacement_x is too_left then cmd_vel_y_linear is small_move_right
- if displacement_x is centered then cmd_vel_y_linear is do_nothing
- if displacement_x is too_right then cmd_vel_y_linear is small_move_left
- if displacement_x is way_right then cmd_vel_y_linear is large_move_left
- if displacement_y is way_low then cmd_vel_x_linear is large_move_backward
- if displacement_y is too_low then cmd_vel_x_linear is small_move_backward
- if displacement_y is centered then cmd_vel_x_linear is do_nothing
- if displacement_y is too_high then cmd_vel_x_linear is small_move_forward

– if displacement_y is way_high then cmd_vel_x_linear is large_move_forward
– if displacement_z is way_high then cmd_vel_z_linear is large_move_down
– if displacement_z is too_high then cmd_vel_z_linear is small_move_down
– if displacement_z is centered then cmd_vel_z_linear is do_nothing
– if displacement_z is too_low then cmd_vel_z_linear is small_move_up

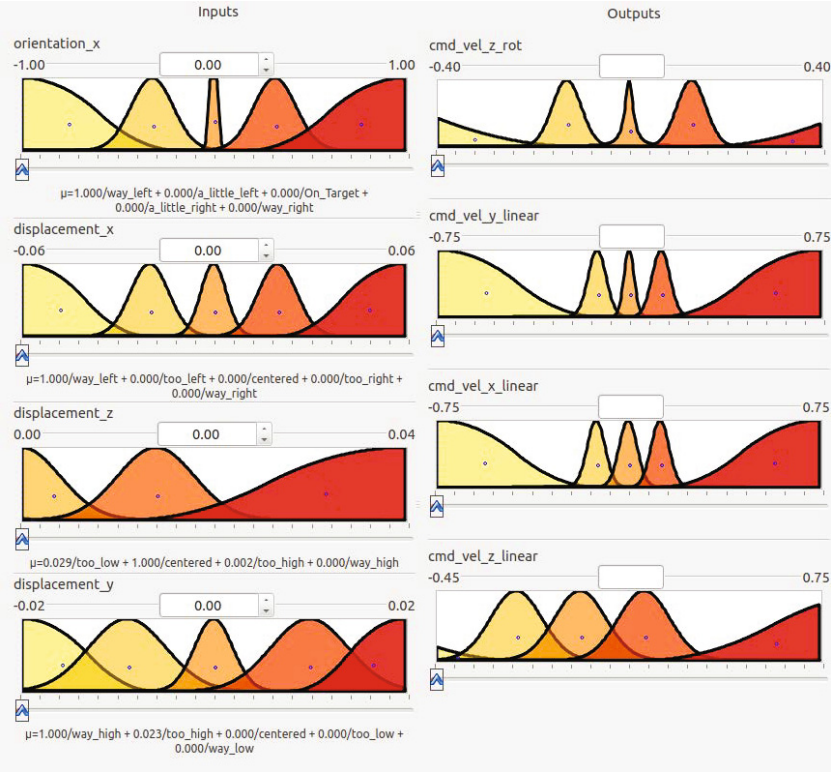


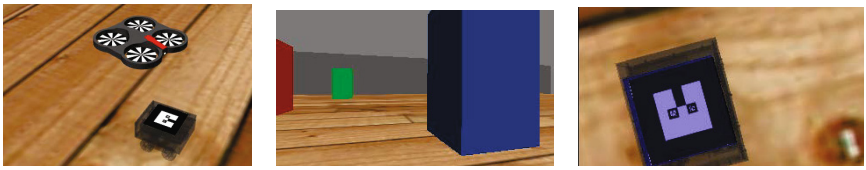
Fig. 5. Landing controller fuzzy inference system

4.2 Simulated Hover

A series of images illustrating the altitude control for the quadcopter is shown below in Figure 6.

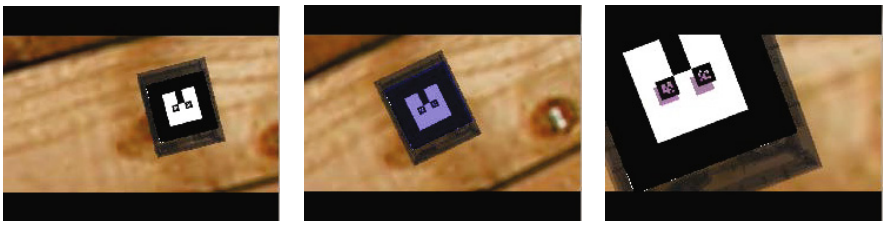
4.3 Simulated Descent

A series of images covering descent of the simulated quadcopter is shown below in Figure 7.



(a) Hovering over mobile base (b) Front camera view (c) Found large tag

Fig. 6. Hovering images from simulator before descent



(a) Tag not yet recognized (b) Found large tag (c) Found small tag

Fig. 7. Landing images from simulator during descent

5 Hardware Results

Hardware testing of the drone controllers demonstrated acceptable performance given that the environment was not too complex. Environmental complexity became an issue when the ArDrone exhibited unwanted behavior in its sensed altitude sensor during altitude control due to a suspected firmware issue. When the sonar passes too close to an object, the onboard firmware appears to re-calibrate the altitude, then re-obtains a valid sonar reading and proceeds to fly erratically using poorly calibrated altitude data. A video with the hardware results will be uploaded to Youtube.

6 Conclusion

In our work we designed fuzzy controllers for controlling altitude and hovering in place above a target. We demonstrated simulation of the controllers performed using the combination of open source tools FuzzyLite, ROS Gazebo with TUM ArDrone Simulator. Simulation results showed acceptable performance from the altitude control and also landing controller. Future work toward the open source community will include providing a solution to the corrupted altitude readings. Future work on development of the system is to integrate the controllers together via a fuzzy state machine.

References

1. Wenzel, K.E., Masselli, A., Zell, A.: Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle. Journal of Intelligent & Robotic Systems 61, 221–238 (2011)
2. Toksoz, T., Redding, J., Michini, M., Michini, B., How, J.P., Vavrina, M., et al.: Automated Battery Swap and Recharge to Enable Persistent UAV Missions. In: AIAA Infotech@ Aerospace Conference (2011)
3. Daly, J.M., Yan, M., Waslander, S.L.: Coordinated landing of a quadrotor on a skid-steered ground vehicle in the presence of time delays. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4961–4966 (2011)
4. Saska, M., Krajnik, T., Pfeucil, L.: Cooperative microUAV-UGV autonomous indoor surveillance. In: 2012 9th International Multi-Conference on Systems, Signals and Devices (SSD), pp. 1–6 (2012)
5. WillowGarage. Documentation - Robot Operating System (October 2012), `http://www.ros.org/wiki/`
6. Koenig, A.H.N.: Gazebo ROS Simulator - gazebo - ROS Wiki (2013), `http://www.ros.org/wiki/gazebo`
7. Engel, J.: TUM ArDrone Simulator - tum_ardrone - ROS Wiki (2013), `http://www.ros.org/wiki/tum_ardrone`
8. Niekum, S.: AR Tag Tracking Library - ar_track_alvar - ROS Wiki (2013), `http://www.ros.org/wiki/ar_track_alvar`
9. Rada-Vilela, J.: fuzzylite - A Fuzzy Logic Control Library and Application in C++ (2013), `http://code.google.com/p/fuzzylite/`

An Innovative Process for Qualitative Group Decision Making Employing Fuzzy-Neural Decision Analyzer

Ki-Young Song[1], Gerald T.G. Seniuk[2], and Madan M. Gupta[3]

[1] Department of Mechanical Engineering
The University of Tokyo
Tokyo, Japan
sky8071@gmail.com
[2] College of Law
University of Saskatchewan
Saskatoon, Canada
seniuk@sasktel.net
[3] Intelligent Systems Research Laboratory
College of Engineering
University of Saskatchewan
Saskatoon, Canada
Madan.gupta@usask.ca

Abstract. Many qualitative group decisions in professional fields such as law, engineering, economics, psychology, and medicine that appear to be crisp and certain are in reality shrouded in fuzziness as a result of uncertain environments and the nature of human cognition within which the group decisions are made. In this paper we introduce an innovative approach to group decision making in uncertain situations by using fuzzy theory and a mean-variance neural approach. The key idea of this proposed approach is to defuzzify the fuzziness of the evaluation values from a group, compute the excluded-mean of individual evaluations and weight it by applying a variance influence function (VIF); this process of weighting the excluded-mean by VIF provides an improved result in the group decision making.

1 Introduction

In any professional field such as law, engineering, economics, psychology, and medicine, we are often faced with ambiguous choices in our decision making processes. A decision making process involves perceptions relating to neurological processes of acquiring and mentally interpreting qualitative (fuzzy) information. Our cognitive process relating to the process of acquiring knowledge by the use of reasoning, intuition, and perception for the evaluation of suitable courses of action (decisions) among several alternative choices plays a very important part in human decision making [1]. Such decisions are made where human judgment is called upon to choose among competing and viable options. These decisions are

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_42,

often made under the environments of vague (fuzzy), incomplete and conflicting evidence. Thus, our decision making process is often masked by fuzzy and statistical uncertainties. A change in the formulation of these cognitive processes, for example by providing new information, can alter the results of our decision significantly.

In recent decades, a number of methods have been developed to improve the quality of decisions by applying some new mathematical methods using computers. Fuzzy logic has been one of the major candidates as a mathematical tool in decision making processes. One application of fuzzy logic in decision making is to employ fuzzy linguistic quantifiers which quantify linguistic expressions such as 'good', 'fair', and 'bad' [2, 3]. Some other ways of using fuzzy logic are to define uncertainties with the possibility of events for strategic decisions [4]. More approaches of fuzzy logic on group decisions can be found in the reference [5].

In this paper, we present a novel approach for qualitative group decision making namely a fuzzy-neural decision analyzer (FNDA) incorporating some feedback mechanism that allows various outcomes to be assessed and evaluated. Here, we introduce a new notion of excluded-mean and excluded-variance. This proposed FNDA consists of the following three stages:

(i) Quantifier (an interface between human and computer) For computational purposes, the quantification of the perceived subjective opinion of individual evaluators on each piece of evidence is converted into some fuzzy numbers, say over the interval $[-10, 10]$;

(ii) Pre-processor First, as a defuzzification process, fuzzy logic is applied to the perceived subjective opinions of evaluators expressed in the form of fuzzy numbers. Then, the two statistical parameters, excluded-mean ($\boldsymbol{\mu}$) and excluded-variance ($\boldsymbol{v}$), of the defuzzified numbers are computed. Finally, the neural input $\boldsymbol{u}$ for the neural decision processor is computed which is the product of $\boldsymbol{\mu}$ and a variance influence function (VIF, $\boldsymbol{f}$);

(iii) Neural decision processor (NDP) The NDP uses $\boldsymbol{u}$, the output of the pre-processor, under various classifications of evidences, and finally yields a decision $\boldsymbol{y}$.

In this paper, we evaluate the performance of the proposed fuzzy-neural decision analyzer (FNDA) on a generic case study.

2 Design of Fuzzy-Neural Decision Analyzer (FNDA)

The group decision making analyzer in a qualitative (fuzzy) environment presented in this paper is based upon the confluence of fuzzy-statistical and neural approaches [6]. For the formulation of this novel group decision making approach in a qualitative environment, we introduce (a) the expertise derived from an evaluation group to provide the defuzzification and (b) three new statistical parameters: excluded-mean ($\boldsymbol{\mu}$), excluded-variance ($\boldsymbol{v}$), and variance influencing function (VIF, $\boldsymbol{f}$). It should be emphasized that without the incorporation of VIF, the decision making will be plagued by the large variance resulting from

the undue influence of the unscreened subjective ranking tacitly present in any qualitative decision making process.

The procedure in the fuzzy-neural decision analyzer (FNDA) for a group decision making, as shown in Fig. 1, is as follows: First, each subjective (qualitative) opinion is quantified into a fuzzy number (evaluation), say s. The pre-processor defuzzyfies the fuzzy numbers depending on the expertise of the group, then computes excluded-mean ($\boldsymbol{\mu}$) and excluded-variance ($\boldsymbol{v}$) for each evidence/factor using the fuzzy data provided by n evaluators. Then, a decision is made by a neural decision processor (NDP).

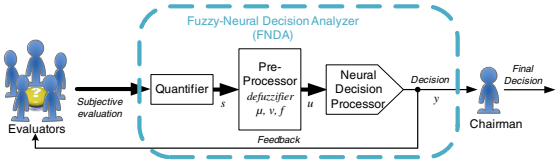


Fig. 1. Flow chart of fuzzy-neural decision analyzer (FNDA)

2.1 Quantifier

In this process, qualitative language is translated to quantitative language. In a group decision making process on a given case, each member of the group expresses his/her opinion (evaluation), this opinion is influenced by individual perception, which, in turn, is influenced by his/her experience. The evaluation employs fuzzy linguistic quantifiers which quantify linguistic expressions such as 'good', 'fair', and 'bad' [2, 3]. In order to express the opinion (evaluation) of each evaluator, the qualitative (fuzzy) opinions are quantified into some fuzzy numbers, say over the interval of $[-10, 10]$. The fuzziness and statistical (random fluctuations) uncertainty can be managed in this pre-processor.

Let us consider a situation with a group of n evaluators and m pieces of evidence (factors). The individual pieces of evidence (factors) are the features of a case to be evaluated, and the evaluators of the group are asked to quantify their opinions for each piece of evidence with a fuzzy number (value) over an interval, $[-10, 10]$. In the evaluation process, each evaluator (j ($j = 1, 2, \ldots, n$)) is asked to quantify his/her opinion on each evidence (factor, i ($i = 1, 2, \ldots, m$)) using a fuzzy number s_{ij} over the interval. In some cases, some p evaluators do not wish to mark a score and instead mark 'X' for some evidence (factor, i) for any of the following reasons:

i) the evaluator feels that this particular evidence is not relevant for this decision; and/or
ii) the evaluator considers himself/herself unqualified to judge this particular evidence appropriately.

2.2 Pre-processor: Defuzzifier

In this process, the fuzziness of the quantified numbers is supervised by a defuzzifier applying fuzzy theory introduced by Zadeh [7]. Fuzzy sets describe the fuzzy number whose boundaries are not precise. By expressing the fuzzy numbers, a fuzzy set shows a mapping γ from a fuzzy number s ($s \in \boldsymbol{S}, \boldsymbol{S}$ is the universe of discourse) to the unit interval [0, 1], which converts non-crisp numbers to crisp numbers. This idea can be represented as a fuzzy set through a triangular membership function with the slope (a) and the degree of fuzziness (uncertainty, Δs) of the evaluation. The operations on two fuzzy sets, A and B, are defined as

$$A \cup B : \gamma_{A \cup B} \equiv \max[\gamma_A(s), \gamma_B(s)], s \in \boldsymbol{S} \tag{1}$$

$$A \cap B : \gamma_{A \cap B} \equiv \min[\gamma_A(s), \gamma_B(s)], s \in \boldsymbol{S} \tag{2}$$

$$-A : \gamma_{-A} \equiv 1 - \gamma_{-A}(s), s \in \boldsymbol{S} \tag{3}$$

The process of defuzzification is performed by Fuzzy Logic Toolbox Software (MATLAB 2010a, The MathWorks) and briefly explained in this paper. The details can be found in the reference [8].

In this study, fuzzy sets (input function) for fuzzy numbers (input) are defined with triangular membership functions considering some uncertainty (Δs). Output functions are also decided with triangular membership functions. Then, fuzzy rules (if-then rule) are constructed between an input function (IF) and an output function (OF) as an IF number corresponds to the same OF number, such as IF1→OF1, IF2→OF2, and so forth. Thus, a mapping between inputs (fuzzy values, s) and outputs (defuzzified values, $\hat{s}$) can be constructed by applying a 'centroid' defuzzification method. The centroid defuzzification method yields a defuzzified value ($\hat{s}$) by retuning the center of area under the output function curve [8].

In the decision making process of an evaluator group, we should take into consideration how much the evaluator group is familiar with the case to be decided. Motivated by this consideration, we generate a fuzzy set with some uncertainty (Δs) according to the expertise of an evaluator group (uneducated, educated and expert) on the particular case. The uneducated group (G_{UE}) represents a group has little experience with the type of case, and the members are not familiar with the particulars of the case. The educated group (G_E) has experience with the case or been educated about it, thus the members are reasonably familiar with the case. The members in the expert group (G_{EX}) are very familiar with the case and the underlying issues, thus the evaluation of a member in this group includes very minimal uncertainty. The number of fuzzy sets in a case is also determined by the expertise of the evaluator group. For G_{UE}, three fuzzy sets (input functions) are constructed with high uncertainty ($\Delta s = 7$), and accordingly three output functions are applied. As a result, the mapping presents a higher oscillation which represents high uncertainty in this group. For G_E, seven input functions and output functions are built with $\Delta s = 2.5$. The mapping contains small oscillation indicating lower uncertainty in this group. The fuzzy sets for G_{EX} are made with 21 input functions and output functions

with $\Delta s = 1$. The mapping of input and output functions is almost linear, which signifies almost no uncertainty in this group. The mapping functions in each group are illustrated in Fig. 2. During the defuzzification process, in case of 'X' ($s = X$), the defuzzified value of the evaluation is also kept as 'X' ($\hat{s} = X$).

2.3 Pre-processor: Excluded-Mean, Excluded-Variance and Variance Influence Function

After defuzzifying the fuzzy values, the pre-processor of FNDA computes statistical values (excluded-mean, $\boldsymbol{\mu}$, and excluded-variance, $\boldsymbol{v}$) for each piece of evidence (factor) using the evaluation of the group evaluators. In the computation of mean and variance, the number of p opinions marked 'X's are excluded. Thus, the excluded-mean (μ_i) and the excluded-variance (v_i) are computed as

$$Excluded - Mean : \mu_i = \left(\frac{1}{n-p}\right)\left[\sum_{j=1}^{n} \hat{s}_{ij}\right], \ for\ \hat{s}_{ij} \neq X \tag{4}$$

$$Excluded - Variance : v_i = \left(\frac{1}{n-p}\right)\left[\sum_{j=1}^{n} (\hat{s}_{ij} - \mu_j)^2\right], \ for\ \hat{s}_{ij} \neq X \tag{5}$$

The excluded-mean of opinions indicates the average opinions of the group, whereas the extended-variance in the scores indicates the degree of inconsistency (disagreement or fluctuations) of the opinions of the evaluators. Then, the excluded-mean is further weighted by a function of excluded-variance, a variance influence function (VIF) that can deal the statistical fluctuations in group opinions. One can formulate many types of VIF. One of the VIFs that we use in this paper is defined as

$$f(\alpha v_i) = \exp(-\alpha v_i), \ (i = 1, 2, \ldots, m) \tag{6}$$

where $\exp(-\alpha v_i)$ is an exponential function, and α is the gain that provides the importance to VIF. It should be noted that a large difference of opinions will yield a large excluded-variance v_i. As excluded-variance v_i increases, the

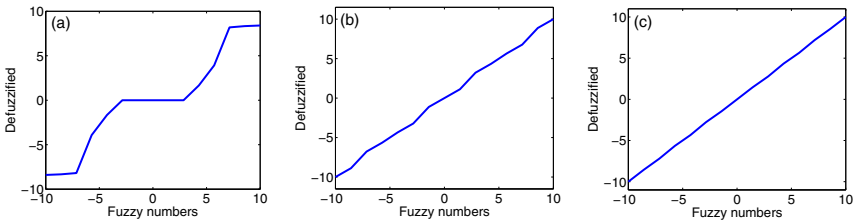


Fig. 2. Mapping functions for (a) G_{UE}, (b) G_E and (c) G_{EX}. The mapping functions are generated from different numbers of input and output functions as well as different uncertainties (Δs)

significance of the evidence $f(\alpha v_i)$ decreases. Also, as the gain α changes, the significance of the relationship changes. In a group decision making process, it is commonly observed that some evaluations are very close (low variance), whereas some evaluations are very diverse (high variance). It is also to be noted that the diversity (fluctuation) of evaluations for certain evidence arises due to the different experiences, subjectivity and cognitions of individual evaluators. In order to assess the agreement of evaluations for each piece of evidence, it is necessary that some weight be assigned to each piece of evidence in the evaluation process. This basic principle of variance influence function (VIF) is expressed as follows:

- *If the evaluations for a piece of evidence are in general agreement, then the excluded-variance is relatively low, and, thereby, the influence of that evidence should be high.*
- *If the evaluations for a piece of evidence are not in general agreement, then the excluded-variance is relatively high, and, thereby, the influence of that evidence should be low.*

2.4 Neural Decision Processor (NDP)

In general, a neural network (NN) is composed of many neural layers, and each neural layer has many neural units. NN is one of the most powerful tools for classification. It provides a superior procedure for identification and classification and has therefore been used widely in this type of research [9]. A sigmoidal function is commonly applied for a mapping function due to its special characteristics exhibiting a progression from a small beginning to an accelerated end as natural processes [10]. Considering the special features of conventional NN, we propose a neural decision processor (NDP) for group decision making as shown in Fig. 3. The NDP presented in this paper is composed of two neural layers. The first and the second neural layers are named category layer and decision layer, respectively [11, 12]. In NN, a neural unit is composed of two operations: the synaptic operation and the somatic operation. The synaptic operation is the sum of products of neural inputs ($\boldsymbol{u}$) and neural weights ($\boldsymbol{w}$) which represent the past experiences. The somatic operation is a nonlinear mapping process.

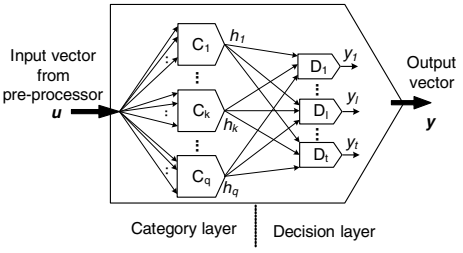


Fig. 3. Neural Decision Processor (NDP) for a group decision making process. This NDP is composed of a category layer and a decision layer.

The input to the NDP is the output vector from the pre-processor. In the NDP, *the category layer* is influenced by the process of group decision making. In most cases, the evidence identified as relevant will have some common features. Therefore, the pieces of evidence that have common features are grouped into a category. In this study, we define a category layer which classifies evidence in q categories $(C_1, C_2, \ldots, C_q)$. Furthermore, each such evidence is weighted by different values (weight, ${}^{c}w_{kb}$, k is the category number, and b is the weight number in C_k) in different categories. For example, evidence F1 can be classified into category C1 as well as in C3; however, the corresponding weights of F1 in C_1 and C_3 may be different due to the degree of importance of F1 in different categories. The weights (${}^{c}\boldsymbol{w}$) of the neural units in the category layer are pre-assigned by unanimous group assessments. The number of neural units in the category layer is determined by the number of categories of evidence/factors. Finally, *the decision layer* of the NDP accumulates the information ($\boldsymbol{h}$) from each neural unit in the category layer for final decisions. In this layer, the importance of the categories is ranked by assigning different weights. Then the individual decisions are made $(D_1, D_2, \ldots, D_t)$. The neural weights (${}^{d}w_{lb}$, l is a decision number, and b is the weight number in D_l) in this decision layer are also pre-assigned by the members of a group.

The fuzzy-neural decision analyzer (FNDA) proposed in this paper is an analytical tool to aid in a group decision making process. This is a generic mathematical decision analysis model that can be applied in a variety of real life decision making processes in a qualitative language (fuzzy) environment.

3 A Case Study: New Product Development

In industries, it is necessary to determine if a new product can be competitive and/or successful in the market before deciding a new product line. Accurate decision making on the new product development becomes more important, and committee members evaluate the idea of the new product under the confluence of various pieces of evidence (factors). Recently, many applications of fuzzy set theory employing evaluations and survey analysis have been used in industrial engineering research [13–15]. We apply FNDA to a group decision for the development of a new product with 50 evaluators and 14 factors (characteristics of the product) on the new product as presented in Ref.[16]. The survey (benchmark) was carried out using these 50 evaluators, and their evaluations are expressed by scores over the interval $[-10, 10]$. 'X' in the table (Ref.[16]) implies 'irrelevance', and it indicates 'no evaluation' which may occur during the evaluation of factors as discussed earlier. The statistical parameters, the excluded-mean, the excluded-variance and the VIF, are further calculated in each group, G_{UE}, G_E and G_{EX}, after the mapping functions for defuzzification as presented in Table 1.

For this particular case study having 14 factors, we classified the factors in four categories ($q = 4$) to make one decision ($t = 1$) as follows:

Table 1. The statistical parameters of the three different groups after defuzzfication

Group	Statistical parameters	Evidences (Factors) F1	F2	F3	F4	F5	F6	F7
G_{UE}	μ	0.91	−1.08	1.68	0	−1.53	−0.39	−1.69
	v	3.14	5.16	8.1	0	5.39	2.08	9.62
	$f(\times 10^{-3})$	43.3	5.7	0.3	1000	4.5	124.6	0.07
G_E	μ	1.96	−2.18	2.63	0.02	−2.91	−0.83	−2.3
	v	8.03	6.94	9.95	1.99	7.58	8.04	13.27
	$f(\times 10^{-3})$	0.33	0.97	0.05	136.7	0.51	0.32	0
G_{EX}	μ	1.93	−2.26	2.61	0.05	−2.9	−0.86	−2.37
	v	7.96	6.82	9.97	2.27	7.75	8.13	12.77
	$f(\times 10^{-3})$	0.35	1.1	0.05	103.3	0.43	0.3	0

Group	Statistical parameters	Evidences (Factors) F8	F9	F10	F11	F12	F13	F14
G_{UE}	μ	3.88	−0.23	0	0.09	1.08	2.72	2.72
	v	13.34	0.71	0	3.55	9.21	11.34	10.84
	$f(\times 10^{-3})$	0	493.5	1000	28.8	0.1	0.01	0.02
G_E	μ	5.19	−1.07	0.5	−0.41	2.04	3.53	3.77
	v	10.4	5.27	0.91	10.02	14.86	11.52	9.18
	$f(\times 10^{-3})$	0.03	5.1	400.9	0.04	0	0.01	0.1
G_{EX}	μ	5.14	−1.15	0.62	−0.37	1.98	3.55	3.82
	v	10.59	5.26	1.37	9.91	14.7	11.55	9.21
	$f(\times 10^{-3})$	0.03	5.2	255.1	0.05	0	0.01	0.1

- *Category 1 (C_1): F2, F5, F6, F7, F8, F9 and F11*
- *Category 2 (C_2): F2, F3, F5, F6, F7, F9, F11 and F12*
- *Category 3 (C_3): F2, F5, F6, F7, F9, F11 and F13*
- *Category 4 (C_4): F1, F4, F10 and F14*

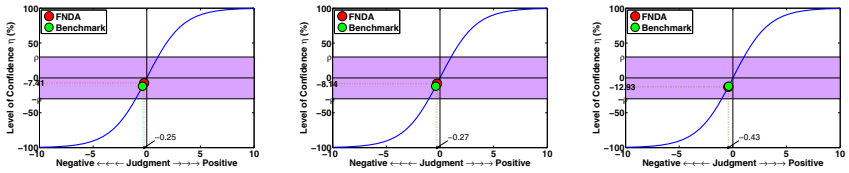
It should be noted that some of the factors are common to various categories. For example, F2 is included in categories C_1 and C_2 as well as in C_3. For the category layer, in the synaptic operation, we assumed an equal weight, ${}^{c}w_{kb} = 1$ ($k = 1 \sim 4$, $b = 1$ number of factors in C_k), for all the factors in each neural unit with the threshold ${}^{c}w_{k0} = 0$ ($k = 1 \sim 4$). In the somatic operation, a linear mapping function with gain g=1 was assigned. For the decision layer, in the synaptic operation, we assigned the weights as ${}^{d}w = [15, 10, 5, 0.75]$ with the threshold ${}^{d}w_0 = 0$. In the somatic operation, the mapping function $\Phi(gz) = 100tanh(gz)$ with gain $g = 0.3$ yields an output over the interval [−100, 100], where a negative value represents the 'Negative' decision with a confidence level between [−100 %, 0 %), where a positive value implies the 'Positive' decision with a confidence level between (0 %, 100 %], and where $\Phi = 0$ implies a neutral decision. In our decision making process, to achieve some meaningful results, we define the confidence zone over the interval $\pm\rho$ %, and in this study, we set $\rho = 30$.

3.1 Validation of FNDA with Group Decision

In order to validate our proposed model, we compare the benchmark group decision and the results of FNDA. In the group decision making, an individual member of the group provides his/her decision in the form of 'YES' or 'NO' as presented in Ref.[16]. If the numbers of 'YES' are Y, and the numbers of 'NO' are N, then we define the confidence level η as

$$\eta = \frac{Y - N}{Y + N} \times 100(\%) \tag{7}$$

A positive η implies a positive decision (YES), and a negative represents a negative decision (NO). In this benchmark group survey with 50 evaluators for the proposed new production line, a Negative decision was made with $\eta = -12\%$. This 'Negative' decision with low confidence (-12%) is not within the significant level. By the validation model of FNDA presented in this study, if the results from FNDA are similar to that of the benchmark group decision, then, we can safely rely on our proposed FNDA configuration. For the FNDA process, first, in the pre-processing, we computed μ, v and f with gain $\alpha = 1$ without any defuzzification process in order to validate the proposed algorithm. Following this, the outputs of the pre-processor were fed to the neural decision processor (NDP) with assigned weights ($^c\boldsymbol{w}$ and $^d\boldsymbol{w}$) in the category and the decision layers. After the decision making process of the FNDA, the decision value (judgment) becomes -0.39 which implies that the FNDA reflects a 'Negative' decision, and accordingly the confidence level is yielded as -11.6% which is very close to the decision of the benchmark group (-12%). From the validation, it can be concluded that with the assigned parameter values such as neural weights, category numbers, and gains are dependable, and the FNDA is safely reliable for further analysis of decision making on the new production line. Now, we consider the groups expertise with the obtained evaluation scores. By doing so, the decision of each group case can be further investigated. In the cases of G_{EU}, G_E and G_{EX}, the decisions by the FNDA are different as shown in Fig. 4. The decisions of the three groups are all 'Negative' judgment, and the levels of confidence in G_{UE}, G_E, and G_{EX} are -7.41%, -8.14%, and -12.93%, respectively. Although the levels of confidence of the three groups are different, their decisions are not too divergent and all lie in lower confidence zone. With the assigned parameters,



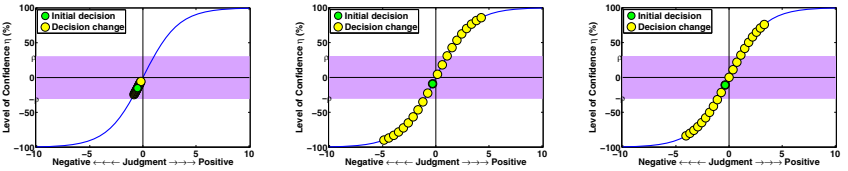
(a) G_{UE} (b) G_E (c) G_{EX}

Fig. 4. The decisions by FNDA after defuzzification. The shaded area represents lower confidence zone.

the decision and level of confidence of the expert group case yields better agreement with those of the benchmark group because there is small uncertainty in the evaluation values in the expert group.

3.2 Fact-Finding Process of Individual Evaluator by FNDA: A Feedback Process in Decision Making

After validating the FNDA with the benchmark group decision, all of the factors could be ranked by the values of neural weights in the category and the decision layers. The process of making such rankings in itself would aid the decision maker in his/her thinking. "Did the evaluator place too much emphasis on that factor?" is a question that would then be open to meaningful review, either by others or individuals alone. The exploration of an evaluation is called the *fact-finding process* which is a feedback (review) process in decision making. We apply the FNDA to review the decision of an individual evaluator. In this review process, the decisions of individual evaluators are examined with feedback created by varying their evaluations on certain factors. The results of the pre-processing shown in Table 1 shows that F4 and F10 give the lowest values of v, thereby, the highest values of f in all of group cases, which indicates that the opinions about these two factors were the most dominant influences on the final decision. Here, as an example, we explore E1's evaluation in each group by changing the evaluation values of F1 and F10 in the fact-finding process. Before changing the evaluation values, the level of confidence in the decision of E1 is found as -15% in G_{UE}, -8% in G_E and -10% in G_{EX}. Then, we alter the evaluation value of F1 from -10 to 10 for each group, and it is discovered that the level of confidence on the decision of E1 does not vary much. The confidence levels of G_{UE}, G_E, and G_{EX} are in the interval of $[-15.3\%, -14.5\%]$, $[-8.4\%, -8.2\%]$, and $[-10.2\%, -9.9\%]$, respectively. The negative value of the confidence level indicates 'Negative' decision. This indicates that the decisions of E1 are mostly the same regardless of E1's evaluations of F1. Similarly, the factors with lower VIF value have little effect on the individual decision process. However, after altering the evaluation value of F10 from -10 to 10 for each group, we can see that the level of confidence in the decision of E1 is much diverged as shown in Fig. 5. The green circle is the decision before the changing the evaluation of E1. The yellow (bigger) circles represent new decision after new evaluation. The yellow (bigger) circles are widely distributed through the level of confidence from negative and positive in the cases of the educated group and the expert group. In the case of the uneducated group, the decision of E1 is not quite diverged; however, some decisions are almost close to the confidence zone. Likewise, higher values of VIF play significant roles in the decision of the new production line. This observation indicates that the decision changes by other evaluators in a group may result into a similar wide distribution over the level of confidence. From the observation of this simulation study, we conclude that the expertise of a group is an important feature to be considered during the decision making process by a group. The expertise affects the defuzzification process significantly. This simulation also revealed that the decision of an uneducated group may not



(a) G_{UE}: [−24.4%, −5.0%] (b) G_E: [−90.8%, 87.4%] (c) G_{EX}: [−84.9%, 78.3%]

Fig. 5. Decision (level of confidence) change of evaluator #1 (E1) by altering the evaluation value of evidence (factor) #10 (F10) from −10 to 10 in each group. The shaded area represents lower confidence zone.

change much by varying the evaluation scores of factors; however, in an educated and an expert group, a slight change of some evaluation of a factor may yield very different decisions.

4 Conclusions

In this paper, we have introduced the fuzzy-neural decision analyzer (FNDA), a new approach for quantitative group decision making. One of the key ideas of this proposed approach is to defuzzify the evaluation values considering the expertise of the evaluation group as uneducated, educated or expert. Therefore, more accuracy can be achieved during the review processes. Another key idea is to update the output of the pre-processor by applying a variance influence function (VIF) which emphasizes the importance of each piece of evidence (factor), thereby reducing the statistical uncertainty that occurs in the process of transforming qualitative expressions to quantitative scoring. This improves the evaluations in group decision making processes. Further, FNDA is applied to an individual fact-finding process as a review process using the outcomes from the group decision, which represents a feedback process during an individual decision making process. A case study of a group decision making of a 'YES' or 'NO' variety on a new product development was carried out to demonstrate the application of FNDA. The advantage of applying FNDA in a process of group decision making is that it allows for more precisiation [17, 18], without losing the fuzzy richness of the reality under consideration.

In summary, FNDA can assist decision makers by this process of weighting and ranking the evidence (factor) they rely on. FNDA is a useful tool to be applied for decision making in qualitative language environments such as business or law or public policy.

References

1. Simon, H.A., Dantzig, G.B., Hogarth, R., Plott, C.R., Raiffa, H., Schelling, T.C., Shepsle, K.A., Thaler, R., Tversky, A., Winter, S.: Decision Making and Problem Solving. Interfaces 17(5), 11–31 (1987)

2. Kacprzyk, J.: Group decision making with a fuzzy linguistic majority. Fuzzy Sets and Systems 18(2), 105–118 (1986)
3. Kacprzyk, J., Fedrizzi, M., Nurmi, H.: Group decision making and consensus under fuzzy preferences and fuzzy majority. Fuzzy Sets and Systems 49(1), 21–31 (1992)
4. Bykzkan, G., Feyoglu, O.: A fuzzy-logic-based decision-making approach for new product development. International Journal of Production Economics 90(1), 27–45 (2004)
5. Fedrizzi, M., Pasi, G., Da Ruan, D.R., Hardeman, F., van der Meer, K.: Fuzzy Logic Approaches to Consensus Modelling in Group Decision Making. Studies in Computational Intelligence 117, 19–37 (2008)
6. Gupta, M.M., Jin, L., Homma, N.: Static and Dynamic Neural Networks: from Fundamentals to Advanced Theory. Wiley (2003)
7. Zadeh, L.A.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)
8. MathWorks: Documentation: Fuzzy Logic Toolbox (2012), `http://www.mathworks.com/help/toolbox/fuzzy/index.html?s_cid=BB`
9. Hou, Z.-G., Song, K.-Y., Gupta, M.M., Tan, M.: Neural Units with Higher-Order Synaptic Operations for Robotic Image Processing Applications. Soft Computing - A Fusion of Foundations, Methodologies and Applications 11(3), 221–228 (2007)
10. Bukovsky, I., Hou, Z.-G., Bila, J., Gupta, M.M.: Foundations of nonconventional neural units and their classification. International Journal of Cognitive Informatics and Natural Intelligence (IJCINI) 2(4), 29–43 (2008)
11. Gupta, M.M., Seniuk, G.T.G., Kozinski, J.: Fuzzy Logic, Law and Evidence-based Decision Making. In: The 3rd International Conference on Data Management (ICDM 2010) (2010)
12. Gupta, M.M., Seniuk, G.T.G., Kozinski, J.: Neural and Fuzzy Logic Approach to Decision-Making (Common Law Trials - as an Example). In: The World Conference on Soft Computing 2011 (WConSC 2011), San Francisco, CA, USA (2011)
13. Lin, L., Lee, H.-M.: Fuzzy Assessment Method on Sampling Survey Analysis. Expert Systems with Applications 36(3, pt. 2), 5955–5961 (2009)
14. Ngan, S.-C.: Decision Making with Extended Fuzzy Linguistic Computing, with Applications to New Product Development and Survey Analysis. Expert Systems with Applications 38(11), 14052–14059 (2011)
15. Wang, W.-P.: Evaluating New Product Development Performance by Fuzzy Linguistic Computing. Expert Systems with Applications 36(6), 9759–9766 (2009)
16. Song, K.-Y., Kozinski, J., Seniuk, G.T.G., Gupta, M.M.: Excluded-Mean-Variance Neural Decision Analyzer for Qualitative Group Decision Making. Advances in Fuzzy Systems 2012(2012), 1–20 (2012)
17. Zadeh, L.A.: The Concept of a Linguistic Variable and its Application to Approximate Reasoning - II. Information Sciences 8(3), 199–249 (1975)
18. Zadeh, L.A.: A Computational Approach to Fuzzy Auantifiers in Natural Languages. Computers & Mathematics with Applications 9(1), 149–184 (1983)

Preprocessing Method for Support Vector Machines Based on Center of Mass

Saied Tadayon and Bijan Tadayon

Z Advanced Computing, Inc.
Maryland, USA
{saiedtadayon,bijantadayon}@ZAdvancedComputing.com
www.ZAdvancedComputing.com

Abstract. We present an iterative preprocessing approach for training a support vector machine for a large dataset, based on balancing the center of mass of input data within a variable margin about the hyperplane. At each iteration, the input data is projected on the hyperplane, and the imbalance of the center of mass for different classes within a variable margin is used to update the direction of the hyperplane within the feature space. The approach provides an estimate for the margin and the regularization constant. In the case of fuzzy membership of the data, the membership function of the input data is used to determine center of mass and to count data points which violate the margin. An extension of this approach to non-linear SVM is suggested based on dimension estimation of the feature space represented via a set of orthonormal feature vectors.

Keywords: SVM, Center of Mass.

1 Introduction

1.1 SVM Training for Large Dataset: Background

Support vector machines (SVMs) have been shown to be power tools for classification of input data based on structural risk minimization [1-2]. SVM uses a hyperplane (within the input space, in case of linear SVM, or in a feature space, in case of non-linear SVM) to separate the input data based on their classification while maximizing the margin from the input data. In case of inseparable dataset, a soft margin version of SVM is used to allow for misclassification error by imposing a penalty, e.g., proportional with the Euclidian distance from the class margin. In such a case, a regularization parameter is used as a tradeoff mechanism between the maximizing the margin and minimizing the error penalty. The appropriate level of tradeoff is generally determined by a validation step to estimate the out-of-sample error.

N number of samples $(\mathbf{x}_i, y_i)$ are used for training an SVM, where $\mathbf{x}_i \in \mathcal{R}^d$ and $y_i \in \{-1,1\}$ (denoting the classification of the i^{th} data sample). A hyperplane classifier is sought to separate the input data according to their classification:

M. Jamshidi et al. (eds.), *Advance Trends in Soft Computing WCSC 2013*,
Studies in Fuzziness and Soft Computing 312,
DOI: 10.1007/978-3-319-03674-8_43,

$$f(\mathbf{x}) = sign(u_i) = sign(\mathbf{w} \cdot \mathbf{x}_i + b) = \begin{cases} +1, & \text{if } y_i = +1 \\ -1, & \text{if } y_i = -1 \end{cases} \tag{1}$$

Direction of $\mathbf{w}$ is perpendicular to the hyperplane, and its inverse of magnitude represents the margin between the hyperplane (having plane number 0) and the margin surface having the plane number u_i set to +1 or -1 (e.g., at the nearest class data points in a linearly separable case):

$$y_i\, u_i = y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1,\ \forall i \tag{2}$$

When the dataset is not linearly separable, a slack (or error) parameter is used to still classify the data point correctly within the slack from the class margin:

$$y_i\, u_i = y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 - \xi_i,\ \xi_i \geq 0, \forall i \tag{3}$$

To maximize the margin $\|\mathbf{w}\|^{-1}$, an objective function is formed to minimize $(\mathbf{w} \cdot \mathbf{w})$ as well as the slack errors, based on the trade off parameter C, subject to (3):

$$\text{Minimize } \left(\frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{N} \xi_i\right) \tag{4}$$

The solution may be found at the saddle point of the Lagrangian:

$$\mathcal{L}(\mathbf{w}, b, \alpha_i, \beta_i) = \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 + \xi_i) - \sum_{i=1}^{N} \beta_i\, \xi_i \tag{5}$$

where $\alpha_i, \beta_i \geq 0$, and the Lagrangian is minimized w.r.t. $(\mathbf{w}, b)$ and maximized w.r.t. (α_i, β_i), yielding:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{6}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{7}$$

$$\alpha_i + \beta_i = C,\ \ 0 \leq \alpha_i, \beta_i \leq C \tag{8}$$

The modified Lagrangian in dual form (i.e., by substituting (6) and using (7) and (8)) is quadratic in α_i, and it is minimized w.r.t. α_i, subject to constraint (8):

$$\mathcal{L}'(\boldsymbol{\alpha}) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

In a non-linear case, where the optimization is done in a feature space $\mathcal{Z}$, the vector product in (9) would become $(\mathbf{z}_i \cdot \mathbf{z}_j)$, where $\mathbf{z}_i = \phi(\mathbf{x}_i)$ is the corresponding feature vector and the dot product in space $\mathcal{Z}$ may be expressed as a corresponding kernel $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ in $\mathcal{X}$ domain, satisfying Mercer condition. The solution for (9) provides a set of $\{\alpha_i\}$ where most are typically zeros, i.e., corresponding to data points that are outside the margin (with zero slack). A non-zero α_i represents an $\mathbf{x}_i$ which is at the

margin or violating the margin with a non-zero slack. The KKT conditions for the solution are [3]:

$$\begin{cases} y_i\, u_i > 1 & \Leftrightarrow & \alpha_i = 0 \\ y_i\, u_i = 1 & \Leftrightarrow & 0 \le \alpha_i \le C \\ y_i\, u_i < 1 & \Leftrightarrow & \alpha_i = C \end{cases} \tag{10}$$

The classification hypothesis may be expressed by the few non-zero α_i's (by substituting (6) in (1)) with their corresponding $\mathbf{x}_i$'s denoted as support vectors (SVs):

$$f(\mathbf{x}) = sign\left(\sum_{\alpha_i>0} \alpha_i y_i \mathcal{K}(\mathbf{x}_i, \mathbf{x}) + b\right) \tag{11}$$

While most α_i's are typically zero, the performance of quadratic (QD) programming solvers suffer for large datasets due to large size of $(N \times N)$ kernel matrix with $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ elements. Various approaches have been developed to address this issue, such as "chunking" to break down a larger QD problem into series of smaller ones [4], [5], and [8] or breaking the problem to the smallest chunk in pair-wise sequential minimal optimization [3]. In addition, the solver typically repeats the optimization by varying the values of C and/or kernel parameter(s) within a wide exponential range, and a grid search is used to determine the optimum hyperparameter(s) likely to minimize out of sample error (e.g., estimated by validation dataset). Other approaches attempt to eliminate subset of input dataset via fuzzy clustering to reduce the workload [6], or use a coarse clustering approach to reduce the runtime [9].

2 Preprocessing for SVM Using Center of Mass

While any data point might be a support vector (as anticipated by (9)), we present a preprocessing approach to quickly identify the potential SVs in linear SVM, as the initial starting point for QD solvers in order to speed up the optimization process. In one approach, a greedy algorithm projects and orders the data points on a direction for hyperplane, in order to find an optimum hyperplane in such a direction. Then, it takes a step in changing the direction of the hyperplane, based on the data points already scanned at or about the margin and the center of masses of margin violators. This approach also estimates the relationship between margin and C to narrow the practical range of C's needed for use with validation. An extension of this approach to non-linear SVM is suggested based on an assumption that relatively few SVs would support the hyperplane having a relatively low effective dimensionality.

2.1 Leverage Model of Lagrange Multipliers

The relations (6) and (7) suggest a view of leverage for the data points (depicted in solid black in Fig. 1) at or inside the margin about the hyperplane. Per (10), α for the points inside the margin gets limited to C, while those at the margin may have α between 0 and C. Furthermore, $\alpha_i y_i$ provides the polarity to the "force" α_i excreted on the margin by the corresponding data point. For example, as shown in Fig. 1, such force by "circle" class (having y = +1) is pointing down, while those from "square" class (with y = -1) are pointing in opposite direction.

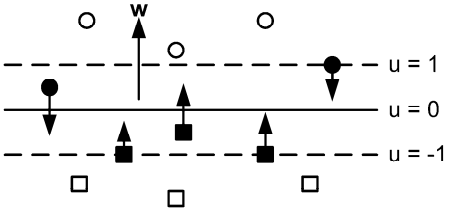


Fig. 1. Leverage model of data points at the margin

Equation (7) may be rewritten as a force balance equation:

$$\sum_{i_+} \alpha_{i_+} = \sum_{i_-} \alpha_{i_-} \quad \left(= \frac{1}{2} \sum_{i=1}^{N} \alpha_i \right) \tag{12}$$

where i_+ and i_- are indexes for non-zero α_i's corresponding to "circle" and "square" classes (i.e., y = +1 and -1), respectively. The soft margin SVM that allows errors (with slack) limits the value of α to C (tradeoff parameter). This can be interpreted as the "skin" of the soft margin only being able to support a point force up to C before getting penetrated. In other words, the force on the margin builds up from 0 to C, as the point is "pushed" from outside the margin through the "skin".

The "force" model (with α playing the role of force) can be extended to a "leverage" model (with torque) by observing that the RHS of (6) resembles a torque $\alpha_i y_i \mathbf{x}_i$ having $\mathbf{x}_i$ as its leverage. To exploit this property, we project (6) on a direction perpendicular to $\mathbf{w}$ (denoted by unit vector $\hat{\mathbf{w}}_\perp$):

$$\sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i . \hat{\boldsymbol{w}}_\perp) = \left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \right) . \hat{\mathbf{w}}_\perp = \mathbf{w} . \hat{\mathbf{w}}_\perp = 0 \tag{13}$$

Note that in $\mathcal{D}$ dimensional space of $\mathbf{w}$, there are $(\mathcal{D} - 1)$ independent $\hat{\mathbf{w}}_\perp$ per $\hat{\mathbf{w}}$ (unit vector in direction of $\mathbf{w}$, also denoted as $\hat{\mathbf{w}}_\parallel$). Equation (13) implies that, at the solution, the torque from the forces $\alpha_i y_i$ balance so not to tilt $\hat{\mathbf{w}}$ in the direction of $\hat{\mathbf{w}}_\perp$. It should be noted that if $\boldsymbol{x}_i$'s are offset by a arbitrary fixed vector $\mathbf{q}$, Equation (6) (as well as (13)) remain invariant under such translation due to (7):

$$\sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i - \mathbf{q}) = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i - \left(\sum_{i=1}^{N} \alpha_i y_i \right) \mathbf{q} = \mathbf{w} - \mathbf{0} = \mathbf{w} \tag{14}$$

By placing $\mathbf{q}$ on the hyperplane (u = 0), Equation (13) demonstrates that the torques balance around such a pivot point in any of $\hat{\mathbf{w}}_\perp$ direction(s), as for example depicted in Fig. 2.

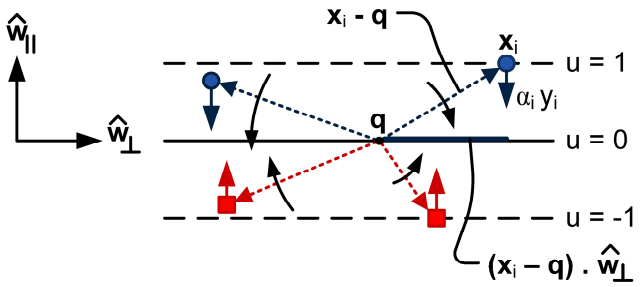


Fig. 2. Balancing torques at pivot point $\mathbf{q}$ with leverage projected on $\hat{\mathbf{w}}_{\perp}$

This also implies that the "center of mass" for "circle class" (for solid circles) should have the same projection on $\hat{\mathbf{w}}_{\perp}$ as the center of mass for "square class" (solid squares), when the optimum solution is at hand. This is because the torque from a set of points from one class can be represented by the torque from their corresponding center of mass. Note that the center of mass for such points is weighted by their corresponding force (α) as shown below:

$$\left(\sum_{i_+} \alpha_{i_+}\right) \cdot \mathbf{COM}_+ = \sum_{i_+} \alpha_{i_+} \boldsymbol{x}_{i_+}, \quad \left(\sum_{i_-} \alpha_{i_-}\right) \cdot \mathbf{COM}_- = \sum_{i_-} \alpha_{i_-} \boldsymbol{x}_{i_-} \tag{15}$$

Given (12) and (6), Equation (13) may be written as follows:

$$\mathbf{COM}_+ \cdot \hat{\boldsymbol{w}}_{\perp} = \mathbf{COM}_- \cdot \hat{\boldsymbol{w}}_{\perp} \text{ or } (\mathbf{COM}_+ - \mathbf{COM}_-) \cdot \hat{\boldsymbol{w}}_{\perp} = \mathbf{0} \tag{16}$$

In cases that the number of points violating the margin (from both classes) are significantly higher than those exactly on the margin, one can assume that most of α_i's are limited to C, and the determination of the center of mass is simplified to a class member head count (violating the margin) and its projection on $\hat{\mathbf{w}}_{\perp}$. It is noteworthy that the projection on the direction of $\mathbf{w}$ (i.e., on $\hat{\mathbf{w}}$ or $\hat{\mathbf{w}}_{||}$ produces the inverse of margin:

$$\sum_{i=1}^{N} \alpha_i y_i (\boldsymbol{x}_i . \hat{\mathbf{w}}_{||}) = \left(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i\right) . \hat{\mathbf{w}}_{||} = \mathbf{w} . \hat{\mathbf{w}}_{||} = \|\mathbf{w}\| = m^{-1} \tag{17}$$

where m is the margin. To setup the problem, for the first iteration, $\mathbf{COM}_{\pm}$ are determined from all class data points via (15) (by ignoring α_i's, e.g., by setting them to the same constant). The initial $\mathbf{w}_{init}$ is estimated as follows:

$$\hat{\mathbf{w}}_{init} = \frac{\mathbf{COM}_+ - \mathbf{COM}_-}{\|\mathbf{COM}_+ - \mathbf{COM}_-\|} \tag{18}$$

A set of $\hat{\mathbf{w}}_{\perp}$'s is determined for $\hat{\mathbf{w}}_{init}$, for example by (reconstructing) a successive pair-wise rotation matrices that define a transformation for aligning the unit vector

associated with the last coordinate to $\hat{\mathbf{w}}_{init}$. The same transformation provides a set of $(\mathcal{D}-1)$ orthonormal $\hat{\mathbf{w}}_{\perp}$'s by operating on the other unit vectors of other $(\mathcal{D}-1)$ dimensions.

2.2 Scanning through Ordered Projection Values

Next, $\boldsymbol{x}_i$'s are projected onto $\hat{\mathbf{w}}_{init}$ (or $\hat{\mathbf{w}}_{\parallel}$) (see Fig.3), and they are sorted based on the projected values (p_i):

$$p_i = \boldsymbol{x}_i . \hat{\mathbf{w}}_{\parallel} \tag{19}$$

Then, for a given set of percentage of population (e.g., 1%, 5%, 10%, 20% … of the class with less members), scan/count from low end of p_i for y = +1 class (denoted as p_{start+}) and high end of p_i for y = -1 class (denoted as p_{start-}) as shown in Fig. 3. The count can also be sequential from each sorted list, for example, based on margin or percentage change. If the data set happens to be separable by $\hat{\mathbf{w}}_{\parallel}$, p_{start+} is more than p_{start-}, and their average (denoted as p_{m0}) marks the hyperplane candidate for zero error, and half of their difference corresponds to its margin.

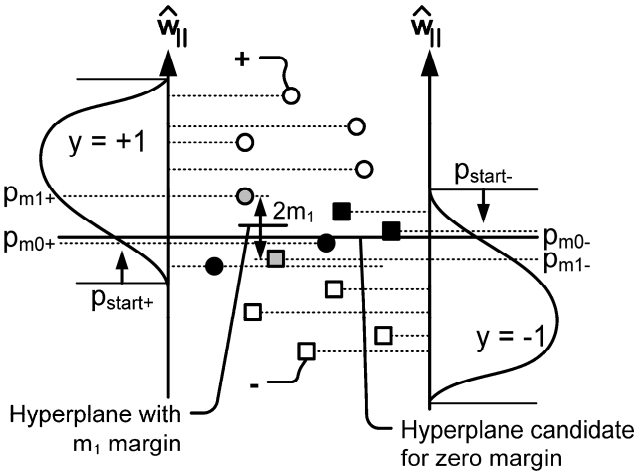


Fig. 3. Projection of $\boldsymbol{x}_i$ on $\hat{\mathbf{w}}_{\parallel}$

If p_{start+} is less than p_{start-} (i.e., not linearly separable in $\hat{\mathbf{w}}_{\parallel}$ direction), the count/scan continues until the class markers pass each other. In such a case, the prior position of the markers (before passing each other), p_{m0+} and p_{m0-}, are used to mark a zero-margin hyperplane candidate under $\hat{\mathbf{w}}_{\parallel}$, for example by taking an average value. The slack error is tracked by simply adding the projected values during the scan/count (to be later offset and scaled by the position of the hyperplane and size of the margin, respectively) as shown in (22) and (23). At given class counter positions, p_{m1+} and p_{m1-} (see Fig. 3), the hyperplane parameters are estimated as follows:

$$m = \frac{1}{2}(p_{m1+} - p_{m1-}) \text{ and } \quad \mathbf{w} = \|\mathbf{w}\|\hat{\mathbf{w}}_{\|} = m^{-1}\,\hat{\mathbf{w}}_{\|} \tag{20}$$

$$b = \left(u - \frac{\hat{\mathbf{w}}_{\|} \cdot \mathbf{x}}{m}\right)\Big|_{at\ hyperplane} = 0 - \frac{p_{m1+} + p_{m1-}}{2\,m} = -\frac{p_{m1+} + p_{m1-}}{p_{m1+} - p_{m1-}} \tag{21}$$

$$Est.Slack\ Err \approx C \cdot [\mathcal{F}_{+}(N_{m1+}) + \mathcal{F}_{-}(N_{m1-})]\,.$$

where

$$\begin{aligned}\mathcal{F}_{\pm}(N_{m1\pm}) &= N_{m1\pm} \cdot \left[\frac{1}{m}\left(p_{m1\pm} - \frac{1}{N_{m1\pm}} \sum_{\substack{i_{\pm} \\ scanned}} p_{i_{\pm}}\right) \cdot y_{\pm}\right] \\ &= N_{m1\pm} \cdot \left[\frac{1}{m}(p_{m1\pm} - \mathbf{COM}_{scanned\pm} \cdot \hat{\mathbf{w}}_{\|})\right] \cdot y_{\pm}\,.\end{aligned} \tag{22}$$

where N_{m1+} and N_{m1-} are number of class points counted/scanned corresponding to p_{m1+} and p_{m1-} projection class markers. $\mathbf{COM}_{scanned\pm}$ is the center of mass for the scanned data point for a class, and $y_{\pm}$ is ± 1, based on the class. In a synchronous counting between classes, where the counts are the same (N_{m1}) and the class errors are weighted equally, the estimated slack error is:

$$\begin{aligned}Est.Slack\ Err &\approx C \cdot N_{m1} \cdot \left[2 - \frac{1}{m \cdot N_{m1}}\left(\sum_{\substack{i_{+} \\ scanned}} p_{i_{+}} - \sum_{\substack{i_{-} \\ scanned}} p_{i_{-}}\right)\right] \\ &= C \cdot N_{m1} \cdot \left[2 - \frac{1}{m}(\mathbf{COM}_{scanned+} - \mathbf{COM}_{scanned-}) \cdot \hat{\mathbf{w}}_{\|}\right].\end{aligned} \tag{23}$$

In this scheme, during one scan/count, various error levels are estimated for a given count N_{m1} or percentage of the population (based on N_{m1}). Summations over p_i's in (22) and (23) represent a cumulative running sum as the counting progresses. Similarly, $\mathbf{COM}_{\pm}$ or their projections on $\hat{\mathbf{w}}_{\|}$ (as well as on $\hat{\mathbf{w}}_{\perp,k}$'s) are determined as running sums based on the scanned data points. Objective function (4) and its elements as well as misclassification ratio (MR) data points can be estimated and tracked:

$$Obj\ Func. \approx \frac{1}{2\,m^2} + Est.Slack\ Err \approx \frac{1}{2\,m^2} + C \cdot [\mathcal{F}_{+}(N_{m1+}) + \mathcal{F}_{-}(N_{m1-})] \tag{24}$$

$$MR \approx \frac{1}{N}\left[IndexLookup\left(\frac{p_{m1+}+p_{m1-}}{2}\right)\Big|_{+} + IndexLookupRev\left(\frac{p_{m1+}+p_{m1-}}{2}\right)\Big|_{-}\right]. \tag{25}$$

where *IndexLookup* and *IndexLookupRev* determine the number of misclassified data points in each class, by looking up the index of the projection value of hyperplane in the ordered list of the projected values of $\boldsymbol{x}_i$'s onto $\widehat{\mathbf{w}}_{\|}$.

Per (20) and (25), a relationship between margin m and in-sample misclassification rate MR is determined for each candidate $\widehat{\mathbf{w}}_{\|}$ being iterated, based on the scan through various values of $N_{m1\pm}$. In addition, $\mathcal{F}_{\pm}(N_{m1\pm})$ as a measure of slack error is made independent of C, per (22). Therefore, the relationship between the objective function (24) and C may conveniently be determined for a candidate $\widehat{\mathbf{w}}_{\|}$ without rescanning the dataset, for a given C. Thus, a range of appropriate C may be estimated, for example, as an order of magnitude below and above $(2\,m^2[\mathcal{F}_{+}(N_{m1+})+\mathcal{F}_{-}(N_{m1-})])^{-1}$ for various $N_{m1\pm}$ encountered during the same scan.

2.3 Predictive Tilting Based on Center of Mass and In/Out Adjustments

For data points on the margin boundary, (6) and (13) may be used to determine the effect of trade off $\Delta\alpha$ between two margin points $\boldsymbol{x}_1$and $\boldsymbol{x}_2$ of the same class (e.g., y = +1) with opposite $\Delta\alpha$:

$$\Delta\mathbf{w} = \Delta\left(\sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i\right) = \Delta\alpha\; y\;(\boldsymbol{x}_2 - \boldsymbol{x}_1) \tag{26}$$

Given $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are on the margin, $(\boldsymbol{x}_2 - \boldsymbol{x}_1)$ is perpendicular to $\widehat{\mathbf{w}}_{\|}$. Therefore, the effect is a tilt to $\mathbf{w}$ in the direction of $(\boldsymbol{x}_2 - \boldsymbol{x}_1)$. The amount of tilt is proportional to $\|\boldsymbol{x}_2 - \boldsymbol{x}_1\|$, i.e., the tradeoff in α for far away points has larger impact on tilting $\mathbf{w}$. The concept applies to points of different classes (having the same $\Delta\alpha$ per (12) and opposite signs for y's). Motivated by (26), we describe an efficient method for providing $\Delta\mathbf{w}$ for the subsequent iteration.

In addition to projection of data points on $\widehat{\mathbf{w}}_{\|}$, the data point(s) $\boldsymbol{x}_i$ are also projected on $\widehat{\mathbf{w}}_{\perp}$'s, and sorted accordingly (see for example Fig. 2):

$$p_{i,k} = \boldsymbol{x}_i.\,\widehat{\mathbf{w}}_{\perp,\mathrm{k}} \tag{27}$$

where k indexes over $(\mathcal{D}-1)$ independent $\widehat{\mathbf{w}}_{\perp}$'s corresponding to $\widehat{\mathbf{w}}_{\|}$. For a given $\widehat{\mathbf{w}}_{\|}$, assume the margin, m_{opt}, and offset, b_{opt}, are optimized as to (24) for a given C, per previous section. Thus, an update to $\widehat{\mathbf{w}}_{\|}$ should be in form of a rotation or a small tilt, $\Delta\mathbf{w}$, perpendicular to $\widehat{\mathbf{w}}_{\|}$ (see Fig. 4).

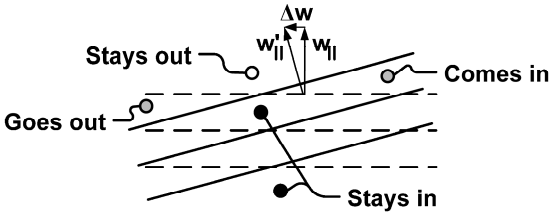


Fig. 4. Tilt in $\widehat{\mathbf{w}}_{\|}$

The small tilt maintains the magnitude of $\widehat{\mathbf{w}}_{\|}$, and therefore, the corresponding portion of the objective function $\frac{1}{2\,m^2}$ does not change. However, the tilt impacts slack errors in (4) in three ways, based on (a) points staying in violation of the margin having different slack error, (b) points going out of margin violation reducing slack error, and (c) points coming into margin and increasing slack error, as depicted in Fig. 4. Assuming the tilt pivot point, **PV**, is located on the hyperplane (i.e., $\mathrm{b} = -\boldsymbol{w} \cdot \mathbf{PV}$), the change in u_i due to $\Delta\boldsymbol{w}$ becomes:

$$\Delta u_i = \Delta\boldsymbol{w} \cdot (\mathbf{x}_i - \mathbf{PV}) \tag{28}$$

Let $\Delta\boldsymbol{w}$ be aligned to $\widehat{\mathbf{w}}_{\perp,\mathrm{k}}$, so that $\Delta\boldsymbol{w} = \|\Delta\boldsymbol{w}\|\ \widehat{\mathbf{w}}_{\perp,\mathrm{k}}$. Then:

$$\Delta u_i = \|\Delta\boldsymbol{w}\| \left(p_{i,k} - \mathbf{PV} \cdot \widehat{\mathbf{w}}_{\perp,\mathrm{k}}\right) \tag{29}$$

Let S_{in-pri} be a set of in-margin data points prior to the tilt (i.e., including those that stay in and go out after the tilt). The total change in plane numbers for each class is:

$$\begin{aligned}\sum_{i_{\pm}\in S_{in-pri}} \Delta u_{i_{\pm}} &= \|\Delta\boldsymbol{w}\| \left(\sum_{i_{\pm}\in S_{in-pri}} p_{i_{\pm},k} - N_{in-pri_{\pm}}\, \mathbf{PV} \cdot \widehat{\mathbf{w}}_{\perp,\mathrm{k}} \right) \\ &= \|\Delta\boldsymbol{w}\| \cdot N_{in-pri_{\pm}} (\mathbf{COM}_{\pm} - \mathbf{PV}) \cdot \widehat{\mathbf{w}}_{\perp,\mathrm{k}}\,. \end{aligned} \tag{30}$$

where $N_{in-pri_{\pm}}$ denotes the number of data points from each class in S_{in-pri}; $i_{\pm}$ indexes points over each class; and $\mathbf{COM}_{\pm}$ is the center of mass for each class of data point in S_{in-pri}, given by

$$\mathbf{COM}_{\pm} = \frac{1}{N_{in-pri_{\pm}}} \sum_{i_{\pm}\in S_{in-pri}} \boldsymbol{x}_{i_{\pm}}\,.$$

The change in slack error for S_{in-pri} becomes:

$$\sum_{i\in S_{in-pri}} \Delta\xi_i = \sum_{i\in S_{in-pri}} -y_i \cdot \Delta u_i = -\|\Delta\boldsymbol{w}\| \cdot N_{in-pri} (\mathbf{COM}_{+} - \mathbf{COM}_{-}) \cdot \widehat{\mathbf{w}}_{\perp,\mathrm{k}} \tag{31}$$

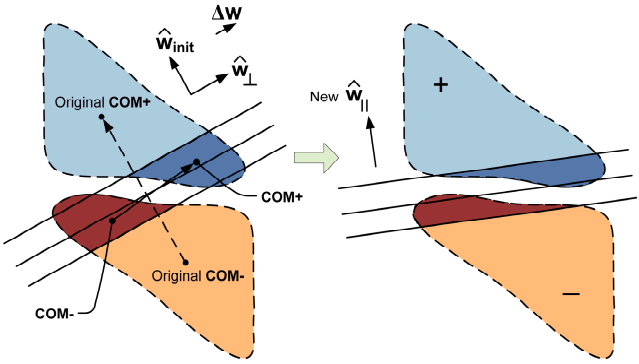


Fig. 5. Reduction of slack error by tilting $\widehat{\mathbf{w}}_{\parallel}$ based on center of masses of data points that violate the margin (shown in darker color)

where it is assumed that $N_{in-pri_{\pm}}$ are equal and denoted by N_{in-pri} (resulting in elimination of **PV**). As depicted in Fig. 5, the imbalance in $\mathbf{COM}_{\pm}$ can be used to reorient $\widehat{\mathbf{w}}_{\parallel}$ via $\Delta \boldsymbol{w}$ in reducing the slack error.

However, the expression for $\Delta\xi$ is now adjusted by those points going out and coming into the margin after the tilt by considering both u_i and Δu_i to account for double counting of those going out (denoted by $S_{out-after}$) which were included in S_{in-pri}, as well as those coming into the margin after the tilt (denoted by $S_{in-after}$).

$$\sum_{i \in S_{in-after}} \Delta\xi_i = \sum_{i \in S_{in-after}} \left(1 - y_i \cdot (u_i + \Delta u_i)\right) \tag{32}$$

Further simplification results:

$$\sum_{i \in S_{in-after}} \Delta\xi_i = N_{in-after} - \left(N_{in-after_+} - N_{in-after_-}\right)\left(\mathrm{b} - \|\Delta \boldsymbol{w}\|\, \mathbf{PV} \cdot \widehat{\mathbf{w}}_{\perp,\mathrm{k}}\right)$$
$$- \sum_{i_+ \in S_{in-after_+}} \frac{1}{m}\left(p_{i_+} + \frac{\|\Delta \boldsymbol{w}\|}{\|\boldsymbol{w}\|}\, p_{i_+,k}\right)$$
$$+ \sum_{i_- \in S_{in-after_-}} \frac{1}{m}\left(p_{i_-} + \frac{\|\Delta \boldsymbol{w}\|}{\|\boldsymbol{w}\|}\, p_{i_-,k}\right).$$

where $N_{in-after_{\pm}}$ are the number data points for each class coming into the margin after the tilt, and $N_{in-after}$ is the sum $\left(N_{in-after_+} + N_{in-after_-}\right)$. The expression for the adjustment of slack error for data points going out of the margin is quite similar, except for minus sign (as the errors are reduced) and labels switching from *in* to *out*:

$$\sum_{i\in S_{out-after}} \Delta\xi_i = \sum_{i\in S_{out-after}} \left(-1 + y_i \cdot (u_i + \Delta u_i)\right) \tag{33}$$

Therefore, at each iteration, (31) may readily be evaluated for a given $\|\Delta \boldsymbol{w}\|$, based on tracking of the running sum of $p_{i_{\pm},k}$ in (30) or center of masses in (31), during the scanning of the ordered list of projected data (19). While (31) is explicitly proportional to $\|\Delta \boldsymbol{w}\|$, (32) and (33) are only indirectly related to $\|\Delta \boldsymbol{w}\|$, through $N_{in-after}$ and $N_{out-after}$. It would be reasonable for most affected points be from regions far from the pivot point per (28) where Δu_i would be greater.

The following approach is adopted to control the tilt so that the scan of projected data along $\widehat{\mathbf{w}}_{\|}$ would provide the required data for evaluating (31), (32), and (33). Based on the scanning of p_{m1+} and p_{m1-} projection class markers, Equations (20)-(25) provide applicable hyperplane (if any) and its associated objective function and slack error for various N_{m1}'s. Therefore, around the marker positions corresponding to margin, m_{opt}, and offset, b_{opt}, other neighboring markers provide such information, e.g., for a larger margin m_2, as depicted in Fig. 6.

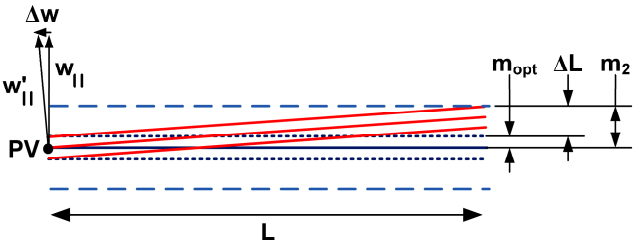


Fig. 6. Limiting the tilt, based on data obtained in projection scan along $\widehat{\mathbf{w}}_{\|}$

The markers corresponding to m_2 are adopted to effectively limit $\|\Delta \boldsymbol{w}\|$, so that the set of data points going into the margin ($S_{in-after}$) due to the tilt would be limited to those data points entering margin m_2 when margin is expanded from m_{opt} in $\widehat{\mathbf{w}}_{\|}$ direction (denoted as $S_{m_2-m_{opt}}$), provided that:

$$\frac{\|\Delta \boldsymbol{w}\|}{\|\boldsymbol{w}\|} \le \frac{\Delta L}{L} = \frac{m_2 - m_{opt}}{L} = \frac{\Delta m}{L} \tag{34}$$

where L is the full extent of data points in $\widehat{\mathbf{w}}_{\perp,\mathrm{k}}$ direction from the pivot point, and the pivot may be taken at the extremes of the range or near a center of mass (e.g., where $\mathbf{COM}_{\pm}$ project on $\widehat{\mathbf{w}}_{\perp,\mathrm{k}}$). Practically, the margin markers which control the margin boundaries are used to determine the limit for ΔL. For example,

$$\frac{\|\Delta \boldsymbol{w}\|}{\|\boldsymbol{w}\|} \le \frac{p_{m2\pm} - p_{opt\pm}}{L} \tag{35}$$

where $p_{m2\pm}$ and $p_{opt\pm}$ are the projections of the margin boundaries of m_2 and m_{opt} on $\widehat{\mathbf{w}}_{\|}$. Consequently, $S_{in-after} \subset S_{m_2-m_{opt}}$, and therefore, a search for elements of $S_{in-after}$ becomes limited to a relatively small dataset $S_{m_2-m_{opt}}$. In evaluating (34), it is first checked whether a given point in $S_{m_2-m_{opt}}$ has entered the tilted margin (e.g., via evaluating $y_i u_i$ or using (29)). Similar check is made for determining $S_{out-after}$ using (29) where the updated u_i indicates whether a point in S_{in-pri} has moved out of margin (i.e., to $S_{out-after}$).

To determine if objective function may be minimized by tilting $\boldsymbol{w}$, (31), (32), and (33) are added for a given $\|\Delta \boldsymbol{w}\|$ to check if the total slack error is reduced (denoted as a functional $\Delta\xi_{\perp,\mathrm{k}}(\|\Delta \boldsymbol{w}\|)$) for a particular $\widehat{\mathbf{w}}_{\perp,\mathrm{k}}$. If so, $\Delta\mathbf{w}$ adopts a component from $\widehat{\mathbf{w}}_{\perp,\mathrm{k}}$ in proportion to the corresponding error reduction:

$$\Delta \mathbf{w} = \eta \sum_{\forall \widehat{\mathbf{w}}_{\perp,\mathrm{k}}} \left(-\Delta\xi_{\perp,\mathrm{k}}(\|\Delta \boldsymbol{w}\|)\right) \widehat{\mathbf{w}}_{\perp,\mathrm{k}} \tag{36}$$

where η is a learning step parameter. In one approach, the contribution to $\Delta\mathbf{w}$ is made if $\Delta\xi_{\perp,\mathrm{k}}(\|\Delta \boldsymbol{w}\|) < -\epsilon$, where ϵ is a threshold parameter. In another approach, the contribution is made when $\left|\Delta\xi_{\perp,\mathrm{k}}(\|\Delta \boldsymbol{w}\|)\right| > \epsilon$, allowing for taking the step in an opposite direction of increasing slack error.

The next step of iteration is performed by updating the candidate $\mathbf{w}$ according to (36), and it stops for example when the objective function (24) does not improve significantly or a maximum iteration count has reached.

2.4 Estimation of Lagrange Multipliers

Based on the above preprocessing iterations, a set of candidate hyperplanes are generated for various values of C. According to KKT condition (10), α_i's are generated for bounded parameters (i.e., $\alpha_i = 0$ or C). For any data point at margin, initialize α_i to $C/2$, and/or use (13) to determine such α_i's based on the distribution of unbounded parameters across $\widehat{\mathbf{w}}_{\perp,\mathrm{k}}$.

An advantage of this approach is that quite immediately in its iterations, proper scale of C is readily determined.

3 Extension to Non-Linear SVM

3.1 A Case of RBF (Gaussian) Kernel

Equation (9) after kernel substitution for feature space becomes:

$$\mathcal{L}'(\boldsymbol{\alpha}) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) \tag{37}$$

In the following approach, instead of attempting to break the problem of $N \times N$ quadratic form in (9), we make an assumption that in a relatively low dimensional space

scenario, there may be significantly more points in the margin than those exactly at the margin, and we attempt to find an approximate solution in the non-linear space via a low dimensional decomposition.

To have low number of support vectors with RBF (Gaussian) kernel, the data points need to be well represented or covered in $\mathcal{R}^d$ via several kernels centered around few central points to be uniquely identifiable, e.g., through triangulation. This requirement helps to ensure that in the feature space, the dimensions represented by the central points provide dimensional coverage for other data points in the dataset. Therefore, in $\mathcal{R}^d$, we assign at least $d+1$ such centers to basically allow for triangulation. However, the requirement calls for coverage as well, so that a given point in the dataset would have non-trivial kernel values with respect to at least $d+1$ central points. RBF (proximity) kernel in input space is between (0,1], with the coverage dependent on the bandwidth parameter σ as shown in (38):

$$\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = exp\left(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right), \gamma = \frac{1}{2\sigma^2} \tag{38}$$

Therefore, the coverage of each central point extends to at most several σ's. Too small a σ creates islands out of each input dataset, resulting in high dimensionality, high number of SVs, and relatively high out of sample error [7]. In other words, the learning process memorizes the training data instead of learning the overall pattern for small σ. On the other hand, too large a σ, it will have difficulty negotiating stronger curves than the shape of the kernel allows.

Let there be l such centers in $\mathcal{R}^d$ covering the input dataset, denoted as $\mathbf{G}_j$ with $j = 1\ to\ l$. Because of coverage of every $\mathbf{x}_i$, the set of $\mathcal{K}(\mathbf{x}_i, \mathbf{G}_j)$'s representing proximity to the centers triangulate $\mathbf{x}_i$ with sufficient accuracy. Therefore, in the transform space Z, there are at least $d+1$ non-trivial dot products between $\mathbf{z}_i$ and $\mathbf{H}_j$ (i.e., the respective transformed counterparts of $\mathbf{x}_i$ and $\mathbf{G}_j$) to provide dimensionality coverage:

$$\mathcal{K}(\mathbf{x}_i, \mathbf{G}_j) = \mathbf{z}_i \cdot \mathbf{H}_j \tag{39}$$

Because of triangulation, $\mathbf{H}_j$ define l dimensional space in feature space capable of supporting similar number of SVs. Note that given neighboring $\mathbf{G}_j$ are within coverage of their closest centers as well (with non-trivial cross kernel), the set of $\mathbf{H}_j$'s do not quite form an orthogonal basis in Z. However, a set of orthonormal feature vectors $\hat{\boldsymbol{v}}_j$ may be constructed in Z domain based on $\mathbf{H}_j$'s (e.g., see [10] for an example of Gram-Schmidt Orthonormalization). The first $\hat{\boldsymbol{v}}_1$ is taken in the same direction as $\mathbf{H}_1$, and the rest are determined, for example, by iterative subtraction process, so that $\hat{\boldsymbol{v}}_j$ would retain basic characteristics of the corresponding $\mathbf{H}_j$ as much as possible:

$$\hat{\boldsymbol{v}}_1 = \frac{\mathbf{H}_1}{\|\mathbf{H}_1\|} = \frac{\mathbf{H}_1}{\sqrt{\mathcal{K}(\mathbf{G}_1, \mathbf{G}_1)}}\ or\ \mathbf{H}_1 = (\mathbf{H}_1 \cdot \hat{\boldsymbol{v}}_1)\hat{\boldsymbol{v}}_1 .$$

$$\hat{\boldsymbol{v}}_2 = \frac{\mathbf{V}_2}{\|\mathbf{V}_2\|}, where\ \mathbf{V}_2 = \mathbf{H}_2 - (\mathbf{H}_2 \cdot \hat{\boldsymbol{v}}_1)\hat{\boldsymbol{v}}_1 .$$

$$\widehat{\boldsymbol{v}}_3 = \frac{\mathbf{V}_3}{\|\mathbf{V}_3\|}, where\ \mathbf{V}_3 = \mathbf{H}_3 - (\mathbf{H}_3 \cdot \widehat{\boldsymbol{v}}_2)\,\widehat{\boldsymbol{v}}_2 - (\mathbf{H}_3 \cdot \widehat{\boldsymbol{v}}_1)\,\widehat{\boldsymbol{v}}_1 \quad \cdots$$

$$\widehat{\boldsymbol{v}}_j = \frac{\mathbf{V}_j}{\|\mathbf{V}_j\|}, where\ \mathbf{V}_j = \mathbf{H}_j - \sum_{j'=1}^{j-1} \left(\mathbf{H}_j \cdot \widehat{\boldsymbol{v}}_{j'}\right) \widehat{\boldsymbol{v}}_{j'} \tag{40}$$

Note that $\|\mathbf{V}_j\|$ as $\sqrt{\boldsymbol{V}_j \cdot \boldsymbol{V}_j}$ may be expressed via dot products of $\mathbf{H}_1$ through $\mathbf{H}_j$, and therefore, accessible in terms of cross kernels of $\mathbf{G}_j$'s: $\mathcal{K}\left(\mathbf{G}_{j'}, \mathbf{G}_{j''}\right)$. All transformed data points $\mathbf{z}_i$'s (including $\mathbf{H}_j$'s) as well as hyperplane related vectors (e.g., $\widehat{\mathbf{w}}_{\|}$ and $\widehat{\mathbf{w}}_{\perp,\mathrm{k}}$) may be expressed based on the orthonormal set of $\widehat{\boldsymbol{v}}_j$'s. For example:

$\mathbf{H}_1 = \sqrt{\mathcal{K}(\mathbf{G}_1, \mathbf{G}_1)}\,\widehat{\boldsymbol{v}}_1$.

$\mathbf{H}_2 = \|\mathbf{V}_2\|\,\widehat{\boldsymbol{v}}_2 + (\mathbf{H}_2 \cdot \widehat{\boldsymbol{v}}_1)\,\widehat{\boldsymbol{v}}_1$.

$\mathbf{H}_3 = \|\mathbf{V}_3\|\,\widehat{\boldsymbol{v}}_3 + (\mathbf{H}_3 \cdot \widehat{\boldsymbol{v}}_2)\,\widehat{\boldsymbol{v}}_2 + (\mathbf{H}_3 \cdot \widehat{\boldsymbol{v}}_1)\,\widehat{\boldsymbol{v}}_1 \quad \cdots$

$$\mathbf{H}_j = \|\mathbf{V}_j\|\,\widehat{\boldsymbol{v}}_j + \sum_{j'=1}^{j-1} \left(\mathbf{H}_j \cdot \widehat{\boldsymbol{v}}_{j'}\right) \widehat{\boldsymbol{v}}_{j'} = \sum_{j'=1}^{j} h_{j,j'}\,\widehat{\boldsymbol{v}}_{j'} \tag{41}$$

where $h_{j,j'}$'s are expressed via cross kernel of $\mathbf{G}_j$'s, and $h_{j,j'} = 0\ for\ j' > j$. Similarly, for a $\mathbf{z}_i$:

$$\mathbf{z}_i = \sum_{j=1}^{l} z_{i,j}\,\widehat{\boldsymbol{v}}_j \tag{42}$$

On the basis of orthonormal set of $\widehat{\boldsymbol{v}}_j$'s, a lower triangular matrix $\mathcal{H}$ is constructed by transposing $\mathbf{H}_j$'s as rows of $\mathcal{H}$:

$$\mathcal{H} = \begin{bmatrix} \cdots \\ \boldsymbol{H}_j^{\,T} \\ \cdots \end{bmatrix} = \left[h_{j,j'}\right]_{l \times l} \tag{43}$$

The components of $\mathbf{z}_i$, i.e., $z_{i,j}$'s, are determined by inverting $\mathcal{H}$ as follows:

$$\mathcal{H}\ \mathbf{z}_i = \begin{bmatrix} \cdots \\ \boldsymbol{H}_j^{\,T} \\ \cdots \end{bmatrix} \mathbf{z}_i = \begin{bmatrix} \cdots \\ \boldsymbol{H}_j \cdot \mathbf{z}_i \\ \cdots \end{bmatrix}_{l \times 1} = \begin{bmatrix} \cdots \\ \mathcal{K}\left(\mathbf{G}_j, \mathbf{x}_i\right) \\ \cdots \end{bmatrix}_{l \times 1} \tag{44}$$

Therefore:

$$\mathbf{z}_i = \begin{bmatrix} \cdots \\ z_{i,j} \\ \cdots \end{bmatrix} = \mathcal{H}^{-1} \begin{bmatrix} \cdots \\ \mathcal{K}\left(\mathbf{G}_j, \mathbf{x}_i\right) \\ \cdots \end{bmatrix}_{l \times 1} \tag{45}$$

The above shows that a slice of original kernel matrix in (37) is used to deal with the data in feature space. The coverage and triangulation in input space implies

redundancy in the kernel matrix. In other words, if two input points $\mathbf{x}_1$ and $\mathbf{x}_2$ may be located in $\mathcal{R}^d$ based on their kernels (or distances) to $\mathbf{G}_j$'s, then per (45), $\mathcal{K}(\mathbf{x}_1, \mathbf{x}_2)$ (which is $(\mathbf{z}_1 \cdot \mathbf{z}_2)$) may be expressed in terms of $\mathcal{K}(\mathbf{G}_j, \mathbf{x}_1)$'s, $\mathcal{K}(\mathbf{G}_j, \mathbf{x}_2)$'s, and $\mathcal{K}(\mathbf{G}_j, \mathbf{G}_{j'})$'s. The accuracy in which this is possible can be used as a validation of coverage and triangulation to ensure the dimensionality of the feature space can support decomposition in (42) and (45).

The machinery of previous sections can thus be brought to bear in the feature space, as all expressions are convertible to dot products (such as projections of the data points on certain directions).

Acknowledgments. We are very grateful to Prof. Lotfi A. Zadeh and Prof. Mo Jamshidi for their mentorship and encouragement and to Ms. Ixel Chavez for all her help.

References

1. Vapnik, V.N.: An Overview of Statistical Learning Theory. IEEE Transactions on Neural Networks 10(5), 988–999 (1999)
2. Lin, C.F., Wang, S.D.: Fuzzy Support Vector Machines. IEEE Transactions on Neural Networks 13(2), 464–471 (2002)
3. Platt, J.: Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14, Microsoft Research (1998)
4. Vapnik, V.: Estimation of Dependences Based on Empirical Data. Springer (1982)
5. Osuna, E., Freund, R., Girosi, F.: An Improved Training Algorithm for Support Vector Machines. In: Proc. IEEE NNSP 1997 (1997)
6. Cervantes, J., Li, X., Yu, W.: Support Vector Machine Classification Based on Fuzzy Clustering for Large Data Sets. In: Gelbukh, A., Reyes-Garcia, C.A. (eds.) MICAI 2006. LNCS (LNAI), vol. 4293, pp. 572–582. Springer, Heidelberg (2006)
7. Ben-Hur, A., Weston, J.: A User's Guide to Support Vector Machines. In: Data Mining Techniques for the Life Sciences, pp. 223–239. Humana Press (2010)
8. Zanni, L., Serafini, T., Zanghirati, G.: Parallel Software for Training Large Scale Support Vector Machines on Multiprocessor Systems. The Journal of Machine Learning Research 7, 1467–1492 (2006)
9. Cotter, A., Srebro, N., Keshet, J.: A GPU-Tailored Approach for Training Kernelized SVMs. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 805–813. ACM (August 2011)
10. Giesen, J., Laue, S., Mueller, J.K.: Basis Expansions for Support Vector Machines, `http://cgl.uni-jena.de/pub/Publications/WebHome/CGL-TR-49.pdf`

Author Index