Hujun Yin
Wenjia Wang
Victor Rayward-Smith (Eds.)

LNCS 6936

Intelligent Data Engineering and Automated Learning – IDEAL 2011

12th International Conference
Norwich, UK, September 2011
Proceedings

Springer

Lecture Notes in Computer Science 6936

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Hujun Yin   Wenjia Wang
Victor Rayward-Smith (Eds.)

Intelligent Data Engineering and Automated Learning – IDEAL 2011

12th International Conference
Norwich, UK, September 7-9, 2011
Proceedings

Springer

Volume Editors

Hujun Yin
University of Manchester
Manchester, M60 1QD, UK
E-mail: hujun.yin@manchester.ac.uk

Wenjia Wang
University of East Anglia
Norwich, NR4 7TJ, UK
E-mail: wenjia.wang@uea.ac.uk

Victor Rayward-Smith
University of East Anglia
Norwich, NR4 7TJ, UK
E-mail: vjrs@uea.ac.uk

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-23877-2 e-ISBN 978-3-642-23878-9
DOI 10.1007/978-3-642-23878-9
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2011935605

CR Subject Classification (1998): I.2, H.3-4, F.1, H.2.8, I.5, J.3

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

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

Preface

The IDEAL conference is an established and broad interdisciplinary forum for experts, researchers, leading academics, practitioners and industries in fields such as machine learning, information processing, data mining, knowledge management, bio-informatics, neuro-informatics, bio-inspired models, agents and distributed systems, and hybrid systems. It has enjoyed a long, vibrant and successful history over 13 years and across ten locations in six different countries. It continues to evolve so as to embrace emerging topics and exciting trends. The conference papers provide a good sample of these topics from methodologies, frameworks and techniques to applications and case studies. The techniques include evolutionary algorithms, artificial neural networks, association rules, probabilistic modelling, agent modelling, particle swarm optimization and kernel methods. The applications cover regression, classification, clustering and generic data mining, biological information processing, text processing, physical systems control, video analysis and time series analysis. At the center of these lie the core themes of the IDEAL conferences: data analysis and data mining and associated learning paradigms and systems.

This volume contains the papers accepted and presented at the 12th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL 2011) held during September 7–9, 2011 at the University of East Anglia, in Norwich, UK. All submissions were strictly peer-reviewed by the Programme Committee and only the papers judged to be of sufficient quality and novelty were accepted and included in the proceedings. The authors of the papers and attendees come from all over the world, Europe, Africa, Asia, Middle East and New Zealand, to North and South America.

IDEAL 2011 enjoyed outstanding keynote speeches by distinguished guest speakers: Alexander Gorban of the University of Leicester, Tom Heskes of Radbound University of Nijmegen, Richard Harvey and Gavin Cawley, both of the University of East Anglia, and Iead Rezek of Imperial College London.

We would like to thank all the people who devoted so much time and effort to the successful running of the conference and in particular the members of the Programme Committee and reviewers, as well as the authors of the papers included in the proceedings of this conference. We are also grateful for the hard work by the local organizing team and the support from the University of East Anglia. Continued support and collaboration from Springer, especially from the LNCS editors, Alfred Hofmann and Anna Kramer, are also appreciated.

July 2011

Hujun Yin
Wenjia Wang
Victor Rayward-Smith

Organization

General Chair

Victor Rayward-Smith	University of East Anglia, UK
Hujun Yin	University of Manchester, UK

Programme Chair

Wenjia Wang	University of East Anglia, UK

Programme Co-chairs

David Clifton	University of Oxford, UK
José Alfredo F. Costa	Federal University, Brazil
Emilio Corchado	University of Salamanca, Spain

International Advisory Committee

Lei Xu (Chair)	Chinese University of Hong Kong, Hong Kong
Yaser Abu-Mostafa	CALTECH, USA
Shun-ichi Amari	RIKEN, Japan
Michael Dempster	University of Cambridge, UK
José R. Dorronsoro	Autonomous University of Madrid, Spain
Nick Jennings	University of Southampton, UK
Soo-Young Lee	KAIST, South Korea
Erkki Oja	Helsinki University of Technology, Finland
Latit M. Patnaik	Indian Institute of Science, India
Burkhard Rost	Columbia University, USA
Xin Yao	University of Birmingham, UK

Steering Committee

Hujun Yin (Chair)	University of Manchester, UK
Laiwan Chan (Chair)	Chinese University of Hong Kong, Hong Kong
Nigel Allinson	University of Sheffield, UK
Yiu-ming Cheung	Hong Kong Baptist University, Hong Kong
Emilio Corchado	University of Burgos, Spain
Colin Fyfe	University of the West of Scotland, UK
Marc van Hulle	K.U. Leuven, Belgium
Samuel Kaski	Helsinki University of Technology, Finland
John Keane	University of Manchester, UK

Jimmy Lee	Chinese University of Hong Kong, Hong Kong
Malik Magdon-Ismail	Rensselaer Polytechnic Institute, USA
Zheng Rong Yang	University of Exeter, UK
Ning Zhong	Maebashi Institute of Technology, Japan

South America Liaison

José Alfredo F. Costa	Federal University, Brazil

Asia Liaison

Yiu-ming Chueng	Funding Chair of IEEE CIS Hong Kong Chapter, Hong Kong

Publicity Co-chairs

Beatriz de la Iglesia	University of East Anglia, UK
Emilio Corchado	University of Salamanca, Spain
Dacheng Tao	Hong Kong Polytechnic University, Hong Kong

Programme Committee

Ajith Abraham	Norwegian University of Science and Technology, Norway
Jesús Alcalá-Fdez	University of Granada, Spain
Jamil Al Shaqsi	Sultan Qaboos University, Oman
Davide Anguita	University of Genoa, Italy
Bruno Apolloni	University of Milan, Italy
Javier Bajo	Pontificial University of Salamanca, Spain
Bruno Baruque	University of Burgos, Spain
Antonio Bella	Universidad Politécnica de Valencia, Spain
José Manuel Benítez	University of Granada, Spain
Mikael Boden	University of Queensland, Australia
Lourdes Borrajo	University of Vigo, Spain
Vicente Botti	Polytechnic University of Valencia, Spain
David Camacho	Universidad Autónoma de Madrid, Spain
Jose. Calvo-Rolle	Universidad de la Coruña, Spain
Maria do Carmo-Nicoletti	Federal University of Sao Carlos, Brazil
Matthew Casey	University of Surrey, UK
Oscar Castillo	Tijuana Institute of Technology, Mexico
Darryl Charles	University of Ulster, UK
Sung-Bae Cho	Yonsei University, Korea
Seungjin Choi	POSTECH, Korea
Juan M. Corchado	University of Salamanca, Spain
Rafael Corchuelo	University of Seville, Spain

Raúl Cruz-Barbosa	Universitat Politècnica de Catalunya, Spain
Leticia Curiel	University of Burgos, Spain
Alfredo Cuzzocrea	University of Calabria, Italy
Keshav Dahal	University of Bradford, UK
Bernard de Baets	Ghent University, Belgium
André de Carvalho	University of Sâo Paulo, Brazil
Ricardo Del Olmo	University of Burgos, Spain
Fernando Díaz	University of Valladolid, Spain
José Dorronsoro	Autónoma de Madrid University, Spain
Jochen Einbeck	University of Durham, UK
Igor Farkas	Comenius University in Bratislava, Slovakia
Florentino Fernández	University of Vigo, Spain
Jose A. Ferreira	Federal University, Brazil
Francisco Ferrer	University of Seville, Spain
Jan Feyereisl	University of Nottingham, UK
Juan J. Flores	Michoacana University, Mexico
Richard Freeman	Michael Page International, UK
Marcus Gallagher	University of Queensland, Australia
Matjaz Gams	Jozef Stefan Institute Ljubljana, Slovenia
John Gan	University of Essex, UK
Salvador García	University of Jaen, Spain
Raúl Giráldez	Pablo de Olavide University, Spain
Mario A. García-Martínez	Instituto Tecnológico de Orizaba, Mexico
David González-Ortega	University of Valladolid, Spain
Petro Gopych	Universal Power Systems USA-Ukraine LLC, Ukraine
Marcin Gorawski	Silesian University of Technology, Poland
Lars Graening	Honda Research Institute Europe GmbH, Germany
Manuel Graña	University of the Basque Country, Spain
Jerzy Grzymala-Busse	University of Kansas, USA
Aboul Ella Hassanien	Cairo University, Egypt
Ioannis Hatzilygeroudis	University of Patras, Greece
P. Javier Herrera	Complutense University of Madrid, Spain
Álvaro Herrero	University of Burgos, Spain
Michael Herrmann	University of Edinburgh, UK
Jaakko Hollmén	Helsinki University of Technology, Finland
Vasant Honavar	Iowa State University, USA
Wei-Chiang S. Hong	Oriental Institute of Technology, Taiwan
David Hoyle	University of Manchester, UK
Jen-Ing Hwang	Fu Jen Catholic University, Taiwan
Jose A. Iglesias	Carlos III University, Spain
Vicent Julián	Universidad Politécnica de Valencia, Spain
Juha Karhunen	Helsinki University of Technology, Finland
Kyung-Joong Kim	Sejong University, Korea

Frank Klawonn	Ostfalia University of Applied Sciences, Germany
Zofia Kruczkiewicz	Wrocław University of Technology, Poland
Rudolf Kruse	Otto-von-Guericke-Universität Magdeburg, Germany
Benaki Lairenjam	JMI university New Delhi, India
Lenka Lhotská	Czech Technical University, Czech Republic
Honghai Liu	University of Portsmouth, UK
Eva Lorenzo	University of Vigo, Spain
Wenjian Luo	University of Science and Technology of China, China
Frederic Maire	Queensland University of Technology
Roque Marín	University of Murcia, Spain
Giancarlo Mauri	University of Milano Bicocca, Italy
José M. Molina	University Carlos III of Madrid, Spain
Daniel Neagu	University of Bradford, UK
Yusuke Nojima	Osaka Prefecture University, Japan
Chung-Ming Ou	Kainan University, Taiwan
Vasile Palade	University of Oxford, UK
Stephan Pareigis	Hamburg University of Applied Sciences, Germany
Juan Pavón	University Complutense of Madrid, Spain
Carlos Pereira	University of Coimbra, Portugal
Bernardete Ribeiro	University of Coimbra, Portugal
Fabrice Rossi	Télécom Paris Tech, France
Roberto Ruiz	Pablo de Olavide University, Spain
Yanira Santana	University of Salamanca, Spain
José Santos	Universidade da Coruña, Spain
Javier Sedano	University of Burgos, Spain
Dragan Simic	Novi Sad Fair, Serbia
Michael Small	Hong Kong Polytechnic University, Hong Kong
Dante I. Tapia	University of Salamanca, Spain
Peter Tino	University of Birmingham, UK
Alicia Troncoso	Pablo de Olavide University, Spain
Eiji Uchino	Yamaguchi University, Japan
Alfredo Vellido	Universidad Politécnica de Cataluña, Spain
Sebastián Ventura	University of Cordoba, Spain
José R. Villar	University of Oviedo, Spain
Lipo Wang	Nanyang Technological University, Singapore
Michal Wozniak	Wroclaw University of Technology, Poland
Wu Ying	Northwestern University, USA
Ron Yang	University of Exeter, UK
Du Zhang	California State University, USA
Huiyu Zhou	Queen's University Belfast, UK
Rodolfo Zunino	University of Genoa, Italy

Local Organizing Committee

Wenjia Wang (Chair)	University of East Anglia, UK
Vic Rayward-Smith (Chair)	University of East Anglia, UK
Beatriz de la Iglesia	University of East Anglia, UK
Geoffrey Guile	University of East Anglia, UK
Tony Bagnall	University of East Anglia, UK
Gavin Cawley	University of East Anglia, UK
Aristidis K. Nikoloulopoulos	University of East Anglia, UK
Elena Kulinskaya	University of East Anglia, UK
Richard Harrison	University of East Anglia, UK
Heidi Prada	University of East Anglia, UK
Oliver Kirkland	University of East Anglia, UK

Table of Contents

Adaptive K-Means for Clustering Air Mass Trajectories

Alex Mace[1], Roberto Sommariva[2], Zoë Fleming[3], and Wenjia Wang[1]

[1] University of East Anglia, Computing Sciences, Norwich, NR4 7TJ, UK
{alex.mace,wenjia.wang}@uea.ac.uk

[2] University of East Anglia, Environmental Sciences, Norwich, NR4 7TJ, UK
r.sommariva@uea.ac.uk

[3] University of Leicester, Department of Chemistry, Leicester, LE1 7RH, UK
zf5@le.ac.uk

Abstract. Clustering air mass trajectories is used to identify source regions of certain chemical species. Current clustering methods only use the trajectory coordinates as clustering variables, and as such, are unable to differentiate between similar shaped trajectories that have different source regions and/or seasonal differences. This can lead to a higher variance in the chemical composition within each cluster and loss of information. We propose an adaptive K-means clustering algorithm that uses both the trajectory variables and the associated chemical value. We show, using carbon monoxide data from the Cape Verde for 2007, that our method produces a far more informative clustering than the existing standard method, whilst achieving a lower level of subjectivity.

1 Introduction

Air mass back trajectories describe the history of an air mass, giving the latitude, longitude and pressure of the air mass in set intervals back in time measured from the point of arrival. In air mass trajectory studies, the air mass back trajectories are calculated over a period of time then ordered into groups based on their origin and path. Chemical observations made at the arrival time for each trajectory can be assigned to the relevant group and difference between groups can identify and provide information on chemical source regions. The trajectories can be grouped manually according to geographical regions crossed (Sector analysis), but this is time consuming, requires expert domain knowledge and is very subjective so the preference is to use more objective clustering techniques.

Clustering algorithms used for trajectory analysis include the Dorling algortihm [1,2,4,6], Ward's algorithm [5,9], and average linkage clustering [3,10]. As the Dorling had been used more frequently than others, it was hence viewed as a de-facto standard in practice in the field. Applications of clustering typically only use the latitude and longitude coordinates and thus focus on grouping similar shaped trajectories in terms of geographical regions crossed. Evaluation of the clustering techniques used is rare but a study using air mass back trajectories arriving at Athens did propose that the K-means algorithm was less sensitive

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 1–8, 2011.

to trajectory arrival height compared to hierarchical clustering algorithms and self-organising maps [7]. K-means [8], however, requires that the number of clusters, k, is known or preset in advance with an estimated value. This is very inconvenient in practice particularly when there is little or no prior knowledge on the number of clusters in the data. Initial calculations in this study (using the 2007 air mass back trajectories for the Cape Verde) also suggested that the more objective algorithms, PAM [12], TwoStep and self-organising maps (latter two both using PASW modeller 13 default methods [11]) would not be suitable for air mass trajectory clustering as they produced inferior clustering models to that of the Dorling algorithm both in terms of intra-cluster similarity and number of informative clusters. Therefore our aim was to adapt the K-means algorithm to propose a model from a range of k.

A key issue when using just latitude and longitude as clustering variables is that similar shaped trajectories can have widely different chemistry, for example, if they are on the border between two different geographical regions such as land and sea. Also such a method would not identify any seasonal differences in chemistry that exist over a particular region. To address these issues, we propose to use the chemistry in addition to the latitude and longitude as clustering variables and observe whether it gives a more informative clustering.

2 Related Work

The Dorling algorithm [4], using latitude and longitude as clustering variables, appears to be the most popular method in this field. The initial clustering uses K-means, where k is 30+, and the total root mean square deviation (TRMSD) of the model calculated, where the root mean square deviation between a trajectory and its cluster centre is:

$$RMSD = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x_i})^2}{n}} \tag{1}$$

The two clusters with the shortest Euclidean distance between their respective centres are then merged and the percentage change in TRMSD caused by reducing k by one is calculated. This process iterates until only one cluster remains. The number of clusters of each model and their percentage changes in TRMSD are plotted and candidate models are those that immediately preceed a sharp increase in TRMSD (typically 5% [2,6]) which indicates that two greatly dissimilar clusters have been merged. From the candidate models the user chooses the one that is most informative for the particular study. The quality of the clustering is dependent on a good choice of initial cluster centres that adequately cover the spread of the trajectories, which adds a subjective element. To limit the subjectivity evenly spaced out artificial trajectories that cover the defined region can be used as the initial cluster centres [1,6].

3 Adaptive K-Means

We modified the classic K-means to propose an optimum cluster model from a range of k. Each K-means model is quantitatively evaluated by the total sum of squares (TSS) from each trajectory to its cluster centre. For each K-means model, 100 sets of random initial cluster centres are used to increase the robustness of the algorithm without greatly impacting performance, with the model with the minimum TSS being proposed. The distance metric used is:

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{y}_i + w(\mathbf{x}_j - \mathbf{y}_j))^2} \qquad (2)$$

where j is the chemical species reading and w the weight that is applied to that chemical species. Initially we generate a K-means model with a value of k greater than the maximum number of useful clusters expected and then sequentially reduce k by one and observe the percentage increase in TSS between the new model and the previous model. As each cluster model is independent of cluster models with higher value of k it is not necessary to have as high a initial k value as the Dorling algorithm. Typically studies of this nature identify <10 useful clusters as higher values tend to produce clusters containing too few trajectories to be informative and so we propose an initial value of k to be 15. To determine when to stop decreasing k we use a threshold, α, at which point the percentage increase in TSS on decreasing k would indicate an unacceptable loss in clustering quality. Whereas sharp increases are observed at 5% when using TRMSD, we did not observe sharp increases until 10% when using TSS. Therefore we propose a 10% increase in TSS as the α threshold.

3.1 Calculating the Weight

To determine the most appropriate weight for the chemical we generate an increasing range of weights starting at zero (the chemistry has no influence on the clustering). Starting at zero, we used the adaptive K-means algorithm to propose a model for each weight and then observed the percentage change in TSS caused by increasing the weight. Sharp increases in TSS are observed when increasing the weight either due to the number of proposed clusters decreasing or a large reassortment in cluster contents and therefore candidates for the 'optimum' weight are those that immediately precede these sharp increases. The threshold percentage increase in TSS for a sharp increase is defined as β and using Cape Verde data, we propose a value of 5%. From the these candidates the most informative one is that maximises the chemical contribution without overpowering the trajectory coordinates and producing regionless clusters. To compare the regionality of the weighted models we use trajTSS, which is the percentage increase in TSS using only the trajectory variables when going from the zero weighted model to the weighted model. We define the trajTSS at which point the contributions from the chemistry and trajectories are best balanced as γ and therefore from candidate weighted models we propose the model that is closest to this value. From applying the algorithms to Cape Verde data, we propose that good balance is achieved when γ is 50% Therefore, the algorithm is given in full as follows:

1. *Choose an initial value for k, α, β and γ.*
2. *Create an increasing range of weights, $\mathbf{w}$ to be applied to the chemistry, initially weighting the chemical value by zero.*
3. *Perform the K-means algorithm, using k clusters, and Euclidean distance, randomly choosing the set of data used as the initial cluster centers.*
4. *Calculate the TSS of the K-means model.*
5. *Repeat Steps 3-4 for a further 99 random sets of cluster centres, and store the K-means model that has the lowest TSS as the cluster model for that value of k.*
6. *Reduce k by one and repeat Steps 3-5. Calculate the percentage change when going from the the $k+1$ model to the k model.*
7. *Repeat Step 6 until the percentage change in TSS is $> \alpha$. The proposed 'optimal' clustering is therefore the $k+1$ model.*
8. *Repeat Steps 3 to 7 for each of the chemical weights. If the percentage change in TSS in going from the $\mathbf{w}_i$ model to the $\mathbf{w}_{i+1}$ model is $> \beta$, store the $\mathbf{w}_i$ model as a candidate.*
9. *If all candidate models trajTSS are $< \gamma$ then repeat Steps 3-7 using a wider range of weights.*
10. *Having identified the weight region at which the trajTSS is closest to γ, consider repeating Steps 3 to 7 using a narrower range of weights focused on this region.*
11. *The proposed model is the candidate that has a trajTSS closest to γ.*

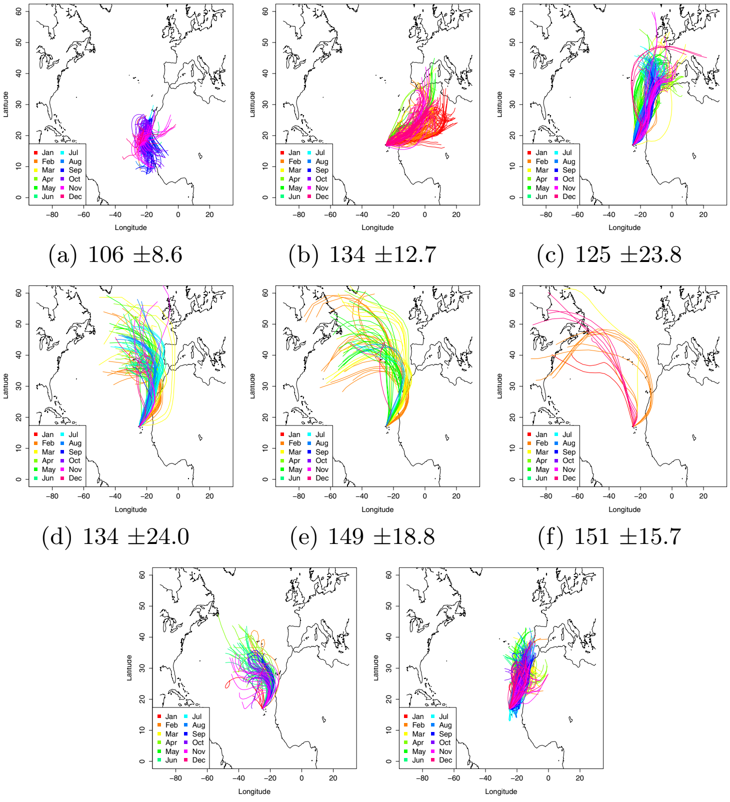
4 Results

To demonstrate the effectiveness of our algorithm we compared it to the Dorling algorithm using five day back trajectories, calculated every six hours, arriving at Cape Verde for 2007. We used carbon monoxide (CO) as the chemical species as it had a good data quality for 2007 and has strong anthropogenic sources. Each trajectory was assigned the mean of the CO data for the six preceeding hours and trajectories without a valid CO reading were discarded. Post cleaning, the data set contained 1124 trajectories each with an associated CO concentration. For the Dorling algorithm we used a RMSD threshold of 5% and 30 seed trajectories of 50^o in length, evenly spaced from 35^o clockwise of the equator going anticlockwise to 165^o anticlockwise of the equator. The Dorling model proposed eight clusters (Fig. 1). For the Adaptive K-means algorithm we set the parameters as follows; $k = 15, \alpha = 10\%, \beta = 5\%, \gamma = 50\%$. The range of weights used for CO was 0 to 3 by 0.1. The algorithm took an acceptable 10 minutes to complete and weighted CO by 2.3 and proposed seven clusters (Fig. 2).

The Dorling model achieves good separation of trajectories based on their shape and length. One can make the general observations that the lowest CO

Table 1. P-values for the Multiple Comparison tests of the cluster CO composition for the Dorling model (below the diagonal) and the Adaptive K-means model (above the diagonal). Values above 0.05 are deemed to be insignificant and are in bold font

Cluster	a	b	c	d	e	f	g
a	-	0.00	0.00	0.00	0.00	0.00	0.00
b	0.00	-	0.00	0.00	0.00	0.00	0.01
c	0.00	0.01	-	0.00	0.00	0.00	0.00
d	0.00	**1.00**	0.02	-	0.00	**1.00**	0.00
e	0.00	0.00	0.00	0.02	-	0.00	0.00
f	0.00	0.00	0.00	**0.08**	**1.00**	-	0.00
g	0.00	0.00	0.00	0.00	0.00	0.00	-
h	0.00	0.00	**0.31**	0.00	0.00	0.00	**1.00**



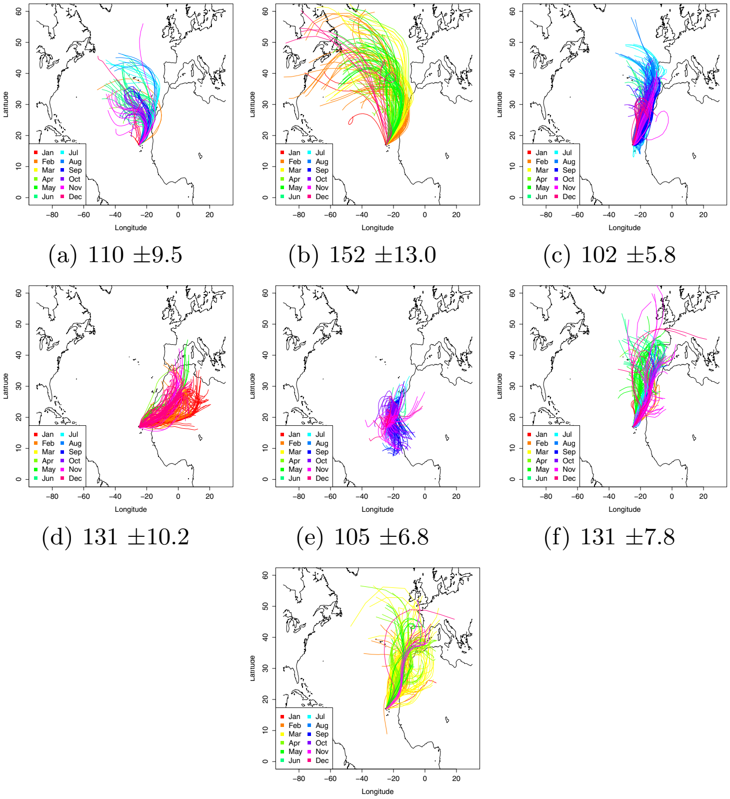
(a) 106 ±8.6 (b) 134 ±12.7 (c) 125 ±23.8

(d) 134 ±24.0 (e) 149 ±18.8 (f) 151 ±15.7

(g) 117 ±16.9 (h) 119 ±21.8

Fig. 1. The eight clusters from the Dorling model and their mean concentration of CO (ppbV) and standard deviation

concentrations are associated with slow local trajectories (Fig. 1a) and the highest associated with fast trajectories from North America (Fig. 1e and 1f). In terms of chemical composition the clustering is not greatly informative as only cluster (1a) has a significantly different chemical compostion from all other clusters (Table 1). This is likely due to the high standard deviation of CO readings relative to the cluster means, caused by clusters containing trajectories with a similar shape but different source regions or seasonality. This is particularly apparent in clusters (1c) and (1h) which appear to contain trajectories with source regions in the Atlantic and Europe, and cluster (1d) with trajectories with source regions in North America and those with an Atlantic influence only. Also cluster (1f) is not informative by itself as it contains very few trajectories relative to the other clusters and is highly similar to cluster (1e) both in source region and chemical composition.



(a) 110 ±9.5

(b) 152 ±13.0

(c) 102 ±5.8

(d) 131 ±10.2

(e) 105 ±6.8

(f) 131 ±7.8

(g) 156 ±8.6

Fig. 2. The seven clusters from the Adaptive K-means model and their mean concentration of CO (ppbV) and standard deviation

The adaptive K-means model produces a markedly more informative model with regards to the chemical information (Fig. 2). Compared to the Dorling model the standard deviation of CO within each cluster is lower suggesting a less variation in soruce regions witihin a cluster. Only the CO composition of clusters (2d) and (2f) are not significantly different (Table 1). Clusters (2a, d and e) are similar to their Dorling model counterparts, clusters (1g, b, a) respectively), albeit with a lower standard deviation. The trajectories that travel near the European/African coastline, but are only influenced by the Atlantic and hence low levels of CO, are separated into their own cluster and therefore do not bias the trajectories with a similar shape from Europe. All fast trajectories with a source region in North America are grouped into a single cluster (2b). The European trajectories are separated into clusters (2f) and (2g), which although the trajectory shapes are similar, have significantly different chemical compositions. Cluster (2g), however, contains mainly spring trajectories whereas cluster (2f) are mainly summer trajectories and therefore the adaptive K-means method is able to identify seasonal effects. The trends we observe from this model are that the trajectories with the highest amount of associated CO are infact from Europe during the spring closely followed by fast trajectories from North America, and that the lowest trajectories are those with a Atlantic only influences.

5 Conclusion

We have developed and used an algorithm for use in trajectory clustering. This algorithm adapts the K-means algorithm and differs from previous algorithms in this field in that it also takes into account the chemistry associated with the trajectory and also contains less subjectivity. Using real data from Cape Verde we show that the algorithm is capable of separating trajectories with similar shapes but differing source regions into separate clusters and also able to identify seasonal effects in chemical emissions.

Acknowledgments. The authors thanks the British Atmospheric Data Centre Trajectory Service and the European Centre for Medium-Range Weather Forecasts for the calculation of the 2007 air mass back trajectories. The authors acknowledge the University of York Cape Verde team (Professor Lucy Carpenter, Professor Alastair Lewis, Dr Katie Read, Dr James Lee, Dr James Hopkins and Sarah Moller) for the CO concentration data for Cape Verde for 2007 and thank Bruno Faria, Luis Mendes and Helder Lopes from the Instituto Nacional de Meteorologia and Geofisica (INMG) of Cape Verde for the daily support and management of Cape Verde Atmospheric Observatory.

References

1. Borge, R., Lumbreras, J., Vardoulakis, S., Kassomenos, P., Rodríguez, E.: Analysis of long-range transport influences on urban PM10 using two-stage atmospheric trajectory clusters. Atmospheric Environment 41, 4434–4450 (2007)
2. Brankov, E., Rao, S.T., Porter, P.S.: A trajectory-clustering-correlation methodology for examining the long-range transport of air pollutants. Atmospheric Environment 32, 1525–1534 (1998)

3. Cape, J.N., Methven, J., Hudson, L.E.: The use of trajectory cluster analysis to interpret trace gas measurements at Mace Head, Ireland. Atmospheric Environment 34, 3651–3663 (2000)
4. Dorling, S.R., Davies, T.D., Pierce, C.E.: Cluster analysis: a technique for estimating the synoptic meteorological controls on air and precipitation chemistry–Results from Eskdalemuir, South Scotland. Atmospheric Environment. Part A. General Topics 26, 2583–2602 (1992)
5. Eneroth, K., Kjellström, E., Holmén, K.: A trajectory climatology for Svalbard; investigating how atmospheric flow patterns influence observed tracer concentrations. Physics and Chemistry of the Earth, Parts A/B/C 28, 1191–1203 (2003)
6. Jorba, O., Pérez, C., Rocadenbosch, F., Baldasano, J.M.: Cluster analysis of 4-day back trajectories arriving in the Barcelona area, Spain, from 1997 to 2002. Journal of Applied Meteorology 43, 887–901 (2004)
7. Kassomenos, P., Vardoulakis, S., Borge, R., Lumbreras, J., Papaloukas, C., Karakitsios, S.: Comparison of statistical clustering techniques for the classification of modelled atmospheric trajectories. Theoretical and Applied Climatology 102, 1–12 (2010)
8. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
9. Moody, J.L., Galloway, J.: Quantifying the relationship between atmospheric transport and the chemical composition of precipitation on Bermuda. Tellus 40, 463–479 (1988)
10. Moy, L.A., Dickerson, R.R., Ryan, W.F.: Relationship between back trajectories and tropospheric trace gas concentrations in rural Virginia. Atmospheric Environment 28, 2789–2800 (1994)
11. SPSS Inc.: PASW 13 Modeler for Windows. Rel 13.0.0.366. SPSS Inc.,Chicago (2009)
12. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 4th edn. Academic Press, London (2008)

On Domain Independence of Author Identification

Masato Shirai and Takao Miura

HOSEI University, Dept.of Elect.& Elect. Engr.
3-7-2 KajinoCho, Koganei, Tokyo, 184–8584 Japan

Abstract. *Latent Dirichlet Allocation* (LDA) is a probabilistic framework by which we may assume each word carries probability distribution to each topic and a topic carries a distribution to each document. By putting all the documents together into one collection by each author, it is possible to identify authors. Here we show that author identification is fully reliable within a framework of LDA independent of documents domains by learning incomplete and massive documents.

Keywords: Text Mining, Author Topic Mode, Author Identification, Domain Independence.

1 Background

Recently there have been put much attention on *author-identification* using extracting several features of authors. Usually some features could be examined to identify authors. Among others, we may distinguish *content words* from *function words* where the former carries general (own) semantics such as nouns, verbs, adjectives and adverbs and the latter plays auxiliary and grammatical roles such as pronouns, prepositions and conjunctions as well as postpositional particles. As for *function words*, we may think about average size of words or sentences, comma and punctuation marks and n-gram words. The approach is called *stylometric* [4,**?**]. On the other hand, it is well-known that content word distribution is not really useful for author identification[5].

A *latent topic model* allows us to identify an author. More specifically we estimate word distribution over topics using *Latent Dirichlet Allocation* (LDA) technique[1]. In this model, a document can be considered as random mixture of several topics, and every topic can be described as probability distribution over words. LDA provides us to examine the mixture over topics and each topic is described over word probability distribution, but the mixture (of topics) is *dynamic* in a sense of *hyper probability*, not depending on training data but on Dirichlet probability model. In fact, we estimate the model from *prior probability distribution* of words in training data by means of Gibbs sampling. Each document is described by means of random mixture over topics which means a topic corresponds to latent semantic unit described by a collection of words[1].

[1] In other words, a *topic* doesn't mean an explicit subject but a kind of cluster putting together by some probabilistic measure.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 9–16, 2011.

In an *author topic model*, we assume topic distributions over authors[6], instead of documents, where each *author* is associated with a multi-nomial distribution over topics and each topic with a multinomial distribution over words. Interesting is that a document by multiple authors can be modeled as a distribution over topics that is random mixture of the distributions associated with the authors. This model assumes that every document reflects author characteristics on topic distribution. By comparing topic distribution of authors with ones of training documents, we expect author identification. In this investigation, we examine two probability distributions directly by χ^2 values or KL divergence.

One of the issues of LDA is that there have been proposed several author identifications but they depend on specific aspects of implicit but subject categories. On the other hand, it is widely believed that a latent topic model is independent of categories, and so is true for an author topic model.

In this investigation we examine whether an author topic model is valid or not for several subject categories such (novels and news papers in this work). In a case of *editorial columns*, we expect that these columns discuss about same subjects and we can identify the authorship.

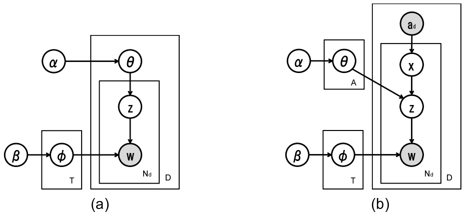
This work is organized as follows. We introduce a latent topic model and an author topic model in section 2 as well as LDA. In section 4 we discuss how to identify authorship based on the model.

2 Latent Dirichlet Allocation

2.1 A Topic Model

A *latent topic model* is an assumption that each document can be described as random mixture over topics and each topic as word distribution, while a *document model* is a probabilistic assumption that each document follows a mixed multi-nominal distribution over words.

A topic model is known as a probabilistic vehicle to extract latent semantics from documents. A *probabilistic Latent Semantic Indexing* (pLSI) is also random mixture over topics and each topic is described over word distribution, but the mixture (i.e., topics distribution) is determined in advance according to training data[3]. LDA [1] is *dynamic* in a sense that topic mixture is determined according to Dirichlet probability distribution, not depending on training data. In fact, we



(a) (b)

Fig. 1. Graphical Models for LDA and Author-Topic Model

estimate the model from *prior distribution* of topics in training data by means of Gibbs sampling so that we estimate the topic probability distribution.

In part (a) of a figure 1, let us illustrate our situation by hierarchical Bayesian model, called a *graphical model* using *plate* notation, which contains a latent variable z. A word "*latent*" means it is hard to observe these values explicitly and we like to estimate probability distributions over topics $\mathbf{z}$ given observed words $\mathbf{w}$ with an assumption of Dirichlet prior distribution of topics.

Dirichlet prior distribution $p(\phi)$ over topics is described by a hyper parameter β and Dirichlet prior distribution $p(\theta_d)$ over topics with respect to documents d is described by another hyper parameter α. Then we can show multi-nominal distribution $p(w|z, \phi_z)$ of words w over each topic z represented by ϕ_z with total N words. Also, given a number of topics by T, we may describe multi-nominal distribution $p(z|\theta_d)$ of topics z in a document d represented by θ_d with D documents, $d = 1, .., D$. Estimation of ϕ and θ means that we need information about latent topics in corpus and the weights of the topics in each document.

In this framework words are generated based on LDA as follows. Given a topic t according to $p(\phi_t)$ drawn from $\phi(\beta)$ and a document d according to $p(\theta_d)$ from $\theta(\alpha)$, we generate a topic z_i from a multi-nominal probability distribution $p(z|\theta_d)$ in a document d at a position i (in a document d) and a word w_i from a distribution $p(w|z, \phi_z)$.

2.2 An Author Topic Model

By using a topic model, no explicit author information is provided because an author write several documents with a variety of latent topics within each document. In an *author topic* model similar to a latent topic model, we introduce a new variable x to capture authors as shown in part (b) of a figure 1. An author x drawn from $\mathbf{a_d}$ uniformly produces a document d, a topic z is selected in a probabilistic manner according to topic distribution of documents as a topic model, and finally words are generated from the chosen topics. Here we assume each document (with respect to an author) is associated with topic distribution drawn from Dirichlet probability distribution. The mixture weights (with respect to the author) give a latent topic z, which in turn gives a word w with multi-nominal probability distribution where the word is associated with the topic z drawn from the Dirichlet distribution $p(\phi)$ over topics.

The author-topic model differs only from a notion of *authors*: each author is assumed to have topic distribution while we assume every document carries topic distribution in a topic model. Since every author is chosen uniformly, a topic model can be seen as a author topic model with single author. To obtain an author topic model, we consider all the documents written by an author a as a single document d_a and estimate latent topic distribution z over all the d_a.

2.3 Obtaining Author Topic model

To estimate topic distribution θ associated with both authors, and word distribution ϕ associated with authors, there have been several techniques proposed such as EM algorithms and variational Bayesian approach. Among other, Gibbs sampling works well very often. Here we discuss how to estimate distributions by Gibbs sampling[2]. Gibbs sampling allows us to obtain posterior probability distribution over topics by which we obtain θ and ϕ. An assumption of Markov chain between state change produces a topic j drawn z from distribution on other topics defined as follows:

$$P(z_i = j, x_i = k | w_i = m, \mathbf{z}_{-i}, \mathbf{x_{-i}}, \mathbf{w}_{-i}, \mathbf{a_d}) \propto$$

$$\frac{C^{WT}_{mj} + \beta}{\sum_m C^{WT}_{mj} + V\beta} \frac{C^{DT}_{dj} + \alpha}{\sum_j C^{DT}_{dj} + T\alpha}$$

Here "$z_i = j, x_i = k$" means the assignment of i-th word to a topic j and of i-th word to an author k, $w_i = m$ means i-th word is m, $\mathbf{z_{-i}}, \mathbf{x_{-1}}$ and $\mathbf{w_{-i}}$ means all topic/author/word assignment except i. Also C^{AT}_{kj} means the number of times an author k is assigned to a topic j, C^{WT}_{mj} means the number of times a word m is assigned to a topic j and V, T the numbers of words and topics respectively.

The formula describes the conditional probability by marginalizing ϕ (the probability of words given an author) and θ (the probability of topics given an author). Finally the random variables ϕ and θ can be estimated as follows:

$$\phi_{mj} = \frac{C^{WT}_{mj} + \beta}{\sum_m C^{WT}_{mj} + V\beta}, \quad \theta_{dj} = \frac{C^{AT}_{dj} + \alpha}{\sum_j C^{AT}_{dj} + T\alpha}$$

Given initial values of the probabilities, we repeat the transition drawing z from this distribution for many times, called *iteration*. We get to convergence, the final state shows latent topic distribution θ and latent word distribution ϕ.

3 Identifying Authors

Given a collection of documents (assumed by single author), let us describe how to identify the author using an author topic model. We assume a collection of authors $d_1, .., d_D$ where each author d_i is considered as all the union of documents written by d_i. First, we estimate an author-topic model for $\{d_1, .., d_D\}$ and we get word distribution ϕ for topics and topic distribution θ for authors. Given a test author d and the documents $V = \{v_1, .., v_d\}$ by d, we get word distribution for topics and topic distribution θ for V. We like to identify who is d. Here we assume every author d_i or d can be characterised by means of topic distribution, and examine similarity between the distributions of a test author d and the known authors $d_1, .., d_D$.

To compare two authors with each other, we examine two kinds of word distributions by using *KL divergence* or X^2 *value*. Given two distributions θ_x and θ_d, KL divergence $KL(\theta_x // \theta_d)$ and X^2 value is defined as follows:

$$KL(\theta_x // \theta_d) = \sum_i \theta_{xi} log \frac{\theta_{xi}}{\theta_{di}}, \quad X^2 = \sum_i \frac{(\theta_{di} - \theta_{xi})^2}{\theta_{xi}}$$

When two distributions become similar, we get smaller values of KL divergence and X^2 values, and we estimate the authorship as the smallest ones.

Let us note that another approach utilized *symmetric* KL divergence[6] since KL divergence is not symmetric. It is enough to examine KL divergence values only through a test author x since we like to identify d.

4 Experiments

4.1 Preliminaries

In this section we examine identification of authors to see how well an author topic model works and to what extent the models work independently of subject categories.

To do that, we examine 2 kinds of test corpus, *novels* and *editorial columns.* As the first corpus we give 3 Japanese authors, "S.Natsume", "O.Mori" and "T.Shimazaki" and their 10 novels each as shown in a table 1. We like to identify one of the 3 authors. To each novel, we give all the sentences in chapters 1 and 2 as training data and the others as a test data. As the second corpus, we give several collections of editorial columns from news articles "Asahi", "Mainichi" and "Yomiuri" in Japanese as shown in a table 2. And we do same works with these different set of training data and parameters. We give all the columns of 9 months as training data, and the others as test data in a table 2. We have applied morphological analysis in advance to all of the corpus and extracted nouns, verbs, adjectives and adverbs. Also we have removed all the infrequent words (which appeared only once).

Here we examine 2 kinds of examinations. Firstly, by using each collection of the corpus as training data and Dirichlet hyper-parameters α, β, we like to examine how well an author topic model works. To do that, we estimate topic/word distributions θ and ϕ and examine whether we can identify authorship correctly or not in each of "Novel" and "Editorial Column" domains. Finally we examine "mixture" cases. That is, we give all of "Asahi", "Mainichi", "Yomiuri","S.Natsume", "O.Mori" and "T.Shimazaki" as training data, and examine whether it is possible to identify authorship or not, and, to what extent we can expect the correctness. This examination differs from the above cases since authorship doesn't depend on specific categories.

In the following, we have examined all the experiments 10 times and and take the average values. As for Dirichlet hyper parameters we give $\alpha = 0.01$ and

Table 1. Japanese Novels

Author	Tittle
Souseki Natsume	Botchan, Garasudo no uchi, Mon, Kokoro, Kusamakura, Koujin, Michikusa, Higansugimade, Nowaki, Sanshiro
Ogai Mori	Kanoyouni, Gyogenki, Gan, Saigo no ikku, Saikikoui, Shinju, Futari no tomo, Takasebune, Kanzanjittoku, Yasuihujin
Toson Shimazaki	Arashi, Fune, Syokudo, Ganseki no aida, Chikumagawa no suketti, Shishu, Bunpai, Warazouri, Ie(first volume), Kyushujin

Table 2. The numbers of Editorial Column

	Mainichi	Asahi	Yomiuri
Learning data	515	507	513
Test data	177	171	175

Table 3. Novel Results

	Natsume	Mori	Shimazaki	Novels
(KL)				
Recall	100	94	97	97
Precision	92	100	100	97
F-value	96	97	98	97
(X^2)				
Recall	75	100	97	91
Precision	100	79	100	93
F-value	85	88	98	92

Table 4. Editorial Columns Results

	Mainichi	Asahi	Yomiuri	Columns
(KL)				
	57	89	66	71
Precision	67	71	75	71
F-value	61	79	69	71
(X^2)				
Recall	65	63	42	57
Precision	48	64	71	61
F-value	55	62	51	59

Table 5. Top 10 topics in Novels

Natsume		Mori		Shimazaki	
Topic	Probability	Topic	Probability	Topic	Probability
17	0.1015	34	0.1526	78	0.0931
30	0.0607	28	0.1036	84	0.0905
84	0.0502	45	0.0925	34	0.0864
95	0.0497	76	0.0705	61	0.0741
91	0.0492	7	0.0541	60	0.0608
54	0.0487	84	0.0541	53	0.0592
26	0.0454	17	0.0538	14	0.0588
41	0.0434	1	0.0417	62	0.0498
76	0.0336	55	0.0393	19	0.0353
31	0.0279	96	0.0364	5	0.0330

Table 6. Top 10 topics in Editorial Columns

Mainichi		Asahi		Yomiuri	
Topic	Probability	Topic	Probability	Topic	Probability
76	0.0692	29	0.0727	76	0.0724
75	0.0522	27	0.0643	54	0.0594
13	0.0465	13	0.0483	75	0.0520
40	0.0437	70	0.0427	26	0.0451
27	0.0428	53	0.0410	70	0.0382
70	0.0420	36	0.0384	72	0.0375
64	0.0415	76	0.0363	5	0.0340
72	0.0352	46	0.0342	40	0.0338
10	0.0334	47	0.0331	7	0.0331
53	0.0290	14	0.0306	53	0.0321

$\beta = 0.01$. In LDA processing we give $T = 100$ and $T = 80$ in novels and news respectively with 500 iteration during Gibbs sampling.

To evaluate our experiments, we examine F-values for author identification. F-value is defined by means of precision P_i and recall R_i. Note that, for a category i, a_i means the number of positive answers to positive items, c_i the number of negative answers to positive items and b_i the number of positive answers to negative items. F-value is defined as a harmonic mean:

$$R_i = \frac{a_i}{a_i + c_i}, \quad P_i = \frac{a_i}{a_i + b_i}, \quad F_i = \frac{2 \times P_i \times R_i}{P_i + R_i}, \quad F = Average_i F_i$$

4.2 Experimental Results

First of all, in tables 3 and 4, we show the results of author identification of the novel authors and As shown in the tables, we get 97% and 71% of F-values for novel and columns respectively by means of KL divergence. Also we get 92% and 59% by X^2 values. Let us note that in the editorial columns there is high variation values among both KL divergence and X^2 values.

We show the topics with top 10 probabilities by each author. Since we build the model based on documents by 3 authors, all the topic number have been shared. Note there is only one common topic "84" among 3 authors but there exist 7, 7 and 9 infrequent topics among 3 authors where an "infrequent" topic appears only once in a table 5. Also we show the topics with top 10 probabilities by each newspaper in a table 6. But in this case, there are 3 common topics ("76", "70", "53") and there are 2, 5 and 4 infrequent topics in newspapers respectively. Finally we examine author identification based on massive training data. We learn using a collection of training data of different kinds of subjects,

Table 7. Mixture Results

	Natsume	Mori	Shimazaki	Novels	Mainichi	Asahi	Yomiuri	Columns
(KL)								
Recall	100	93	92	95	51	81	69	67
Precision	87	100	100	96	62	74	65	67
F-value	93	96	96	95	56	77	66	67
(X^2)								
Recall	88	100	92	93	41	61	50	50
Precision	92	48	100	80	48	55	54	53
F-value	89	63	96	86	43	57	51	51

Table 8. Cross Comparison among 6 authors

	Natsume	Mori	Shimazaki	Mainichi	Asahi	Yomiuri
(KL)						
Natsume	0	3.256	3.927	13.018	11.373	13.606
Mori	5.274	0	4.653	13.442	12.591	13.688
Shimazaki	5.879	4.838	0	14.096	12.996	13.019
Mainichi	10.133	10.471	11.374	0	0.620	0.954
Asahi	10.615	10.209	11.282	0.467	0	3.173
Yomiuri	9.804	10.599	11.547	0.338	0.654	0
(X^2)						
Natsume	0	73341	56024	86067	101628	75854
Mori	6774	0	23866	54325	46076	67472
Shimazaki	17809	41944	0	137839	118245	171731
Mainichi	851838	1231037	1886942	0	141	6
Asahi	834555	1252465	1636291	16485	0	18
Yomiuri	1081328	1279498	860258	27994	322294	0

3 novel authors and 3 newspapers. Then we like to identify a test author among 6 authors. In a table 7 we show all the results of author identification.

We get 95% and 67% of F-values in Novels and Columns respectively using KL divergence, and 86% and 51% using X^2 values.

Here is the cross comparison among 6 authors in a table 8. As the readers see, there exist sharp distinction among novel authors, between novel authors and newspapers, but not among newspapers. A table 9 contains the top 10 topics in this model. There exist no common topic among all the authors nor between novels and columns. There is one common topic "12" among novels, but 5 topics ("37", "40", "66", "75", "79") among news papers.

4.3 Discussion

As for author identification, we got excellent results to novel authors (97% of F-value by KL divergence). In tables 5, we can detect few common topics, topic 84 in our experience. We show the content of topic 84 in a table 10 but no specific subject is found here and it seems to appear accidentally.

Table 9. Top 10 topics in Mixture

Natsume		Mori		Shimazaki		Mainichi		Asahi		Yomiuri	
Topic	Probability	Topic	Probability	Topic	Probability	Topic	Probability	Topic	Probability	Topic	Probability
32	0.1395	62	0.1300	87	0.2096	37	0.1398	43	0.0936	37	0.1873
24	0.0982	32	0.1158	9	0.1117	36	0.0958	92	0.0918	46	0.0827
34	0.0792	70	0.1096	52	0.0972	17	0.0705	37	0.0849	40	0.0765
9	0.0668	49	0.0925	67	0.0792	40	0.0675	75	0.0576	36	0.0652
4	0.0569	12	0.0711	98	0.0637	75	0.0559	40	0.0522	26	0.0584
91	0.0523	65	0.0625	12	0.0543	79	0.0481	66	0.0515	99	0.0450
96	0.0521	4	0.0543	27	0.0515	26	0.0416	79	0.0514	66	0.0429
31	0.0514	45	0.0529	100	0.0454	46	0.0412	17	0.0483	79	0.0402
100	0.0426	24	0.0464	25	0.0442	66	0.0401	36	0.0458	75	0.0331
12	0.0402	31	0.0306	71	0.0407	97	0.0297	15	0.0329	84	0.0287

Table 10. Topic 84 (Novel)

Word	Probability
you (*in order to*)	0.1126
iru (*doing*)	0.1085
toki (*when*)	0.0398
mono (*some*)	0.0371
nai (*none*)	0.0361
sou (*so*)	0.0338
omou (*guess*)	0.0332
gakko (*school*)	0.0227
kureru (*give*)	0.0211
ichi (*one*)	0.0179

KL divergence comes from amount of information (or entropy) and X^2 from amount of errors. Although we got difference F-values by KL divergence and X^2, we see an identical author of the best F-value in both novels and columns.

The topic distribution shows clear distinction as shown in a table 8. Also we got no common topics between novels and columns so that we can separate the two domains correctly. This result means that we can detect several "topics" reflecting word distribution specific to "authors" even if the authors come from different domains, and the model allows us to distinguish authors from others.

5 Conclusion

Here in this work, we have shown the domain independence of authorship identification within a framework of LDA. That is, we have shown empirically that an author topic model is independent of domain categories since topics capture latent dintinguishable properties among authors.

Through our experiment, we can say that an author topic model works very well, because (1) the approach allows us to detect several "topics" which reflect word distribution specific to each "author", (2) topics depend on subject categories though documents don't generally, and (3) the model works well by means of the mixture of these topics.

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
2. Griffiths, T.L., Steyvers, M.: Finding Scientific Topics. Proc. National Academy of Sciences 101 (2004)
3. Hofmann, T.: Probabilistic Latent Semantic Indexing. In: SIGIR (1999)
4. Holmes, D., Forsyth, R.: The Federalist revised: New directions in authorship attribution. Literary and Linguistic Computing 10-2, 111–127 (1995)
5. Nakayama, M., Miura, T.: Identifying Topics by using Word Distribution. In: Proc. PACRIM (2007)
6. Rosen-Zvi, M., Griffiths, Steyvers, M., Smyth, T.: The author-topic model for authors and documents. In: UAI 2004 Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (2004)

Empirical Comparison of Resampling Methods Using Genetic Fuzzy Systems for a Regression Problem

Tadeusz Lasota[1], Zbigniew Telec[2], Grzegorz Trawiński[3], and Bogdan Trawiński[2]

[1] Wrocław University of Environmental and Life Sciences, Dept. of Spatial Management
ul. Norwida 25/27, 50-375 Wrocław, Poland
[2] Wrocław University of Technology, Institute of Informatics,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
[3] Wrocław University of Technology, Faculty of Electronics,
Wybrzeże S. Wyspiańskiego 27, 50-370 Wrocław, Poland
tadeusz.lasota@up.wroc.pl, grzegorztrawinski@wp.pl,
{zbigniew.telec,bogdan.trawinski}@pwr.wroc.pl

Abstract. Much attention has been given in machine learning field to the study of numerous resampling techniques during the last fifteen years. In the paper the investigation of m-out-of-n bagging with and without replacement and repeated cross-validation using genetic fuzzy systems is presented. All experiments were conducted with real-world data derived from a cadastral system and registry of real estate transactions. The bagging ensembles created using genetic fuzzy systems revealed prediction accuracy not worse than the experts' method employed in reality. It confirms that automated valuation models can be successfully utilized to support appraisers' work.

Keywords: genetic fuzzy systems, bagging, subagging, cross-validation.

1 Introduction

Resampling techniques and ensemble models have been focusing the attention of many researchers for last fifteen years. Bagging, which stands for **b**ootstrap **agg**regat**ing**, devised by Breiman [4] belongs to the most intuitive and simplest ensemble algorithms providing a good performance. Diversity of learners is obtained by using bootstrapped replicas of the training data. That is, different training data subsets are randomly drawn with replacement from the original base dataset. So obtained training data subsets, called also bags, are used then to train different classification or regression models. Finally, individual learners are combined through algebraic expressions. The classic form of bagging is the *n-out-of-n with replacement* bootstrap where the number of samples in each bag equals to the cardinality of a base dataset and as a test set the whole original dataset is used. Much effort has been made to achieve better computational effectiveness by introducing subsampling techniques which consisted in drawing from an original dataset smaller numbers of samples, with or without replacement. The *m-out-of-n without replacement* bagging, where at each step *m* observations less than *n* are distinctly chosen at random within the base dataset, belongs to such variants. This alternative aggregation scheme was called by

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 17–24, 2011.

Bühlmann and Yu [5] subagging for **sub**sample **agg**regat**ing**. In the literature the resampling methods of the same nature as subagging are also named Monte Carlo cross-validation [24] or repeated holdout [3]. In turn, subagging with replacement was called moon-bagging, standing for **m**-**o**ut-**o**f-**n** **b**ootstrap **agg**regat**ing** [2].

The above mentioned resampling techniques are still under active theoretical and experimental investigation [2], [3], [5], [6], [11], [12], [24]. Theoretical analyses and experimental results to date proved benefits of bagging especially in terms of stability improvement and variance reduction of learners for both classification and regression problems. Bagging techniques both with and without replacement may provide improvements in prediction accuracy in a range of settings. Moreover, n-out-of-n with replacement bootstrap and n/2-out-of-n without replacement sampling, i.e. half-sampling, may give fairly similar results. Majority of the experiments were conducted employing statistical models, decision trees, and neural networks which are less computationally intensive than genetic fuzzy systems reported in the paper.

The size of bootstrapped replicas in bagging usually is equal to the number of instances in an original dataset and the base dataset is commonly used as a test set for each generated component model. However, it is claimed it leads to an optimistic overestimation of the prediction error. So, as test error out-of-bag samples are applied, i.e. those included in the base dataset but not drawn to respective bags. These, in turn may cause a pessimistic underestimation of the prediction error. In consequence, the .632 and .632+ corrections of the out-of-bag prediction error were proposed [3], [10].

The main focus of soft computing techniques to assist with real estate appraisals has been directed towards neural networks [17], [22], less researchers have been involved in the application of fuzzy systems [1], [13]. So far, we have investigated several methods to construct regression models to assist with real estate appraisal: evolutionary fuzzy systems, neural networks, decision trees, and statistical algorithms using MATLAB, KEEL, RapidMiner, and WEKA data mining systems [15], [18], [20]. We have studied also ensemble models created with these computational intelligence techniques [16], [19], [21] employing classic bagging approach.

In this paper we make one step forward, we compare m-out-of-n bagging with and without replacement with different sizes of samples with a property valuating method employed by professional appraisers in reality and the standard 10-fold cross-validation. We apply genetic fuzzy systems to real-world regression problem of predicting the prices of residential premises based on historical data of sales/purchase transactions obtained from a cadastral system. As it is often necessary to understand the behaviour of the property valuation models, we chose genetic fuzzy system as base learners because of its high interpretability compared to neural networks or support vector machines. The investigation was conducted with our newly developed system in Matlab to test multiple models using different resampling methods.

2 Methods Used and Experimental Setup

The investigation was conducted with our new experimental system implemented in Matlab environment using Fuzzy Logic, Global Optimization, Neural Network, and Statistics toolboxes [9], [14]. The system was designed to carry out research into machine learning algorithms using various resampling methods and constructing and evaluating ensemble models for regression problems.

Real-world dataset used in experiments was drawn from an unrefined dataset containing above 50 000 records referring to residential premises transactions accomplished in one Polish big city with the population of 640 000 within 11 years from 1998 to 2008. The final dataset counted the 5213 samples for which the experts could estimate the value using their pairwise comparison method. Due to the fact that the prices of premises change substantially in the course of time, the whole 11-year dataset cannot be used to create data-driven models, therefore it was split into 20 half-year subsets. The sizes of half-year data subsets are given in Table 1.

Table 1. Number of instances in half-year datasets

1998-2	**1999-1**	**1999-2**	**2000-1**	**2000-2**	**2001-1**	**2001-2**	**2002-1**	**2002-2**	**2003-1**
202	213	264	162	167	228	235	267	263	267
2003-2	**2004-1**	**2004-2**	**2005-1**	**2005-2**	**2006-1**	**2006-2**	**2007-1**	**2007-2**	**2008-1**
386	278	268	244	336	300	377	289	286	181

In order to compare evolutionary machine learning algorithms with techniques applied to property valuation we asked experts to evaluate premises using their pairwise comparison method to historical data of sales/purchase transactions recorded in a cadastral system. The experts worked out a computer program which simulated their routine work and was able to estimate the experts' prices of a great number of premises automatically.

First of all the whole area of the city was divided into 6 quality zones. Next, the premises located in each zone were classified into 243 groups determined by 5 following quantitative features selected as the main price drivers: *Area, Year, Storeys, Rooms,* and *Centre.* Domains of each feature were split into three brackets as follows:

Area denotes the usable area of premises and comprises small flats up to 40 m^2, medium flats in the bracket 40 to 60 m^2, and big flats above 60 m^2.

Year (*Age*) means the year of a building construction and consists of old buildings constructed before 1945, medium age ones built in the period 1945 to 1960, and new buildings constructed between 1960 and 1996, the buildings falling into individual ranges are treated as in bad, medium, and good physical condition respectively.

Storeys are intended for the height of a building and are composed of low houses up to three storeys, multi-family houses from 4 to 5 storeys, and tower blocks above 5 storeys.

Rooms are designated for the number of rooms in a flat including a kitchen. The data contain small flats up to 2 rooms, medium flats in the bracket 3 to 4, and big flats above 4 rooms.

Centre stands for the distance from the city centre and includes buildings located near the centre i.e. up to 1.5 km, in a medium distance from the centre - in the brackets 1.5 to 5 km, and far from the centre - above 5 km.

Then the prices of premises were updated according to the trends of the value changes over time. Starting from the second half-year of 1998 the prices were updated for the last day of consecutive half-years. The trends were modelled by polynomials of degree three. Premises estimation procedure employed a two-year time window to take into consideration transaction data of similar premises.

1. Take next premises to estimate.
2. Check the completeness of values of all five features and note a transaction date.
3. Select all premises sold earlier than the one being appraised, within current and one preceding year and assigned to the same group.
4. If there are at least three such premises calculate the average price taking the prices updated for the last day of a given half-year.
5. Return this average as the estimated value of the premises.
6. Repeat steps 1 to 5 for all premises to be appraised.
7. For all premises not satisfying the condition determined in step 4 extend the quality zones by merging 1 & 2, 3 & 4, and 5 & 6 zones. Moreover, extend the time window to include current and two preceding years.
8. Repeat steps 1 to 5 for all remaining premises.

Our study consisted in the application of an evolutionary approach to real-world regression problem of predicting the prices of residential premises based on historical data of sales/purchase transactions obtained from a cadastral system, namely genetic fuzzy systems (GFS). In GFS approach for each input variable three triangular and trapezoidal membership functions, and for output - five functions, were automatically determined by the symmetric division of the individual attribute domains. The evolutionary optimization process combined both learning the rule base and tuning the membership functions using real-coded chromosomes. Similar designs are described in [7], [8], [18]. Following resampling methods were applied in the experiments and compared with the standard 10cv and the experts' method.

Bag: B100, B70, B50, B30 – m-out-of-n bagging with replacement with different sizes of samples using the whole base dataset as a test set. The numbers in the codes indicate what percentage of the base set was drawn to create training sets.

OoB: O100, O70, O50, O30 – m-out-of-n bagging with replacement with different sizes of samples tested with the out-of-bag datasets. The numbers in the codes mean what percentage of the base dataset was drawn to create a training set.

RHO: H90, H70, H50, H30 – repeated holdout (50 times in our research), m-out-of-n bagging without replacement with different sizes of samples. The numbers in the codes point out what percentage of the base dataset was drawn to create a training set.

RCV: 1x50cv, 5x10cv, 10x5cv, 25x2cv – repeated cross-validation, k-fold cross-validation splits, for *k=50, 10, 5*, and *2* respectively, were repeated 1, 5, 10, and 25 times respectively, to obtain 50 pairs of training and test sets.

In the case of bagging methods 50 bootstrap replicates (bags) were created on the basis of each base dataset, as performance functions the mean square error (MSE) was used, and as aggregation functions simple averages were employed. The normalization of data was accomplished using the min-max approach.

3 Results of Experiments

The performance of *Bag*, *OoB*, *RHO,* and *RCV* models created by genetic fuzzy systems (GFS) in terms of MSE is illustrated graphically in Figures 1 and 2 respectively. In each figure, for comparison, the same results for *10cv* and *Expert* methods are shown. The Friedman test performed in respect of MSE values of all models built over 20 half-year datasets showed that there are significant differences between some models. Average ranks of individual models are shown in Table 2, where the lower rank value the better model. In Table 3 and 4 the results of nonparametric Wilcoxon signed-rank test to pairwise comparison of the model performance are presented. The zero hypothesis stated there were not significant differences in accuracy, in terms of MSE, between given pairs of models. In both tables + denotes that the model in the row performed significantly better than, – significantly worse than, and ≈ statistically equivalent to the one in the corresponding column, respectively. In turn, / (slashes) separate the results for individual methods. The significance level considered for the null hypothesis rejection was 5%.

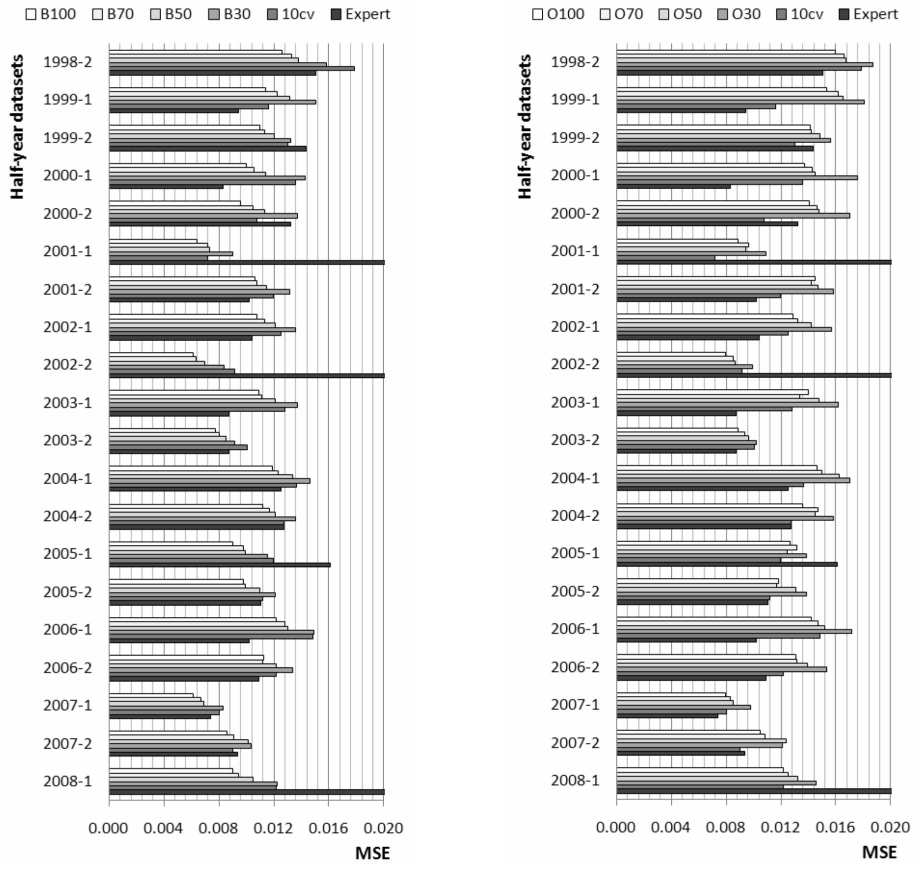


Fig. 1 Performance of *Bag* (left) and *OoB* (right) models generated using GFS

Table 2. Average rank positions of models determined during Friedman test

	1st	2nd	3rd	4th	5th	6th
Bag	B100 (1.40)	B70 (2.40)	Expert (3.50)	B50 (3.70)	10cv (4.45)	B30 (5.55)
OoB	10cv (2.20)	Expert (2.30)	O100 (2.70)	O70 (3.55)	O50 (4.50)	O30 (5.75)
RHO	H90 (2.50)	Expert 2.55)	10cv (3.05)	H70 (3.10)	H50 (4.10)	H30 (5.70)
RCV	1x50cv (2.50)	Expert (3.00)	5x10cv (3.15)	10x5cv (3.25)	10cv (3.80)	25x2cv (5.30)

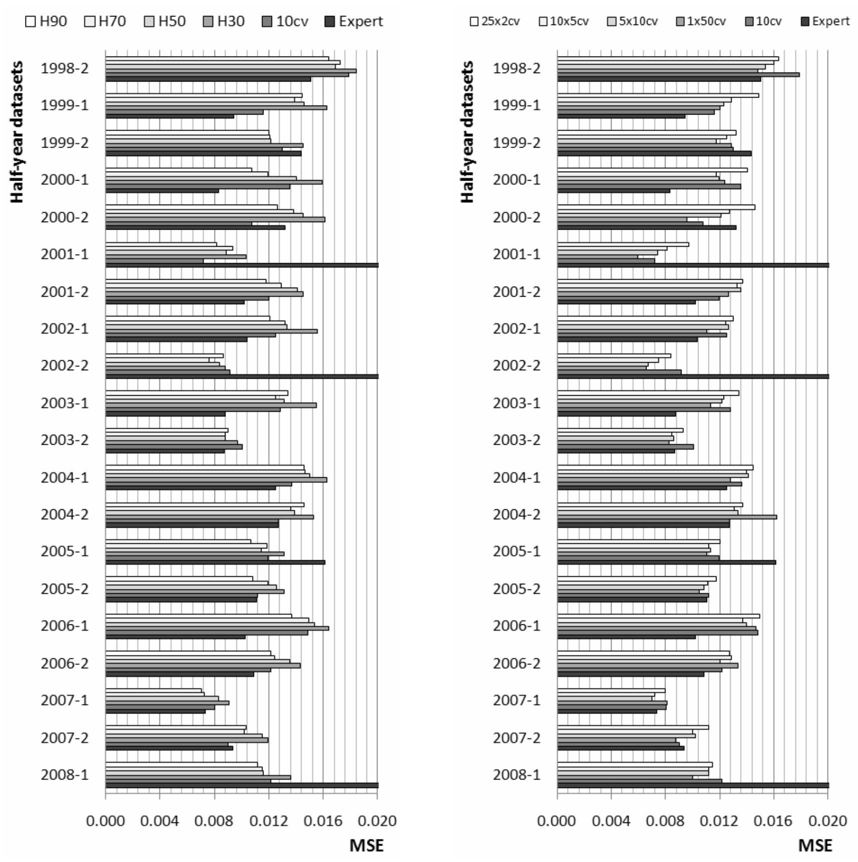


Fig. 2. Performance of *RHO* (left) and *RCV* (right) models created by GFS

Table 3. Results of Wilcoxon tests for the performance of *Bag* and *OoB* models

	B100/O100	B70/O70	B50/O50	B30/O30	10cv	Expert
B100/O100		+/ +	+ / +	+ / +	+ / –	≈ / ≈
B70/O70	– / –		+ / +	+ / +	+ / –	≈ / ≈
B50/O50	– / –	– / –		+ / +	+ / –	≈ / ≈
B30/O30	– / –	– / –	– / –		– / –	≈ / ≈
10cv	– / +	– / +	– / +	+ / +		≈ / ≈
Expert	– / ≈	≈ / ≈	≈ / ≈	≈ / ≈	≈ / ≈	

Table 4. Results of Wilcoxon tests for the performance of *RHO* and *RCV* models

	H90/1x50cv	H70/5x10cv	H50/10x5cv	H30/25x2cv	10cv	Expert
H90/1x50cv		+/ ≈	+ / +	+ / +	≈ / +	≈ / ≈
H70/5x10cv	– / ≈		+ / ≈	+ / +	≈ / ≈	≈ / ≈
H50/10x5cv	– / –	– / ≈		+ / +	– / ≈	≈ / ≈
H30/25x2cv	– / –	– / –	– / –		– / –	≈ / ≈
10cv	≈ / –	≈ / ≈	+ / ≈	+ / +		≈ / ≈
Expert	≈ / ≈	≈ / ≈	≈ / ≈	≈ / ≈	≈ / ≈	

The general outcome is as follows. Firstly, the performance of the experts' method fluctuates strongly achieving for some datasets excessively high MSE values and for others the lowest values; MSE ranges from 0.007 to 0.023. Therefore, no significant difference in accuracy between the experts' method and any other technique can be observed. Secondly, The bagging models created over 30% subsamples perform significantly worse than ones trained using bigger portions of base datasets for all methods. The same applies to 25x2cv. Thirdly, for bagging and subagging, the greater portion of a base set used as a training set the better accuracy of the models created.

More specifically, the B100, B70, and B50 bagging ensembles outperform single base models assessed using 10cv. In turn, 10cv provides better results than out-of-bag O100, O70, and O50 and repeated holdout H70 and H50 ensembles. H90 and 10cv models perform equivalently. Finally, 1x50cv turns out to be better than any other cross-validation model but one 5x10cv. No significant differences are observed among 5x10cv, 10x5cv, and 10cv.

4 Conclusions and Future Work

The computationally intensive experiments aimed to compare the performance of bagging and subagging ensembles as well as repeated cross-validation models built using genetic fuzzy systems over real-world data taken from a cadastral system with different numbers of training samples. Moreover, the predictive accuracy of a pairwise comparison method applied by professional appraisers in reality was compared with our genetic fuzzy systems aiding in a residential premises valuation.

The overall results of our investigation were as follows. The bagging ensembles created using genetic fuzzy systems revealed prediction accuracy not worse than the experts' method employed in reality. It confirms that automated valuation models can be successfully utilized to support appraisers' work.

Moreover, we conducted our experiments with the use of genetic fuzzy rule-based systems which have the advantage of knowledge extraction and representation when modeling complex systems in a way that they could be understood by humans. Processing time needed to generate models is higher when compared to other computational intelligence or statistical techniques, such as neural networks and support vector regression, but this drawback has lower impact on the effectiveness of Computer Assisted Mass Appraisal systems which may operate in off-line mode.

Acknowledgments. This paper was partially supported by the Polish National Science Centre under grant no. N N516 483840.

References

1. Bagnoli, C., Smith, H.C.: The Theory of Fuzzy Logic and its Application to Real Estate Valuation. Journal of Real Estate Research 16(2), 169–199 (1998)
2. Biau, G., Cérou, F., Guyader, A.: On the Rate of Convergence of the Bagged Nearest Neighbor Estimate. Journal of Machine Learning Research 11, 687–712 (2010)
3. Borra, S., Di Ciaccio, A.: Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. Computational Statistics & Data Analysis 54(12), 2976–2989 (2010)
4. Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)

5. Bühlmann, P., Yu, B.: Analyzing bagging. Annals of Statistics 30, 927–961 (2002)
6. Buja, A., Stuetzle, W.: Observations on bagging. Statistica Sinica 16, 323–352 (2006)
7. Cordón, O., Gomide, F., Herrera, F., Hoffmann, F., Magdalena, L.: Ten years of genetic fuzzy systems: current framework and new trends. Fuzzy Sets and Systems 141, 5–31 (2004)
8. Cordón, O., Herrera, F.: A Two-Stage Evolutionary Process for Designing TSK Fuzzy Rule-Based Systems. IEEE Tr. on Sys., Man, and Cyb.-Part B 29(6), 703–715 (1999)
9. Czuczwara, K.: Comparative analysis of selected evolutionary algorithms for optimization of neural network architectures. Master's Thesis. Wrocław University of Technology, Wrocław, Poland (2010) (in Polish)
10. Efron, B., Tibshirani, R.J.: Improvements on cross-validation: the .632+ bootstrap method. Journal of the American Statistical Association 92(438), 548–560 (1997)
11. Friedman, J.H., Hall, P.: On bagging and nonlinear estimation. Journal of Statistical Planning and Inference 137(3), 669–683 (2007)
12. Fumera, G., Roli, F., Serrau, A.: A theoretical analysis of bagging as a linear combination of classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(7), 1293–1299 (2008)
13. González, M.A.S., Formoso, C.T.: Mass appraisal with genetic fuzzy rule-based systems. Property Management 24(1), 20–30 (2006)
14. Góral, M.: Comparative analysis of selected evolutionary algorithms for optimization of fuzzy models for real estate appraisals. Master's Thesis (in Polish). Wrocław University of Technology, Wrocław, Poland (2010)
15. Graczyk, M., Lasota, T., Trawiński, B.: Comparative Analysis of Premises Valuation Models Using KEEL, RapidMiner, and WEKA. In: Nguyen, N.T., Kowalczyk, R., Chen, S.-M. (eds.) ICCCI 2009. LNCS (LNAI), vol. 5796, pp. 800–812. Springer, Heidelberg (2009)
16. Graczyk, M., Lasota, T., Trawiński, B., Trawiński, K.: Comparison of Bagging, Boosting and Stacking Ensembles Applied to Real Estate Appraisal. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) Intelligent Information and Database Systems. LNCS (LNAI), vol. 5991, pp. 340–350. Springer, Heidelberg (2010)
17. Kontrimas, V., Verikas, A.: The mass appraisal of the real estate by computational intelligence. Applied Soft Computing 11(1), 443–448 (2011)
18. Król, D., Lasota, T., Trawiński, B., Trawiński, K.: Investigation of Evolutionary Optimization Methods of TSK Fuzzy Model for Real Estate Appraisal. International Journal of Hybrid Intelligent Systems 5(3), 111–128 (2008)
19. Krzystanek, M., Lasota, T., Telec, Z., Trawiński, B.: Analysis of Bagging Ensembles of Fuzzy Models for Premises Valuation. In: Nguyen, N.T., Le, M.T., Świątek, J. (eds.) Intelligent Information and Database Systems. LNCS (LNAI), vol. 5991, pp. 330–339. Springer, Heidelberg (2010)
20. Lasota, T., Mazurkiewicz, J., Trawiński, B., Trawiński, K.: Comparison of Data Driven Models for the Validation of Residential Premises using KEEL. International Journal of Hybrid Intelligent Systems 7(1), 3–16 (2010)
21. Lasota, T., Telec, Z., Trawiński, B., Trawiński, K.: Exploration of Bagging Ensembles Comprising Genetic Fuzzy Models to Assist with Real Estate Appraisals. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 554–561. Springer, Heidelberg (2009)
22. Lewis, O.M., Ware, J.A., Jenkins, D.: A novel neural network technique for the valuation of residential property. Neural Computing & Applications 5(4), 224–229 (1997)
23. Martínez-Muñoz, G., Suárez, A.: Out-of-bag estimation of the optimal sample size in bagging. Pattern Recognition 43, 143–152 (2010)
24. Molinaro, A.N., Simon, R., Pfeiffer, R.M.: Prediction error estimation: a comparison of resampling methods. Bioinformatics 21(15), 3301–3307 (2005)

Linear Time Heuristics for Topographic Mapping of Dissimilarity Data

Andrej Gisbrecht, Frank-Michael Schleif, Xibin Zhu, and Barbara Hammer

CITEC Centre of Excellence, Bielefeld University, 33615 Bielefeld - Germany
bhammer@techfak.uni-bielefeld.de

Abstract. Topographic mapping offers an intuitive interface to inspect large quantities of electronic data. Recently, it has been extended to data described by general dissimilarities rather than Euclidean vectors. Unlike its Euclidean counterpart, the technique has quadratic time complexity due to the underlying quadratic dissimilarity matrix. Thus, it is infeasible already for medium sized data sets. We introduce two approximation techniques which speed up the complexity to linear time algorithms: the Nyström approximation and patch processing, respectively. We evaluate the techniques on three examples from the biomedical domain.

1 Introduction

In many application areas such as bioinformatics, technical systems, or the web, electronic data sets are increasing rapidly with respect to size and complexity. Automated analysis tools offer indispensable techniques to extract relevant information from these data. Popular approaches provide diverse techniques for data structuring and data inspection. Visualization or clustering still constitute one of the most common tasks in this context. Topographic mapping such as offered by the self-organizing map (SOM) [12] and its statistic counterpart, the generative topographic mapping (GTM) [3] provide simultaneous clustering, data visualization, compression by means of prototypes, and inference of the topographic structure of the data manifold in one intuitive framework. For this reason, topographic mapping constitutes a popular tool in diverse areas ranging from remote sensing or biomedical domains up to robotics or telecommunication [12].

Like many classical machine learning techniques, SOM and GTM have been proposed for Euclidean vectorial data. Modern data are often associated to dedicated structures which make a representation in terms of Euclidean vectors difficult: biological sequence data, text files, XML data, trees, graphs, or time series, for example. These data are inherently compositional and a feature representation leads to information loss. As an alternative, a dedicated dissimilarity measure such as pairwise alignment, or kernels for trees or graph can be used as the interface to the data. In such cases, machine learning techniques which can deal with pairwise similarities or dissimilarities have to be used.

Quite a few extensions of topographic mapping towards pairwise similarities or dissimilarities have been proposed in the literature. Some are based on a kernelization of existing approaches [4,18], while others restrict the setting to exemplar

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 25–33, 2011.

based techniques [5,13]. Some techniques built on alternative cost functions and advanced optimization methods [16,9]. A very intuitive method which directly extends prototype based clustering to dissimilarity data has been proposed in the context of fuzzy clustering [11] and later been extended to topographic mapping such as SOM and GTM [10,8]. Due to its direct correspondence to standard topographic mapping in the Euclidean case, we will focus on the latter techniques. Further, we restrict to the GTM because of its excellent visualization capabilities and its foundation as a stochastic model.

One drawback of machine learning techniques for dissimilarities is given by their high computational costs: since they depend on the full (quadratic) dissimilarity matrix, they have squared time complexity; further, they require the availability of the full dissimilarity matrix, which is even the more severe bottleneck if complex dissimilarities such as e.g. alignment techniques are used. This fact makes the methods unsuitable already for medium sized data sets.

Here, we propose two different approximations to speed up GTM for dissimilarities: the Nyström approximation has been proposed in the context of kernel methods as a low rank approximation of the matrix [17]. In [7], preliminary work extends these results to dissimilarities. In this contribution, we demonstrate that the technique provides a suitable linear time approximation for GTM for dissimilarities. As an alternative, patch processing has been proposed in the context of topographic mapping of Euclidean data [1] and later been extended to clustering of dissimilarities [10]. Here we transfer the technique to GTM for dissimilarities, resulting in a linear time method which is even suited if data are not i.i.d. i.e. a representative subpart of the matrix is not accessible priorly.

2 Relational Topographic Mapping

The GTM has been proposed in [3] as a probabilistic counterpart to SOM. It models given data $\mathbf{x} \in \mathbb{R}^D$ by a constraint mixture of Gaussians induced by a low dimensional latent space. More precisely, regular lattice points $\mathbf{w}$ are fixed in latent space and mapped to target vectors $\mathbf{w} \mapsto \mathbf{t} = y(\mathbf{w}, \mathbf{W})$ in the data space, where the function y is typically chosen as generalized linear regression model $y : \mathbf{w} \mapsto \Phi(\mathbf{w}) \cdot \mathbf{W}$ induced by base functions Φ such as equally spaced Gaussians with bandwidth σ. Every latent point induces a Gaussian

$$p(\mathbf{x}|\mathbf{w}, \mathbf{W}, \beta) = \left(\frac{\beta}{2\pi}\right)^{D/2} \exp\left(-\frac{\beta}{2}\|\mathbf{x} - y(\mathbf{w}, \mathbf{W})\|^2\right) \tag{1}$$

A mixture of K modes $p(\mathbf{x}|\mathbf{W}, \beta) = \sum_{k=1}^{K} \frac{1}{K} p(\mathbf{x}|\mathbf{w}_k, \mathbf{W}, \beta)$ is generated. GTM training optimizes the data log-likelihood with respect to $\mathbf{W}$ and β. This can be done by an EM approach, iteratively computing responsibilities

$$R_{kn}(\mathbf{W}, \beta) = p(\mathbf{w}_k|\mathbf{x}_n, \mathbf{W}, \beta) = \frac{p(\mathbf{x}_n|\mathbf{w}_k, \mathbf{W}, \beta)p(\mathbf{w}_k)}{\sum_{k'} p(\mathbf{x}_n|\mathbf{w}_{k'}, \mathbf{W}, \beta)p(\mathbf{w}_{k'})} \tag{2}$$

of component k for point number n, and optimizing model parameters by means of the formulas

$$\mathbf{\Phi}^T \mathbf{G}_{\text{old}} \mathbf{\Phi} \mathbf{W}_{\text{new}}^T = \mathbf{\Phi}^T \mathbf{R}_{\text{old}} \mathbf{X} \tag{3}$$

for $\mathbf{W}$, where $\mathbf{\Phi}$ refers to the matrix of base functions Φ evaluated at points $\mathbf{w}_k$, $\mathbf{X}$ to the data points, $\mathbf{R}$ to the responsibilities, and $\mathbf{G}$ is a diagonal matrix with accumulated responsibilities $G_{nn} = \sum_n R_{kn}(\mathbf{W}, \beta)$. The bandwidth is given by

$$\frac{1}{\beta_{\text{new}}} = \frac{1}{ND} \sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \|\Phi(\mathbf{w}_k)\mathbf{W}_{\text{new}} - \mathbf{x}_n\|^2 \tag{4}$$

where D is the data dimensionality and N the number of points. GTM is initialized by aligning the lattice image and the the first two data principal components.

GTM has been extended to general dissimilarities in [8]. We assume that data $\mathbf{x}$ are given by pairwise dissimilarities $d_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|^2$ with corresponding dissimilarity matrix D, where the vector representation $\mathbf{x}$ of the data is unknown and $\|\cdot\|^2$ can be induced by any symmetric bilinear form. As pointed out in [11,10], if prototypes are restricted to linear combinations of the form $\mathbf{t}_k = \sum_{n=1}^{N} \alpha_{kn}\mathbf{x}_n$ with $\sum_{n=1}^{N} \alpha_{kn} = 1$, the prototypes $\mathbf{t}_k$ can be represented indirectly by means of the coefficients $\boldsymbol{\alpha}_k$ and distances can be computed by

$$\|\mathbf{x}_n - \mathbf{t}_k\|^2 = [\mathbf{D}\boldsymbol{\alpha}_k]_n - \frac{1}{2} \cdot \boldsymbol{\alpha}_k^T \mathbf{D} \boldsymbol{\alpha}_k \tag{5}$$

This constitutes the key observation to transfer GTM to relational data $\mathbf{D}$.

As before, targets $\mathbf{t}_k$ induce a Gaussian mixture distribution in the data space. They are obtained as images of points $\mathbf{w}$ in latent space via a generalized linear regression model where, now, the mapping is to the coefficients $y : \mathbf{w}_k \mapsto \boldsymbol{\alpha}_k = \Phi(\mathbf{w}_k) \cdot \mathbf{W}$ with $\mathbf{W} \in \mathbb{R}^{d \times N}$. The restriction $\sum_n [\Phi(\mathbf{w}_k) \cdot \mathbf{W}]_n = 1$ is automatically fulfilled for optima of the data log likelihood. Hence the likelihood function can be computed based on (1) and the distance computation can be performed indirectly using (5). An EM optimization scheme leads to solutions for the parameters β and $\mathbf{W}$, and an expression for the hidden variables given by the responsibilities of the modes for the data points. Algorithmically, Eqn. (2) using (5) and the optimization of the expectation $\sum_{k,n} R_{kn}(\mathbf{W}_{\text{old}}, \beta_{\text{old}}) \ln p(\mathbf{x}_n | \mathbf{w}_k, \mathbf{W}_{\text{new}}, \beta_{\text{new}})$ with respect to $\mathbf{W}$ and β take place in turn. The latter yields model parameters which can be determined in analogy to (3,4) where, now, functions Φ map from the latent space to the space of coefficients α and $\mathbf{X}$ denotes the unity matrix in the space of coefficients. We refer to this iterative update scheme as relational GTM (RGTM). Initialization takes place by referring to the first MDS directions of $\mathbf{D}$.

3 The Nyström Approximation

We shortly review the Nyström technique as presented in [17]. By the Mercer theorem kernels $k(\mathbf{x}, \mathbf{y})$ can be expanded by orthonormal eigenfunctions ϕ_i and non negative eigenvalues λ_i in the form $k(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{y})$. The eigenfunctions and eigenvalues of a kernel are the solution of $\int k(\mathbf{y}, \mathbf{x}) \phi_i(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} = \lambda_i \phi_i(\mathbf{y})$, which can be approximated based on the Nyström technique by sampling $\mathbf{x}_k$ i.i.d. according to p: $\frac{1}{m} \sum_{k=1}^{m} k(\mathbf{y}, \mathbf{x}_k) \phi_i(\mathbf{x}_k) \approx \lambda_i \phi_i(\mathbf{y})$. Using the matrix eigenproblem $\mathbf{K}^{(m)}\mathbf{U}^{(m)} = \mathbf{U}^{(m)}\mathbf{\Lambda}^{(m)}$ of the $m \times m$ Gram matrix $\mathbf{K}^{(m)}$ we can derive the approximations for the eigenfunctions and eigenvalues

$$\lambda_i \approx \frac{\lambda_i^{(m)}}{m}, \quad \phi_i(\mathbf{y}) \approx \frac{\sqrt{m}}{\lambda_i^{(m)}}\mathbf{k}_y\mathbf{u}_i^{(m)}, \tag{6}$$

where $\mathbf{u}_i^{(m)}$ is the ith column of $\mathbf{U}^{(m)}$. Thus, we can approximate ϕ_i at an arbitrary point $\mathbf{y}$ as long as we know the vector $\mathbf{k}_y = (k(\mathbf{x}_1, \mathbf{y}), ..., k(\mathbf{x}_m, \mathbf{y}))^T$.

One well known way to approximate a $n \times n$ Gram matrix, is to use a low-rank approximation. This can be done by computing the eigendecomposition of the kernel $\mathbf{K} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$, where $\mathbf{U}$ is orthonormal and $\mathbf{\Lambda}$ is diagonal with $\mathbf{\Lambda}_{11} \geq \mathbf{\Lambda}_{22} \geq ... \geq 0$, and keeping only the m eigenspaces which correspond to the m largest eigenvalues of the matrix. The approximation is $\mathbf{K} \approx \mathbf{U}_{n,m}\mathbf{\Lambda}_{m,m}\mathbf{U}_{m,n}$, where the indices refer to the size of the corresponding submatrix. The Nyström method can approximate a kernel in a similar way, without computing the eigendecomposition of the whole matrix, which is an $O(n^3)$ operation. For a given $n \times n$ Gram matrix $\mathbf{K}$ we randomly choose m rows and respective columns. We denote these rows by $\mathbf{K}_{m,n}$. Using the formulas (6) we obtain $\tilde{\mathbf{K}} = \sum_{i=1}^m 1/\lambda_i^{(m)} \cdot \mathbf{K}_{m,n}^T\mathbf{u}_i^{(m)}(\mathbf{u}_i^{(m)})^T\mathbf{K}_{m,n}$, where $\lambda_i^{(m)}$ and $\mathbf{u}_i^{(m)}$ correspond to the $m \times m$ eigenproblem. Thus we get, $\mathbf{K}_{m,m}^{-1}$ denoting the pseudoinverse,

$$\tilde{\mathbf{K}} = \mathbf{K}_{m,n}^T\mathbf{K}_{m,m}^{-1}\mathbf{K}_{m,n}. \tag{7}$$

This approximation is exact, if $K_{m,m}$ has the same rank as K.

For dissimilarity data, a direct transfer is possible, see [7] for preliminary work on this topic. A symmetric dissimilarity matrix $\mathbf{D}$ is a normal matrix and according to the spectral theorem can be diagonalized $\mathbf{D} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ with $\mathbf{U}$ being a unitary matrix whose column vectors are the orthonormal eigenvectors of $\mathbf{D}$ and $\mathbf{\Lambda}$ a diagonal matrix with the eigenvalues of $\mathbf{D}$, which can be negative for non-Euclidean distances. Therefore the dissimilarity matrix can be seen as an operator $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^N \lambda_i\phi_i(\mathbf{x})\phi_i(\mathbf{y})$ where $\lambda_i \in \mathbb{R}$ correspond to the diagonal elements of $\mathbf{\Lambda}$ and ϕ_i denote the eigenfunctions. The only difference to an expansion of a kernel is that the eigenvalues can be negative. All further mathematical manipulations can be applied in the same way.

Using the approximation (7) for the distance matrix, we can apply this result for RGTM. It allows to approximate (5) in the way

$$\|\mathbf{x}_n - \mathbf{t}_k\|^2 \approx \left[\mathbf{D}_{m,n}^T\left(\mathbf{D}_{m,m}^{-1}\left(\mathbf{D}_{m,n}\boldsymbol{\alpha}_k\right)\right)\right]_n - \frac{1}{2} \cdot \left(\boldsymbol{\alpha}_k^T\mathbf{D}_{m,n}^T\right) \cdot \left(\mathbf{D}_{m,m}^{-1}\left(\mathbf{D}_{m,n}\boldsymbol{\alpha}_k\right)\right) \tag{8}$$

with a linear submatrix of m rows and a low rank matrix $\mathbf{D}_{m,m}$ corresponding to the eigenproblem. This computation is $\mathcal{O}(m^2n)$ instead of $\mathcal{O}(n^2)$, i.e. it is linear in the number of data points n, assuming fixed approximation m. The last statement holds for constant data space complexity, by means of the eigenproblem and has to be adapted otherwise.

A benefit of the Nyström technique is that it can be decided priorly which linear parts of the dissimilarity matrix will be used in training. A drawback is that a good approximation can only be achieved if the rank of $\mathbf{D}_{m,m}$ is close to the rank of $\mathbf{D}$ as much as possible, i.e. the chosen subset should be representative.

4 Patch Processing

Patch processing takes a different perspective and processes data consecutively in patches of small size m. It has been proposed in [10] in the context of clustering dissimilarity data. Here we present an extension to RGTM.

The principled idea is to compress all already seen data by means of the prototypes as found by RGTM. These prototypes are taken as additional inputs in the next step in the same way as 'standard' points. Since they compress several data points, they are counted with multiplicities according to the size of their receptive fields. This way, eventually, all data points are processed.

Two extensions are necessary to apply this scheme: we need an efficient realization of RGTM if some data are contained in the training set more than once, i.e. data point $\boldsymbol{x}_i$ comes with multiplicity m_i. Further, since prototypes in RGTM are represented only implicitly by means of coefficient vectors, an efficient approximation of prototype $\mathbf{t}_j$ by means of a priorly fixed number of data points needs to be chosen. Both issues can be dealt with:

- *Extension of RGTM to multiple data points:* Multiple data points affect Eqns. 3, 4. In Eqn. 3, the matrices $\mathbf{G}$ and $\mathbf{R}$ need to weight the responsibilities according to the multiplicities of the data. In Eqn. 4, the summands are weighted by the multiplicities and N is changed accordingly. Similarly, the MDS initialization of RGTM can be extended to multiplicities.
- *Approximation of prototypes by a finite number of points:* fixing the quality k of the approximation, we represent a prototype $\mathbf{t}_j$ by its k closest data points $\mathbf{x}_i$. The union of these data points is taken and every data point is weighted according to the sum of multiplicities of its receptive field.

```
init:
        $E := \emptyset$, number of patch $p := 1$
repeat:
        read patch of size $m$, i.e. $P_{m,m} := \{d_{ij} \mid p \cdot m < i, j \leq (p+1) \cdot m\}$
        compute dissimilarities of patch and $E$,
                                i.e. $P_{m,|E|} := \{d_{i,j} \mid p \cdot m < i \leq (p+1) \cdot m, \mathbf{x}_j \in E\}$
        compute dissimilarities in $E$, i.e. $P_{|E|,|E|} := \{d_{ij} \mid \mathbf{x}_i, \mathbf{x}_j \in E\}$
        this gives the matrix $P := \begin{pmatrix} P_{m,m} & P_{m,|E|} \\ P^t_{m,|E|} & P_{|E|,|E|} \end{pmatrix}$
        set the multiplicities to $m_i := \begin{cases} 1 & \text{if } p < i \leq (p+1)m \\ m_i & \text{if } (\mathbf{x}_i, m_i) \in E \end{cases}$
        perform RGTM with multiplicities for $P$
        set $E :=$ union of the $k$ nearest data points in $P$ for every prototype $\mathbf{t}_j$
                                with multiplicities according to their receptive fields
```

Fig. 1. Principled algorithm for patch RGTM

The algorithm of patch RGTM is displayed in Fig. 1. Since all data are taken into account either directly in the current patch or indirectly represented by the prototypes, processing of data sets in non i.i.d. order is possible. Since it becomes apparent only during training which parts of the dissimilarity matrix are used for training, it is required to compute dissimilarities during training on demand. Only a linear subpart of the dissimilarity corresponding to the size m needs to be considered, hence the algorithm is $\mathcal{O}(m^2n)$ instead of $\mathcal{O}(n^2)$.

5 Experiments

We evaluate the techniques on three benchmarks from the biomedical domain:

- The *Copenhagen Chromosomes data set* constitutes a benchmark from cytogenetics [14]. A set of 4200 human chromosomes from 22 classes (the autosomal chromosomes) are represented by grey-valued images. These are transferred to strings measuring the thickness of their silhouettes. These strings are compared using edit distance with insertion/deletion costs 4.5.
- The *vibrio data set* consists of 1100 samples of vibrio bacteria populations characterized by mass spectra. The spectra encounter approx. 42000 mass positions. The full data set consists of 49 classes of vibrio-sub-species. The mass spectra are preprocessed with a standard workflow using the BioTyper software [15]. According to the functional form of mass spectra, dedicated similarities as provided by the BioTyper software are used [15].
- Similar to an application presented in [13], we extract roughly 11000 protein sequences of the *SwissProt data base* according to 32 functional labels given by PROSITE [6]. Sequence alignment is done using FASTA [19].

For the chromosomes and vibrio data sets, we use 20 × 20 prototypes, 10 × 10 base functions with bandwidth 1, and 50 epochs for training. For patch RGTM, we use 10 patches with a k-approximation with $k \in \{1, 3, 5\}$. For the Nyström approximation, two different fractions of landmarks are tested.

The results of a ten-fold cross-validation with ten repeats are reported in the Tables 1 and 2. The classification accuracy is evaluated using posterior labeling

Table 1. Results of the methods on the Chromosome data set, standard deviation and speed up are given in parentheses

Chromosome	Classification accuracy	Streaming data	CPU time in sec
RGTM	0.916 (0.003)		2650
RGTM (Nyström 0.01)	0.878 (0.022)	0.626(0.164)	394 (6.7)
RGTM (Nyström 0.1)	0.552 (0.065)	0.365(0.210)	619 (4.3)
Patch RGTM (k=1)	0.845 (0.005)	0.737 (0.023)	318 (8.3)
Patch RGTM (k=3)	0.851 (0.003)	0.777 (0.013)	523 (5.1)
Patch RGTM (k=5)	0.867 (0.004)	0.804 (0.013)	615 (4.3)

Table 2. Results on the Vibrio data set reporting standard deviation and speed up in parentheses

Vibrio	Classification accuracy	Streaming data	CPU time in sec
RGTM	0.947 (0.005)		78
RGTM (Nyström 0.05)	0.927 (0.005)	0.652 (0.043)	32 (2.4)
RGTM (Nyström 0.1)	0.937 (0.010)	0.590 (0.053)	36 (2.2)
Patch RGTM (k=1)	0.677 (0.020)	0.421 (0.048)	77 (1)
Patch RGTM (k=3)	0.833 (0.012)	0.592 (0.043)	107 (0.7)
Patch RGTM (k=5)	0.889 (0.010)	0.648 (0.044)	149 (0.5)

of the prototypes, the standard deviation is given in parenthesis. Further, the CPU time in seconds is reported, the relative speed up as compared to the (not accelerated) RGTM is given in parenthesis. We test the robustness of the techniques with respect to non i.i.d. data by sorting data according to the given class labeling, with only 30 percent random sampling (referred to as 'streaming data'), versus standard random ordering.

Interestingly, the Nyström technique as well as patch processing lead to improved speed (up to a factor 8) on the Chromosome data already for this comparably small data set. The classification accuracy for this data set is only slightly reduced (by less than 5%) for appropriate settings. Obviously, the Nyström technique requires representative sampling while patch processing is more robust against non i.i.d. ordering of data. For the Vibrio data set, no speed up can be achieved using patch processing and the results are massively worse in this case probably due to the fact that a compression of data by few prototypes is not adequately possible. In contrast, the Nyström approximation seems well suited.

For the SwissProt data set we used 40 × 40 prototypes and bandwidth 0.2. Patch RGTM is done with 11 patches. The results of a ten-fold cross-validation with five repeats (only one repeat for RGTM) and the CPU time required to train the map once for the full data set are reported in Table 3. Apparently, the Nyström approximation does not deteriorate the accuracy of the map, while patch processing is not suited due to the incompressibility of the data by few prototypes.

This data set is of medium size, such that the speed up of the Nyström approximation becomes apparent; it accounts for a factor almost 10. Interestingly, also

Table 3. Results on the SwissProt data set; standard deviation, speed up in parentheses

SwissProt	Classification Accuracy	CPU time in sec
RGTM	0.596	53135
RGTM (Nyström 0.009)	0.630 (0.017)	5892 (9)
Patch RGTM (k=5)	0.388 (0.006)	18623 (2.85)

the required memory is widely reduced: in the given example, assuming double precision, about 500 Megabyte are necessary to store the full dissimilarity matrix as compared to about 4.5 Megabyte for the dissimilarities referred to by the Nyström approximation. Using the same number of landmarks and assuming a standard RAM of 12 Gigabyte, this technique would allow to store the required dissimilarities of almost 30 million data points when using the Nyström approximation as compared to only 30 thousand data if the full dissimilarity matrix is required. The speed-up would be in the same order of magnitude due to the dominating factor required to compute the pairwise dissimilarities. Extensions of the technique to a larger fraction of the data set are the subject of ongoing work.

6 Conclusions

Relational GTM offers a highly flexible tool to simultaneously cluster and order dissimilarity data in a topographic mapping. Due to the dependency on the full matrix, the method requires squared time complexity and memory to store the dissimilarities. We have proposed two speed-up techniques which both lead to linear effort: patch processing and the Nyström approximation. Using three examples from the biomedical domain, we demonstrated that already for comparably small data sets the techniques can greatly improve speed while not losing too much information contained in the data.

Acknowledgment. This work was supported by the "German Science Foundation (DFG)" under grant number HA-2719/4-1. Further, financial support from the Cluster of Excellence 277 Cognitive Interaction Technology funded in the framework of the German Excellence Initiative is gratefully acknowledged.

References

1. Alex, N., Hasenfuss, A., Hammer, B.: Patch clustering for massive data sets. Neurocomputing 72(7-9), 1455–1469 (2009)
2. Barbuddhe, S.B., Maier, T., Schwarz, G., Kostrzewa, M., Hof, H., Domann, E., Chakraborty, T., Hain, T.: Rapid identification and typing of listeria species by matrix-assisted laser desorption ionization-time of flight mass spectrometry. Applied and Environmental Microbiology 74(17), 5402–5407 (2008)
3. Bishop, C., Svensen, M., Williams, C.: The generative topographic mapping. Neural Computation 10(1), 215–234 (1998)
4. Boulet, R., Jouve, B., Rossi, F., Villa-Vialaneix, N.: Batch kernel SOM and related Laplacian methods for social network analysis. Neurocomputing 71(7-9), 1257–1273 (2008)
5. Cottrell, M., Hammer, B., Hasenfuss, A., Villmann, T.: Batch and median neural gas. Neural Networks 19, 762–771 (2006)
6. Gasteiger, E., Gattiker, A., Hoogland, C., Ivanyi, I., Appel, R.D., Bairoch, A.: ExPASy: the proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Res. 31, 3784–3788 (2003)

7. Gisbrecht, A., Mokbel, B., Hammer, B.: The Nystrom approximation for relational generative topographic mappings. In: NIPS Workshop on Challenges of Data Visualization (2010)
8. Gisbrecht, A., Mokbel, B., Hammer, B.: Relational Generative Topographic Mapping. Neurocomputing 74, 1359–1371 (2011)
9. Graepel, T., Obermayer, K.: A stochastic self-organizing map for proximity data. Neural Computation 11, 139–155 (1999)
10. Hammer, B., Hasenfuss, A.: Topographic Mapping of Large Dissimilarity Data Sets. Neural Computation 22(9), 2229–2284 (2010)
11. Hathaway, R.J., Bezdek, J.C.: Nerf c-means: Non-Euclidean relational fuzzy clustering. Pattern Recognition 27(3), 429–437 (1994)
12. Kohonen, T. (ed.): Self-Organizing Maps, 3rd edn. Springer-Verlag New York, Inc., Secaucus (2001)
13. Kohonen, T., Somervuo, P.: How to make large self-organizing maps for nonvectorial data. Neural Networks 15, 945–952 (2002)
14. Lundsteen, C., Phillip, J., Granum, E.: Quantitative analysis of 6985 digitized trypsin G-banded human metaphase chromosomes. Clinical Genetics 18, 355–370 (1980)
15. Maier, T., Klebel, S., Renner, U., Kostrzewa, M.: Fast and reliable MALDI-TOF ms–based microorganism identification. Nature Methods 3 (2006)
16. Seo, S., Obermayer, K.: Self-organizing maps and clustering methods for matrix data. Neural Networks 17, 1211–1230 (2004)
17. Williams, C.K.I., Seeger, M.: Using the Nyström method to speed up kernel machines. In: Advances in Neural Information Processing Systems, vol. 13, pp. 682–688 (2001)
18. Yin, H.: On the equivalence between kernel self-organising maps and self-organising mixture density networks. Neural Networks 19(6-7), 780–784 (2006)
19. Lipman, D.J., Pearson, W.R.: Rapid and sensitive protein similarity searches. Science, 227, 1435–1441

Interpreting Hidden Neurons in Boolean Constructive Neural Networks

Maria do Carmo Nicoletti[1,3,*], João R. Bertini Jr.[2], and Osvaldo Luiz de Oliveira[3]

[1] CS Dept, UFSCar, S. Carlos, SP, Brazil
[2] ICMC, USP, S. Carlos, SP, Brazil
[3] FACCAMP, Campo Limpo Paulista, SP, Brazil
carmo@dc.ufscar.br

Abstract. A particular group of neural network (NN) learning algorithms known as constructive algorithms (CoNN) congregates algorithms that dynamically combine two processes: (1) the definition of the NN architecture and (2) learning. Generally both processes alternate, depending on each others´ performance. During training CoNN algorithms incrementally add hidden neurons and connections to the network until some stopping criterion is satisfied. This paper describes an investigation into the semantic role played by the hidden neurons added into the NN, when learning Boolean functions. Five CoNN algorithms namely Tower, Pyramid, Tiling, Perceptron-Cascade and Shift are examined in that respect. Results show that hidden neurons represent Boolean sub-expressions whose combination represents a disjunction of prime implicants.

Keywords: interpreting hidden neurons, constructive neural network algorithms, Tower, Pyramid, Tiling, Perceptron-Cascade, Shift.

1 Introduction

While conventional neural network algorithms require the specification of the NN architecture before training begins, constructive neural network (CoNN) algorithms dynamically construct the neural network architecture along with (and as a consequence of) the training process. Considering that the pre-definition of a suitable NN architecture for a certain problem can be a hard task, CoNN algorithms can be a very convenient option when compared to conventional algorithms in that respect.

CoNN algorithms typically start with a network having the input layer and no hidden layers; the dynamic construction of the network's hidden layer(s) occurs simultaneously with training. The CoNN algorithms found in the literature differ from each other in several aspects, such as the number of nodes they add per layer, the direction they grow the network (from input to output neurons or vice-versa), the functionality of the added neurons (do they all perform the same role?), the stopping criteria, the connectivity patterns of the newly added neuron, the algorithm used for training individual neuron (generally the Pocket or the PRM (Pocket with Ratchet

[*] Corresponding author.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 34–41, 2011.

Modification) [1]), the type of input patterns they are able to deal with (binary (or bipolar) valued only or real valued attributes), problems they solve (classification (two or multi-class) or regression) and so on [2]. The description of a few well-known CoNN algorithms can be found in [1], [3] and [2]. Franco and co-workers in [4] gathered a collection of CoNN algorithms including several new proposals.

The investigation of possibilities for symbolically interpreting conventional neural network knowledge has given rise to several algorithms and techniques, as can be seen in various works, e.g. [5], [6], [7], [8] and [9]. The group of algorithms especialised in dealing with the symbolic aspect of NNs can be further divided, as described in [10], into (1) algorithms designed to insert knowledge into NNs (knowledge initialization); (2) algorithms for extracting rules from NNs (rule extraction) and (3) algorithms that use NNs to refine existing rule bases (rule refinement). Particularly, algorithms for extracting rules from NNs have received a great deal of attention and are more popular than the others due to the demand, in critical knowledge domains, for a comprehensible justification of the reasons a NN reached a certain conclusion. As stressed in [6], “The explanation capability of neural networks can be achieved by the extraction of symbolic knowledge”.

Although the work described in this paper intends to search for symbolic interpretations of hidden neurons of Boolean constructive neural networks aiming at the symbolic interpretation of the whole NN, the motivation for interpreting hidden nodes is supported by the idea that such neurons could represent some sort of pre-existing concept associated to the Boolean expression represented by the NN. In this sense, the hidden neurons added to the architecture of a NN during learning could be approached as a process of feature construction [11].

The remainder of the paper is organized as follows: Section 2 presents the main characteristics of five two-class CoNN algorithms used in the experiments: Tower, Pyramid [1], Tiling [12], Perceptron-Cascade [13] and Shift [14]. Although they all grow neural networks, each algorithm has particularities of its own. Section 3 describes the experiments conducted to identify the symbolic interpretation of hidden neurons created (and trained) during the constructive process carried out by the algorithms; results are compared and discussed. In Section 4 the main ideas described in the paper are summarized and some conclusions of the work are highlighted.

2 A Brief Description of CoNN Algorithms Tower, Pyramid, Tiling, Perceptron-Cascade and Shift

The basic task performed by CoNN algorithms consists in inserting an individual neuron, generally a threshold logic unit (TLU), into the NN being constructed and immediately training it. In the literature there are many TLU training algorithms available; the PRM has been chosen for the empirical experiments described in Section 3 of this paper due to its good results found in the literature (see [1]).

The choice of Tower and Pyramid algorithms as CoNN algorithms for investigating the interpretation of hidden neurons has been motivated by the particular way the two algorithms carry on the constructive process. Tower iteratively grows a neural network by adding hidden layers where each new hidden layer has only one TLU that is connected all input neurons as well as to the only neuron that defines the previous hidden layer added to the NN. With the dynamic addition of neurons the algorithm tends to correctly classify a greater number of training instances. The Pyramid algorithm also adds one single neuron per layer while growing the NN. It only differs from Tower in relation to the architecture of connections among neurons. While in a Tower-NN each new hidden neuron has connections with every input neuron as well as with the last hidden neuron created, in a Pyramid-NN a new hidden neuron has connections with every input neuron as well as with all the previous hidden neurons created.

The choice of the Tiling algorithm was motivated by the very particular way the algorithm constructs the NN. Each hidden layer has one *master neuron* that works as the output neuron for that layer. If the master neuron is unable to correctly classify all training patterns, Tiling adds TLUs (one at a time) to the layer (the so called *ancillary neurons*), aiming at obtaining a faithful representation of the training set. The output layer has only the master neuron. Figure 1 shows the architecture of a Tiling-NN.

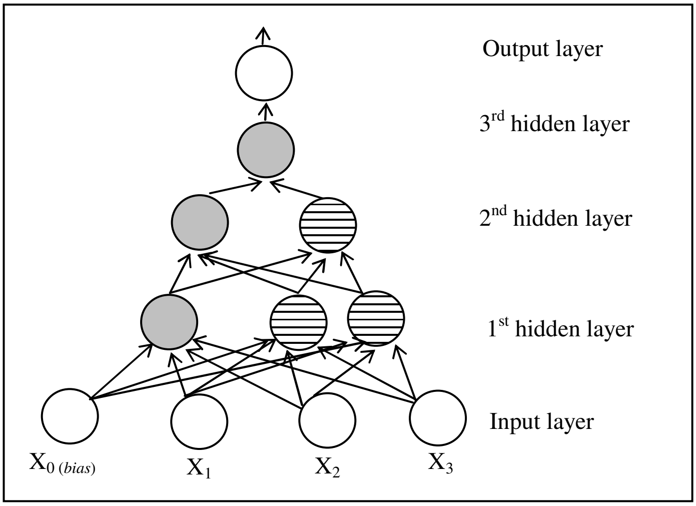


Fig. 1. General architecture of a Tiling-NN. Master neurons: shadowed; ancillary: crosshatched.

The *faithfulness criterion* employed by Tiling establishes that no two training patterns, belonging to different classes, should produce the same outputs at any given layer. Gallant in [1] comments: “The role of these units (ancillary) is to increase the number of cells for layer L so that no two training examples with *different classifications* have the *same set of activations* in layer L. Thus each succeeding layer has a different representation for the inputs, and no two training examples with different classifications have the same representation in any layer. Layers with this property are termed *faithful layers*, and faithfulness of layers is clearly a necessary condition for a strictly layered network to correctly classify all training examples”.

Tower, Pyramid and Tiling share the direction they grow the NN – from the input layer towards the output layer. The two other algorithms chosen for the experiments i.e. Perceptron-Cascade and Shift grow the NN in the opposite direction i.e., from the output layer towards the input layer. Perceptron-Cascade (PC) induces the same architecture of a neural network created by Cascade Correlation algorithm [15] and adopts the same error-correction strategy of the Upstart algorithm [16]. PC begins the construction of the network by training the output neuron. If this neuron does not classify all training instances correctly, the algorithm begins to add hidden neurons to the network. Each new added hidden neuron is connected to all previous hidden neurons as well as to the input neurons. The new hidden neuron is then connected to the output neuron; each time a hidden neuron is added, the output neuron needs to be retrained. The addition of a new hidden neuron enlarges the space in one dimension. The Shift algorithm creates only one hidden layer, iteratively adding neurons to it; each added neuron is connected to the input neurons and to the output neuron. The error correcting procedure used by Shift is also similar to the one used by Upstart.

3 Boolean Interpretation of Hidden Neurons

This section investigates possible interpretations associated with hidden neurons added to the NN when learning Boolean functions using the five CoNN algorithms from Section 2. The experiments were conducted using four Boolean functions. Table 1 shows the DNF (Disjunctive Normal Form) of Boolean functions f_1 and f_2 ($\{0,1\}^3 \rightarrow \{0,1\}$) as well as f_3 and f_4 ($\{0,1\}^4 \rightarrow \{0,1\}$) respectively. In this paper a Boolean function has been represented by its associated Boolean form. In what follows X1, X2, X3 and X4 represent Boolean variables. Also, the apostrophe is used to represent the unary operation in the Boolean algebra $<\{0,1\}, \vee, \wedge, ', 0, 1>$. A Boolean form in which the literals are combined by the $\wedge$ operator alone is known as a product. To simply notation, generally a product is represented only by its literals (e.g., X1’X2X3’ represents X1’ $\wedge$ X2 $\wedge$ X3’).

Table 1. DNF representation of Boolean functions f_1, f_2, f_3 and f_4

	DNF representation		DNF representation
f_1	X1’X2X3’ ∨ X1X2X3’ ∨ X1X2X3 ∨ X1X2’X3 ∨ X1’X2’X3	f_3	X1’X2X3’X4 ∨ X1’X2X3X4 ∨ X1X2X3X4 ∨ X1X2’X3’X4 ∨ X1X2X3X4’
f_2	X1’X2’X3’ ∨ X1’X2X3’ ∨ X1’X2X3 ∨ X1X2X3 ∨ X1X2’X3	f_4	X1’X2’X3X4 ∨ X1’X2X3X4 ∨ X1’X2X3’X4 ∨ X1X2X3’X4 ∨ X1X2X3’X4’ ∨ X1X2’X3’X4’ ∨ X1X2’X3X4’

Minimization methods are based on the concept of prime implicants. As defined in [17], a Boolean form α *covers* a Boolean form β if $\nu(\alpha)=1$ for all value assignments that make $\nu(\beta)=1$ (where ν is the valuation function). If α covers β and β is a simple product, then β is an *implicant* of α. If β is an implicant of α, and no other form obtainable from β by removal of literals is an implicant of α, then β is a *prime implicant* of α.

The minimized representations of the four functions shown in Table 2 were obtained using the Quine-McCluskey method [18][19]. It is worth reminding first that the products in a minimal sum of products equivalent to a disjunctive normal form must all be prime implicants. The method first generates the set of all prime implicants of the normal form (1st. column). Then it selects a subset of the prime implicants that defines a minimal sum of products (2nd. column). In order to do that it creates a table of covers, where rows correspond to the products of the normal form and columns to the prime implicants and uses the concept of cover to select the minimal sum of products [17]. Products that are covered by only one prime implicant turn that prime implicant into an *essential prime implicant*; every essential prime implicant must go into the minimal representation.

Table 2. Set of prime implicants and minimal sum of products of Boolean functions f$_1$, f$_2$, f$_3$ and f$_4$. Essential prime implicants are bold faced.

	Set of Prime Implicants	Minimized Representation
f$_1$	{X1X2, X1X3, **X2X3’**, **X2’X3**}	**X2X3’** ∨ **X2’X3** ∨ X1X2 or **X2X3’** ∨ **X2’X3** ∨ X1X3
f$_2$	{**X1X3**, X2X3, X1’X2, **X1’X3’**}	**X1X3** ∨ **X1’X3’** ∨ X1’X2 or **X1X3** ∨ **X1’X3’** ∨ X2X3
f$_3$	{**X1X2X3**, X2X3X4, **X1’X2X4**, **X1X2’X3’X4**}	**X1’X2’X3’X4** ∨ **X1X2X3** ∨ **X1’X2X4**
f$_4$	{X1X2X3’, X2X3’X4, X1’X2X4, **X1X2’X4’**, **X1’X3X4**, X1X3’X4’}	**X1X2’X4’** ∨ **X1’X3X4** ∨ X2X3’X4 ∨ X1X3’X4’ or **X1X2’X4’** ∨ **X1’X3X4** ∨ X2X3’X4 ∨ X1X2 X3’ or **X1X2’X4’** ∨ **X1’X3X4** ∨ X1X2X3’∨ X1’X2X4

The following tables use the notation: To (Tower), Py (Pyramid), PC(Perceptron-Cascade), Sh (Shift) and Ti(Tiling). The four first algorithms use HN$_i$ to represent the ith hidden neuron created (last one being the output). Tiling is the only algorithm whose inserted nodes can be functionally different (master and ancillary) and, for this reason, they are noted as HN$_{ij}$, where index i represents the layer and j the number of the neuron in the corresponding layer. Value j=1 identifies the master neuron of the corresponding layer and neurons having j > 1 represent ancillary neurons of the corresponding layer.

Hidden and output neurons are represented as the Boolean forms associated to corresponding Boolean functions from $\{0,1\}^3 \rightarrow \{0,1\}$ (f$_1$ and f$_2$ – Table 3 and Table 4 respectively) and from $\{0,1\}^4 \rightarrow \{0,1\}$ (f$_3$ and f$_4$ – Table 5 and tables 6-7 respectively). In the tables the Boolean forms associated with output neurons are bold faced.

Table 3. HN interpretations of CoNNs representing f$_1$

	To/Py	PC/Sh	Ti
HN$_1$	X2 ∨ X3	X1’X2X3	(HN$_{11}$) X2 ∨ X3
HN$_2$	**X1X2** ∨ **X1X3** ∨ **X2’X3** ∨ **X2X3’**	**X1X2** ∨ **X1X3** ∨ **X2’X3** ∨ **X2X3’**	(HN$_{12}$) X1 ∨ X2’ ∨ X3
			(HN$_{21}$) **X1X2** ∨ **X1X3** ∨ **X2’X3** ∨ **X2X3’**

For function f$_1$ (Table 3), Tower and Pyramid induced the same neural network whose hidden neurons are represented by the same Boolean expression. The same happened with PC and Shift. The first hidden neuron added by To/Py and by PC/Sh, however, represents different Boolean sub-expressions. The second neuron created by the four algorithms represents the disjunction of the four prime implicants of f$_1$, which is a simplified representation of the original function although not minimal. The same can be said about the results related to f$_2$ presented in Table 4.

Table 4. HN interpretations of CoNNs representing f_2

	To/Py	PC/Sh	Ti
HN$_1$	X1’ ∨ X3	X1’X2’X3	(HN$_{11}$) X1’ ∨ X3
HN$_2$	**X1X3** ∨ **X2X3** ∨ **X1’X2** ∨ **X1’X3’**	X1X3 ∨ X2X3 ∨ X1’X2 ∨ X1’X3’	(HN$_{12}$) X1 ∨ X2 ∨ X3’
			(HN$_{21}$) **X1X3** ∨ **X2X3** ∨ **X1’X2** ∨ **X1’X3’**

Table 5. HN interpretations of CoNNs representing f_3

	To/Py	PC/Sh	Ti
HN$_1$	X1’X2X4 ∨ X2X3X4	X1’X2’X3’X4	(HN$_{11}$) X1X3’X4 ∨ X1X2 ∨ X2X4
HN$_2$	X1’X2X4 ∨ X2X3X4 ∨ X1X2X3	X1X2X3X4’	(HN$_{12}$) X1’∨ X3 ∨ X2’X4 ∨
HN$_3$	**X1X2X3** ∨ **X2X3X4** ∨ **X1’X2X4** ∨ **X1X2’X3’X4**	**X1X2X3** ∨ **X2X3X4** ∨ **X1’X2X4** ∨ **X1X2’X3’X4**	(HN$_{21}$) **X1X2X3** ∨ **X2X3X4** ∨ **X1’X2X4** ∨ **X1X2’X3’X4**

Table 6. HN interpretations of CoNN (To, Py, PC, Sh) representing f_4

	To	Py	PC	Sh
HN$_1$	X1X2’X4’ ∨ X1X3’	X1X3’	X1’X3X4 ∨ X1’X2X4	X1’X3X4
HN$_2$	X1X2’X4’ ∨ X1’X3X4 ∨ X1’X2X4 ∨ X2X3’X4 ∨ X1X2’X4’ ∨ X1X3’	X1’X3X4 ∨ X1’X2X4 ∨ X2X3’X4 ∨ X1X3’	X2’X3’X4	X1X2’X3X4
HN$_3$	**X1X2’X4’** ∨ **X1X3’X4’** ∨ **X1’X3X4** ∨ **X1’X2X4** ∨ **X1X2X3’** ∨ **X2X3’X4**	X1X3’X4’ ∨ X1’X2X4 ∨ X1’X3X4 ∨ X2X3’X4 ∨ X1X2X3’	**X1X2’X4’** ∨ **X1X3’X4’** ∨ **X1’X3X4** ∨ **X1’X2X4** ∨ **X1X2X3’** ∨ **X2X3’X4**	X1’X2X4’
HN$_4$		**X1X2’X4** ∨ **X1X3’X4’** ∨ **X1’X3X4** ∨ **X1’X2X4** ∨ **X1X2X3’** ∨ **X2X3’X4**		**X1X2’X4’** ∨ **X1X3’X4’** ∨ **X1’X3X4** ∨ **X1’X2X4** ∨ **X1X2X3’** ∨ **X2X3’X4**

Table 7. HN interpretations of CoNN (Tiling) representing f_4

HN$_{11}$	HN$_{12}$	HN$_{13}$	HN$_{21}$	HN$_{22}$	HN$_{31}$
X2X3’ ∨ X1X3’X4	X1X3’ ∨ X3’X4 ∨ X1X4	X4 ∨ X1X2’	X1X3’X4’ ∨ X2X3’X4 ∨ X1X2X3’	X1X2’X3X4’ ∨ X1X3X4	**X1X2’X4’** ∨ **X1X3’X4’** ∨ **X1’X3X4** ∨ **X1’X2X4** ∨ **X1X2X3’** ∨ **X2X3’X4**

As expected, Tower and Pyramid performed similarly. As far as functions f_1, f_2 and f_3 are concerned, the two algorithms induced exactly the same NNs (apart from connections), where the created hidden neurons represent the same Boolean sub-expressions respectively. When learning f_4, however, Tower and Pyramid induced different architectures and, apart from the last neuron created by both (the output neuron representing the disjunction of all prime implicants), all the other hidden neurons represent different Boolean sub-expressions. The differences can be justified mainly by the random initialization of the weight vector and, perhaps, the reduced number of training instances. Tower and Pyramid add neurons sequentially until convergence is reached; the output neuron directly reflects the previously added neuron sub-expressions.

Surprisingly, PC and Shift, in spite of being distinctive algorithms, also induced NNs whose sequence of created hidden neurons is represented exactly by the same Boolean sub-expressions, when learning functions f_1, f_2 and f_3. PC and Shift start the construction of the NN from the output neuron, by adding new neurons according to the most frequent error criterion (*wrongly-on* or *wrongly-off* errors). The most frequent error criterion is the predominant strategy employed by backward-driven constructive NN algorithms. Both algorithms also expand the output weight vector by one dimension each time a hidden neuron is added to the NN. Particularly in a PC architecture, not only the output neuron but each new added neuron has one dimension more than the previously added neuron. The backward-driven PC and Shift do not have an embedded link among hidden and output neuron sub-expressions due to the constant retraining of the output neuron.

The strategy of employing hidden neurons with different functionalities, as implemented by Tiling, adds an extra degree of refinement to the algorithm which, not necessarily, translates into a more suitable procedure for Boolean learning. Although reaching the same results as the others, Tiling-NNs have the tendency of being bigger. Also, the change in representation promoted by the use of ancillary neurons is more suitable for real-valued tasks.

4 Conclusions

This paper investigates the symbolic interpretation of the hidden nodes added to a NN during the learning process of a Boolean expression, focusing on five CoNN algorithms: Tower, Pyramid, Perceptron Cascade, Shift and Tiling. Hidden nodes can be interpreted as Boolean sub-expressions which are combined in such a way that the final neuron added represents the given function represented as a disjunction of its prime implicants. In this sense CoNN algorithms play the same role as the first half of the Quine-McCluskey algorithm. The next step of the work will focus on a few other CoNN algorithms (initially BabCoNN [20], PTI [14], Upstart [16]) to gather empirical evidence to support the conclusions reported in this paper.

Acknowledgements. To CNPq for the research scholarship given to the first and second authors.

References

[1] Gallant, S.I.: Neural Network Learning & Expert Systems. The MIT Press, Cambridge (1994)

[2] do Carmo Nicoletti, M., Bertini Jr., J.R., Elizondo, D., Franco, L., Jerez, J.M.: Constructive neural network algorithms for feedforward architectures suitable for classification tasks. In: Franco, L., Elizondo, D.A., Jerez, J.M. (eds.) Constructive Neural Networks. SCI, vol. 258, pp. 1–23. Springer, Heidelberg (2009)

[3] Bishop, C.M.: Neural networks for pattern recognition. Oxford University Press, Great Britain (1999)

[4] Franco, L., Elizondo, D.A., Jérez, J.M. (eds.): Constructive Neural Networks. SCI, vol. 258. Springer, Germany (2009)

[5] Setiono, R., Leow, W.K.: FERNN: an algorithm for fast extraction of rules from neural networks. Appl. Intell. 1, 15–25 (2000)

[6] Garcez, A.S.A., Broda, K., Gavia, D.M.: Symbolic knowledge extraction from trained neural network: a sound approach. Artif. Intell. 125, 155–207 (2001)

[7] Fu, L.: Rule generation from neural networks. IEEE Trans. Syst. Man Cybern. 24(8), 1114–1124 (1994)

[8] Towell, G.G., Shavlik, J.W.: Extracting refined rules from knowledge-based neural networks. Mach. Learn. 13, 71–101 (1993)

[9] Craven, M.W., Shavlik, J.W.: Learning symbolic rules using artificial neural networks. In: Proc. of the 10th ICML, pp. 73–80. Morgan Kaufmann, San Mateo (1993)

[10] Andrews, R., Diederich, J., Tickle, A.B.: Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowl. Base. Syst. 8(6), 373–389 (1995)

[11] Pagallo, G., Haussler, D.: Boolean feature discovery in empirical learning. Mach. Learn. 5, 71–99 (1990)

[12] Mézard, M., Nadal, J.: Learning feedforward networks: the tiling algorithm. J. Phys. A: Math. Gen. 22, 2191–2203 (1989)

[13] Burgess, N.: A constructive algorithm that converges for real-valued input patterns. Int. J. Neural Syst. 5(1), 59–66 (1994)

[14] Amaldi, E., Guenin, B.: Two constructive methods for designing compact feedfoward networks of threshold units. Int. J. Neural Syst. 8(5&6), 629–645 (1997)

[15] Fahlman, S., Lebiere, C.: The cascade correction architecture. In: Advances in Neural Information Processing Systems, vol. 2, pp. 524–532. Morgan Kaufmann, San Mateo (1990)

[16] Frean, M.: The upstart algorithm: a method for constructing and training feedforward neural networks. Neural Comput. 2, 198–209 (1990)

[17] Berztiss, A.T.: Data structures – theory and practice, 2nd edn. Academic Press, N.Y (1975)

[18] Quine, W.V.: The problem of simplifying truth tables. Amer. Math. Month. 59, 51–531 (1952)

[19] McCluskey, E.J.: Minimization of Boolean functions. Bell System Tech. J. 35, 1417–1444 (1956)

[20] Bertini Jr., J.R., Nicoletti, M.C.: A constructive neural network algorithm based on the geometric concept of barycenter of convext hull. In: Rutkoski, R.L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J. (eds.) Computational Intelligence: Methods and Applications, pp. 1–12. Academic Publishing House Exit, Poland (2008)

An Experimental Study on Asymmetric Self-Organizing Map

Dominik Olszewski

Faculty of Electrical Engineering,
Warsaw University of Technology, Poland
olszewsd@ee.pw.edu.pl

Abstract. The paper presents an extension of the justification for use of the asymmetric Self-Organizing Map (SOM). We claim that it can successfully applied in the wider area of research than the textual data analysis. The results of our experimental study in the fields of sound recognition and heart rhythm recognition confirm this claim, and report the superiority of the asymmetric approach over the symmetric one, in both parts of our experiments.

Keywords: Self-Organizing Map, Asymmetric Self-Organizing Map, Sound recognition, Heart rhythm recognition.

1 Introduction

The Self-Organizing Map (SOM) [1] is an example of the artificial neural network architecture. It was introduced by T. Kohonen, and it can be also interpreted as a visualization technique, since the algorithm performs a projection from multidimensional space to 2-dimensional space, this way creating a map structure. The location of points in 2-dimensional grid aims to reflect the similarities between the corresponding objects in multidimensional space. Therefore, the SOM algorithm allows for visualization of relationships between objects in multidimensional space. The asymmetric version of the SOM algorithm was introduced in [2]. However, the justification provided in this paper was related to the hierarchical associations in textual data. The aim of this paper is to show that similar phenomenon can be found in case of the sound signals and human heart rhythm signals. Consequently, the same assertion referring to hierarchical asymmetric relationships in data can be used to justify the use of the asymmetric SOM algorithm version. In other words, we can assert that our paper extends the range of application of the asymmetric SOM method.

The rest of this paper is organized as follows: in Section 2, the traditional symmetric version of the SOM algorithm is described; in Section 3, the asymmetric relationships in data sets are discussed; in Section 4, the asymmetric version of the SOM algorithm is presented; in Section 5, our experimental results are reported; while Section 6 summarizes the whole paper, and gives some concluding remarks.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 42–49, 2011.

2 Symmetric Self-Organizing Map

The SOM algorithm provides a non-linear mapping between a high-dimensional original data space and a 2-dimensional map of neurons. The neurons are arranged according to a regular grid, in such a way that the similar vectors in input space are represented by the neurons close in the grid. Therefore, the SOM technique visualize the data associations in the input high-dimensional space.

It was shown in [3] that the results obtained by the SOM method are equivalent to the results obtained by optimizing the following error function:

$$e(\mathcal{W}) = \sum_r \sum_{x_\mu \in V_r} \sum_s h_{rs} D(x_\mu,\ w_s) \tag{1}$$

$$\approx \sum_r \sum_{x_\mu \in V_r} D(x_\mu,\ w_r)\ +\ K \sum_r \sum_{s \neq r} h_{rs} D(w_r,\ w_s)\ , \tag{2}$$

where x_μ are the objects in high-dimensional space, w_r and w_s are the prototypes of objects on the grid, h_{rs} is a neighborhood function (for example, the Gaussian kernel) that transforms non-linearly the neuron distances (see [1] for other choices of neighborhood functions), $D(\bullet,\ \bullet)$ is the squared Euclidean distance, and V_r is the Voronoi region corresponding to prototype w_r. The number of prototypes is sufficiently large so that $D(x_\mu,\ w_s) \approx D(x_\mu,\ w_r)\ +\ D(w_r,\ w_s)$.

According to equation (2), the SOM error function can be decomposed as the sum of the quantization error and the topographic error. The first one minimizes the loss of information, when the input patterns are represented by a set of prototypes. By minimizing the second one, we assure the maximal correlation between the prototype dissimilarities and the corresponding neuron distances, this way assuring the visualization of the data relationships in the input space.

The SOM error function can be optimized by an iterative algorithm consisting of two steps (discussed in [3]). First, a quantization algorithm is executed. This algorithm represents each input pattern by the nearest neighbor prototype. This operation minimizes the first component in equation (2). Next, the prototypes are arranged along the grid of neurons by minimizing the second component in the error function. This optimization problem can be solved explicitly using the following adaptation rule for each prototype [1]:

$$w_s\ =\ \frac{\sum_{r=1}^{M} \sum_{x_\mu \in V_r} h_{rs} x_\mu}{\sum_{r=1}^{M} \sum_{x_\mu \in V_r} h_{rs}}\ , \tag{3}$$

where M is the number of neurons, and h_{rs} is a neighborhood function (for example, the Gaussian kernel of width $\sigma(t)$). The width of the kernel is adapted in each iteration of the algorithm using the rule proposed by [4], i.e., $\sigma(t)\ =\ \sigma_i (\sigma_f / \sigma_i)^{t/N_{iter}}$, where $\sigma_i\ \approx\ M/2$ is typically assumed in the literature (for example, in [1]), and σ_f is the parameter that determines the smoothing degree of the principal curve generated by the SOM algorithm [4].

3 Asymmetry in Data

The problem of asymmetry in data analysis was widely studied in the literature. The research of A. Okada and T. Imaizumi [5] is focused on using the dominance point governing asymmetry in the proximity relationships among objects, represented as points in the multidimensional Euclidean space. They claim that ignoring or neglecting the asymmetry in proximity analysis discards potentially valuable information. On the other hand, B. Zielman and W. Heiser in [6] consider the models for asymmetric proximities as a combination of a symmetric similarity component and an asymmetric dominance component. The author of [7], introduces the asymmetric version of the well-known k-means clustering algorithm. Finally, the paper [2] proposes the asymmetric version of the SOM algorithm, which was an inspiration for our research.

When an analyzed data set appears to have asymmetric properties, the symmetric measures of similarity or dissimilarity (for example, the most popular Euclidean distance) does not apply properly to this phenomenon, and for most pairs of data points, they produce small values (similarities) or big values (dissimilarities). Consequently, they do not reflect accurately the relationships between objects. The asymmetry in data set arises, for example, in case, when the data associations have a hierarchical nature. The hierarchical connections in data are closely related to the asymmetry. This relation has been noticed in [8]. In case of the dissimilarity, when it is computed in the direction – from a more general entity to a more specific one – it should be greater than in the opposite direction. As stated in [2], asymmetry can be interpreted as a particular type of hierarchy.

An idea to overcome this problem is to employ the asymmetric similarities and dissimilarities. They should be applied in algorithms in such way, so that they would properly reflect the hierarchical asymmetric relationships between objects in the analyzed data set. Therefore, it should be guaranteed that their application is consistent with the hierarchical associations in data. This can be achieved by use of the asymmetric coefficients, inserted in the formulae of symmetric measures. This way, we can obtain the asymmetric measures on the basis of the symmetric ones. The asymmetric coefficients should assure the consistence with the hierarchy. Hence, in case of the dissimilarities, they should assure greater values in the direction – from more general concept to more specific one.

This paper points out that the phenomenon of the hierarchy-caused asymmetry occurs in a wider range of applications than the text analysis, as it was presented in [2]. Our experimental study concerns the sound signals clustering and human heart rhythm clustering, and confirms the existence of the same phenomenon.

3.1 Asymmetric Coefficients

Asymmetric coefficients convey the information provided by asymmetry. Two coefficients were introduced in [9]. The first one is derived from the fuzzy logic similarity, and the second one formulated on the basis of the Kullback-Leibler divergence. Both of these quantities are widely used in statistics and probability theory. In our experimental study, we have used the first of these coefficients.

Hence, the fuzzy-logic-based asymmetric coefficient is formulated as follows:

$$a_i = \frac{|f_i|}{\max_j \left(|f_j|\right)} , \tag{4}$$

where f_i are the features of objects in the analyzed data set (f_i are the entries of the vectors representing the objects), and $|\bullet|$ is the L_1-norm meaning the number of objects possessing the feature given as the argument.

This coefficient takes values in the $[0, 1]$ interval. Intuitively speaking, it will become large for general (broad) concepts with large L_1-norm.

Note that the asymmetric coefficients must be computed and assigned to each feature of every object in the analyzed data set.

4 Asymmetric Self-Organizing Map

In order to formulate the asymmetric version of the SOM algorithm, we will refer to the error function (2). As it was stated in Section 2, the results produced by the SOM method are identical to the results obtained by optimizing the function (2).

The asymmetric SOM algorithm is derived in three steps:

Step 1. Transform a symmetric dissimilarity (for example, the Euclidean distance) into a similarity:

$$S_{ij}^{\mathrm{SYM}} = C - d^2\left(x_i, x_j\right) , \tag{5}$$

where $d^2\left(x_i, x_j\right)$ is the squared Euclidean distance between objects x_i and x_j, and the constant C is the upper boundary of the squared Euclidean distance.

Step 2. Transform the symmetric similarity into the asymmetric similarity:

$$S_{ij}^{\mathrm{ASYM}} = a_i \left(C - d^2\left(x_i, x_j\right)\right) , \tag{6}$$

where a_i is the asymmetric coefficient defined in Subsection 3.1, in (4), and the rest of notation is described in (5). The asymmetric similarity defined this way, with use of the asymmetric coefficient guarantees the consistency with the asymmetric hierarchical associations among objects in the data set.

Step 3. Insert the asymmetric similarity in the error function (2), in order to obtain the energy function, which needs to maximized:

$$E\left(\mathcal{W}\right) = \sum_{r} \sum_{x_\mu \in V_r} \sum_{s} h_{rs} a_i \left(C - d^2\left(x_i, x_j\right)\right) , \tag{7}$$

where the notation is explained in (2), (5), and (6). The energy function (7) can be optimized in the similar way as the error function (2). Firstly, we run the quantization algorithm, which generates the SOM prototypes w_s. Secondly, the energy function is maximized by solving the set of linear equations $\partial E(\mathcal{W}) / \partial w_s = 0$. This system of linear equation can be solved explicitly, by using the following updating formula for the SOM prototypes w_s:

$$w_s = \frac{\sum_{r=1}^{M} \sum_{x_\mu \in V_r} h_{rs} a_\mu x_\mu}{\sum_{r=1}^{M} \sum_{x_\mu \in V_r} h_{rs} a_\mu}, \tag{8}$$

where a_i is the asymmetric coefficient, h_{rs} is a neighborhood function (for example, the Gaussian kernel), and the rest of notation is the same as in (3). This updating formula is similar to the adaptation rule (3), with the difference that the asymmetric coefficient is inserted. In case of use of the Gaussian kernel, its width $\sigma(t)$ can be adapted, like it is done in (3). An important property of the asymmetric SOM algorithm is that it maintains the simplicity of the traditional symmetric approach, and does not increase the computational complexity.

5 Experiments

Our experimental study aims to confirm that the asymmetric version of the SOM algorithm can be applied in the wider area of research than it was proposed in [2], and the justification referring to the textual data analysis can be extended to the other types of data, for example, to sound signals and human heart rhythm signals, which were the subject of our empirical study.

In case of both parts of our experiments, we have compared the results obtained with use of the symmetric and asymmetric SOM algorithms. As the basis of the comparisons, i.e., as the evaluation metrics, we have used the accuracy degree [7], and the entropy measure [2].

- **Accuracy degree.** This evaluation metric determines the number of correctly assigned objects divided by the total number of objects in the data set. Firstly, the centroids of the clustered data are computed, and next, each object is assigned to the cluster represented by the centroid – nearest to this object. Finally, the number of correctly assigned objects is divided by the total number of all objects. The accuracy degree assumes values in the interval $[0, 1]$, and naturally, greater values are preferred.
- **Entropy measure.** This evaluation metric determines the number of overlapping objects divided by the total number of objects in the data set. This means, the number of objects, which are in the overlapping area between clusters, divided by the total number of objects. In other words, it determines the uncertainty for the classification of objects that belong to the same cluster. The entropy measure assumes values in the interval $[0, 1]$, and, smaller values are desirable.

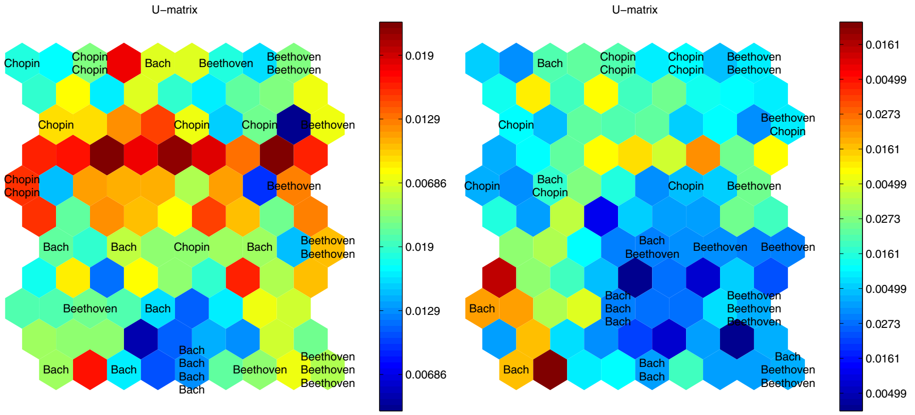
The sound signals, we have analyzed, were the piano music recordings, and the human heart rhythm signals were analyzed on the basis of the ECG recordings derived from the MIT-BIH ECG Databases.

5.1 Piano Music Composer Clustering

In this part of our experiments, we have tested our enhancement to the SOM algorithm and the classical SOM forming three clusters representing three piano music composers: Johann Sebastian Bach, Ludwig van Beethoven, and Fryderyk Chopin. Each music piece was represented with a 20-seconds sound signal sampled with the 44100 Hz frequency. The entire data set was composed of 32 sound signals. The feature extraction process was carried out according to the traditional Discrete-Fourier-Transform-based (DFT-based) method. The DFT was implemented with the fast Fourier transform (FFT) algorithm. Sampling signals with the 44100 Hz frequency resulted in the 44100/2 Hz value of the upper boundary of the FFT result range.

The results of this part of our experiments are demonstrated in Fig. 1 and in Table 1. Figure 1 presents the U-matrices generated by the symmetric (Fig. 1(a)) and asymmetric (Fig. 1(b)) SOM algorithms. Table 1, in turn, presents the accuracy degrees and the entropy measures corresponding to the symmetric and asymmetric SOM approaches.

The results of this part of our experimental study report the superiority of the asymmetric SOM algorithm over the symmetric counterpart. The asymmetric approach leads to the higher clustering accuracy measured on the basis of the accuracy degree (0.9375 vs. 0.8438), and also, it leads to the lower cluster overlapping determined on the basis of the entropy measure (0.1563 vs. 0.2500).



(a) Symmetric SOM

(b) Asymmetric SOM

Fig. 1. Piano Music Composer Clustering Maps

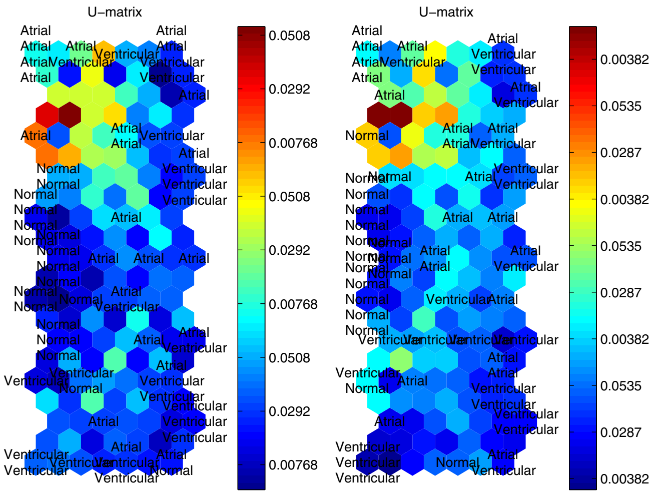
Table 1. Accuracy degrees and entropy measures of the piano music composer clustering

	Symmetric SOM	Asymmetric SOM
Accuracy degree	$27/32 = 0.8438$	$30/32 = 0.9375$
Entropy measure	$8/32 = 0.2500$	$5/32 = 0.1563$

5.2 Human Heart Rhythm Clustering

In this part of our experiments, we have investigated asymmetric SOM and the traditional SOM approach forming three clusters representing three types of human heart rhythms: normal sinus rhythm, atrial arrhythmia, and ventricular arrhythmia. This kind of clustering can be interpreted as the cardiac arrhythmia detection and recognition based on the ECG recordings. In general, the cardiac arrhythmia disease may be classified either by rate (tachycardias – the heart beat is too fast, and bradycardias – the heart beat is too slow) or by site of origin (atrial arrhythmias – they begin in the atria, and ventricular arrhythmias – they begin in the ventricles). Our clustering recognizes the normal rhythm, and, also, recognizes arrhythmias originating in the atria, and in the ventricles. We analyzed 20-minutes ECG holter recordings sampled with the 250 Hz frequency. The entire data set was composed of 63 ECG signals. The feature extraction was carried out in the same way, like it was done with the piano music composer clustering.

The results of this part of our experiments are presented in Fig. 2 and in Table 2, which are constructed in the same way as in Subsection 5.1.



(a) Symmetric SOM (b) Asymmetric SOM

Fig. 2. Human Heart Rhythm Clustering Maps

It is clear that, in the case of the ECG recording clustering, the asymmetric SOM method, again, outperformed the symmetric one, by providing the higher clustering quality (accuracy degree: 0.7778 vs. 0.7143), and the lower clustering uncertainty (entropy measure: 0.2540 vs. 0.2857).

Table 2. Accuracy degrees and entropy measures of the human heart rhythm clustering

	Symmetric SOM	Asymmetric SOM
Accuracy degree	$45/63 = 0.7143$	$45/63 = 0.7143$
Entropy measure	$18/63 = 0.2857$	$16/63 = 0.2540$

6 Summary

The paper presented the results of the experimental study on the asymmetric SOM algorithm, in the fields of piano music composer clustering, and human heart rhythm clustering. According to our experimental results, the asymmetric SOM outperforms the symmetric one, in both studied cases, by providing the higher clustering accuracy, and lower entropy measure. This means that our results confirmed that the hierarchy-caused asymmetric relationships also occur in the analyzed data sets. This conclusion extends the justification of use of the asymmetric SOM algorithm beyond the textual data analysis, which was the aim of this paper.

References

1. Kohonen, T.: Self-Organizing Maps, 3rd edn. Springer, Heidelberg (2001)
2. Martín-Merino, M., Muñoz, A.: Visualizing Asymmetric Proximities with SOM and MDS Models. Neurocomputing 63, 171–192 (2005)
3. Heskes, T.: Self-Organizing Maps, Vector Quantization, and Mixture Modeling. IEEE Transactions on Neural Networks 12(6), 1299–1305 (2001)
4. Mulier, F., Cherkassky, V.: Self-Organization as an Iterative Kernel Smoothing Process. Neural Computation 7(6), 1165–1177 (1995)
5. Okada, A., Imaizumi, T.: Multidimensional Scaling of Asymmetric Proximities with a Dominance Point. In: Advances in Data Analysis. Studies in Classification, Data Analysis, and Knowledge Organization, pp. 307–318. Springer, Heidelberg (2007)
6. Zielman, B., Heiser, W.J.: Models for Asymmetric Proximities. British Journal of Mathematical and Statistical Psychology 49, 127–146 (1996)
7. Olszewski, D.: Asymmetric *k*-Means Algorithm. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part II. LNCS, vol. 6594, pp. 1–10. Springer, Heidelberg (2011)
8. Muñoz, A., Martin, I., Moguerza, J.M.: Support Vector Machine Classifiers for Asymmetric Proximities. In: Kaynak, O., Alpaydın, E., Oja, E., Xu, L. (eds.) ICANN 2003 and ICONIP 2003. LNCS, vol. 2714, pp. 217–224. Springer, Heidelberg (2003)
9. Muñoz, A., Martín-Merino, M.: New Asymmetric Iterative Scaling Models for the Generation of Textual Word Maps. In: Proceedings of the International Conference on Textual Data Statistical Analysis JADT 2002, pp. 593–603 (2002)

A Language Specification Tool for Model-Based Parsing

Luis Quesada, Fernando Berzal, and Juan-Carlos Cubero

Department of Computer Science and Artificial Intelligence, CITIC,
University of Granada, Granada 18071, Spain
{lquesada,fberzal,jc.cubero}@decsai.ugr.es

Abstract. Typically, formal languages are described by providing a textual BNF-like notation specification, which is then manually annotated for syntax-directed translation. When the use of an explicit model is required, its implementation requires the development of the conversion steps between the model and the grammar, and between the parse tree and the model instance. Whenever the language specification is modified, the developer has to manually propagate changes throughout the entire language processor pipeline. These updates are time-consuming, tedious, and error-prone. Besides, in the case that different applications use the same language, the developer has to maintain several copies of the same language specification. In this paper, we introduce a model-based parser generator that decouples language specification from language processing, hence avoiding many of the problems caused by grammar-driven parsers and parser generators.

Keywords: Language specification, parser generator, Model-Driven Software Development (MDSD).

1 Introduction

Formal languages allow the expression of information in the form of symbol sequences [3]. A formal language consists of an alphabet, which describes the basic symbol or character set of the language, and a grammar, which describes how to form valid sentences in the language. In Computer Science, formal languages are used for the precise definition of the syntax of data formats and programming languages, among other things.

Most existing language specification techniques [2] require the developer to provide a textual specification of the language grammar. The proper specification of such a grammar is a nontrivial process that depends on the lexical and syntactic analysis techniques to be used, since each kind of technique requires the grammar to comply with different restrictions.

When the use of an explicit model is required, its implementation requires the development of the conversion steps between the model and the grammar, and between the parse tree and the model instance. Thus, in this case, the implementation of the language processor becomes harder.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 50–57, 2011.

Whenever the language specification is modified, the developer has to manually propagate changes throughout the entire language processor pipeline. These updates are time-consuming, tedious, and error-prone. This hampers the maintainability and evolution of the language [11].

Typically, different applications that use the same language are developed. For example, the compiler, different code generators, and the tools within the IDE, such as the editor or the debugger. The traditional language processor development procedure enforces the maintenance of several copies of the same language specification in sync.

In contrast, generating a model-based language specification is performed visually and does not require the development of any conversion steps. By following this approach, the model can be modified as needed without having to worry about the language processor, which will be automatically updated accordingly. Also, as the software code can be combined with the model in a clean fashion, there is no embedding or mixing with the language processor. Finally, as the model is not bound to a specific analysis technique, it is possible to evaluate the alternative or complementary techniques that fit a specific problem, without propagating the restrictions of the used analysis technique into the model.

Our approach to model-based language specification has direct applications in the following fields:

- The generation of language processors (compilers and interpreters) [1].
- The specification of domain-specific languages (DSLs), which are languages oriented to the domain of a particular problem, its representation, or the representation of a specific technique to solve it [7,8,17].
- The development of Model-Driven Software Development (MDSD) tools [21].
- Data integration, as part of the preprocessing process in data mining [22].
- Text mining applications [23,4], in order to extract high quality information from the analysis of huge text data bases.
- Natural language processing [9] in restricted lexical and syntactic domains.
- The corpus-based induction of models [12].

Although there are tools that generate language processors from graphical language specifications [19,6], to the best of our knowledge, no existing tool follows the approach we describe in this paper.

In this paper, we introduce ModelCC, a model-based tool for language specification. ModelCC acts as a parser generator that decouples language specification from language processing, hence avoiding many of the problems caused by grammar-driven parsers and parser generators.

2 Background

Formal grammars are used to specify the syntax of a language [1].

Context-free grammars are formal grammars in which the productions are of the form $N \rightarrow (\Sigma \cup N)^*$ [3]. These grammars generate context-free languages.

A context-free grammar is said to be ambiguous if there exists a string that can be generated in more than one way. A context-free language is inherently ambiguous if all context-free grammars generating it are ambiguous.

Typically, language processing tools divide the analysis into two separate phases; namely, scanning (or lexical analysis) and parsing (or syntax analysis).

A lexical analyzer, also called lexer or scanner, processes an input string conforming to a language specification and produces the tokens found within it. A syntactic analyzer, also called parser, processes an input data structure consisting of tokens and determines its grammatical structure with respect to the given language grammar, usually in the form of parse trees.

Traditional efficient parsers for restricted context-free grammars, as the LL [20], SLL, LR [14], SLR, LR(1), or LALR parsers [1], do not consider ambiguities in syntactic analysis, so they cannot be used to perform parsing in those cases. The efficiency of these parsers is $O(n)$, being n the token sequence length.

Existing parsers for unrestricted context-free grammar parsing, as the CYK parser [24,10] and the Earley parser [5], can consider syntactic ambiguities. The efficiency of these parsers is $O(n^3)$, being n the token sequence length.

Lex and *yacc* [15] are well-known lexer generator and parser generator, respectively. It is difficult to specify all the constructions of a language in BNF-like notation without causing conflicts that these tools do not support [1].

ANTLR [18] is a lexer and parser generator that allows the generation of tree parsers. Tree parsers are recognizers that process abstract syntax trees instead of symbol sequences. This tool generates an LL(*) parser, which does not accept ambiguous grammar specifications either.

YAJco [19] is a lexer and parser generator that accepts as input a set of Java classes with annotations that specify the prefixes, suffixes, operators, tokens, parentheses and optional elements. This tool generates a BNF specification for JavaCC [16], which is a lexer and parser generator that supports LL(k) grammars. Therefore, the developer still has to be careful so the grammar implicit in the Java class set complies with the LL(k) grammar restrictions.

3 Model-Based Language Specification

We introduce ModelCC, a model-based tool for language specification that generates a language processor from a model.

3.1 Abstract Syntax versus Concrete Syntax

The abstract syntax of a language is just a representation of the structure of the different components of a language without the superfluous details related to its particular textual representation [13]. On the other hand, concrete syntax is a particularization of the abstract syntax that defines, with precision, a specific textual or graphical representation of a language. It should be noted that a single abstract syntax can be shared by several concrete syntaxes.

For example, the abstract syntax of the typical *if-then-optional else* sentence of an imperative programming language could be specified as a composition of a condition and one or two sentences. Two concrete syntaxes corresponding to specific textual representations of such a conditional sentence could be specified as: {“if”, “(”, expression, “)”, sentence, and optionally “else” and another sentence}, and {“IF”, expression, “THEN”, sentence, optionally “ELSE” and another sentence, and “ENDIF”}.

When using ModelCC, the language designer has to focus on the language abstract syntax model instead of focusing on specifying the BNF-like notation that describes a concrete syntax.

The advantages of this approach have been widely studied [13]:

- Specifying the abstract syntax seems to be a better starting point than specifying a concrete syntax.
- The language designer is able to modify the abstract syntax model and generate a working IDE on the run.
- It is not necessary for the developer to have advanced knowledge on parser generators to develop a language interpreter. In particular, the developer will not need to face the restrictions these parser generators usually impose.
- Priorities and associativity restrictions between elements that can cause ambiguities can be effortlessly established and modified.

3.2 Metamodel-Based Approach versus Traditional Approach

A diagram summarizing the traditional language specification procedure is shown in Figure 1. It illustrates the requirements of giving a BNF-like language specification and converting it into an attribute grammar. It also shows the lack of an explicit representation of the abstract syntax model, and the fact that the concrete syntax is the starting point of the process.

A diagram summarizing the model-based language specification approach used by ModelCC is shown in Figure 2. Developer workload is reduced as it just involves defining an abstract syntax model, which is annotated to automatically generate the grammar of the concrete syntax and its corresponding parser.

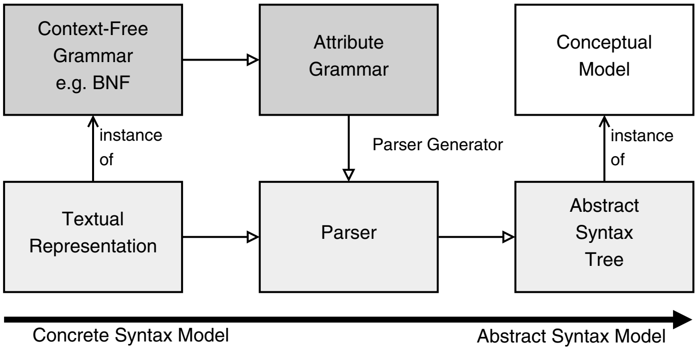


Fig. 1. Traditional language processing approach

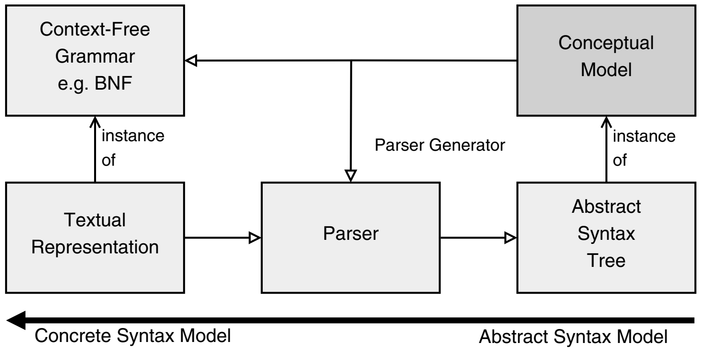


Fig. 2. Our approach to model-based language specification and processing

3.3 Model Specification

ModelCC provides the developer several mechanisms that can be used to create a model. Two of them are typical in model specification: inheritance and composition. The rest are annotations that complement the model elements by specifying patterns, delimiters, cardinality, and evaluation order. A summary of the annotations supported by ModelCC is shown in Figure 3

Constraints on...	Annotation	Usage
Patterns	@Pattern	Pattern matcher for a specific element.
	@Value	Field where the matched text should be stored.
Delimiters	@Prefix	Element prefix(es).
	@Suffix	Element suffix(es).
	@Separator	Element enumeration separator(s).
Cardinality	@Optional	Composition optionality.
	@Minimum	Minimum element multiplicity.
	@Maximum	Maximum element multiplicity.
Evaluation order	@Associativity	Element associativity (e.g. left-to-right).
	@Composition	Eager or lazy constructions.
	@Priority	Element precedence level/relationships.

Fig. 3. Summary of the annotations supported by ModelCC

4 Benefits of Model-Based Language Specification

As a simple example of the expression power of ModelCC, we have specified a simple calculator-like language that supports the following constructions:

- Unary operators: +, and -.
- Binary operators: +, -, *, and /.
- The binary operators * and / share the higher precedence.

- The binary operators + and - share the lower precedence.
- Parenthesis can be used to enforce precedence.
- Integer and real number support, althought results are always real numbers.

The model-based specification of this language is shown in Figure 4.

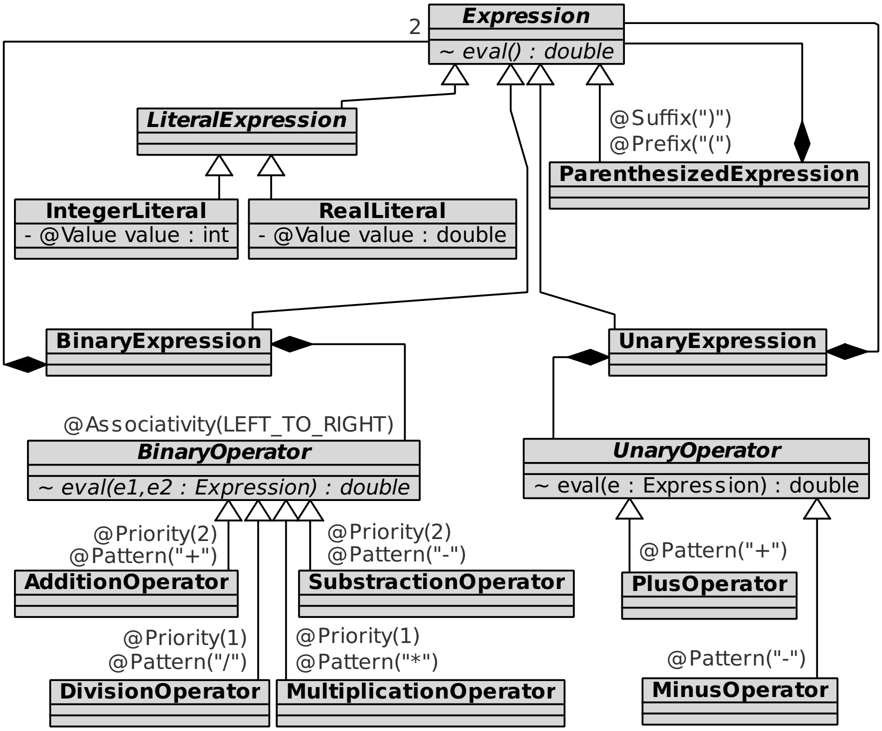


Fig. 4. ModelCC specification of a simple calculator language

Besides the model-based approach, the main functional advantages of using ModelCC over other existing tools such as *lex/yacc*, *YAJCo*, or *ANTLR* are the following:

- Apart from regular expressions, ModelCC allows the usage of pattern matching classes, which can be coded for specific purposes. For example, a dictionary-based matcher, in contrast to a regular expression-based matcher, could be used for detecting verbal forms in ModelCC.
- ModelCC supports multiple composition constructions. There is no need to bring the BNF-like notation recursion of enumeration specifications to the model.
- ModelCC offers a generic associativity and priority mechanism instead of a specific and limited operator specification mechanism. It supports creating

operator-alike constructions as complex as needed. For example, it allows the usage of non terminal symbols as operators and defining n-ary operators.
- ModelCC provides mechanisms that allow the developer to solve most language ambiguities. For example, expression nesting ambiguities can be solved by using associativities and priorities, and *if-then-optional else* sentence nesting ambiguities can be solved by using composition restrictions.

5 Conclusions and Future Work

We have introduced ModelCC, a model-based tool for language specification that automatically generates a parser from a model representing the abstract syntax of the language.

ModelCC automates several steps of the language processor implementation process and it improves the maintainability of languages.

Moreover, ModelCC allows the reuse of a language specification among different applications, eliminating the duplication required by conventional tools and improving the modularity of a language processing tool set, since it allows the use of object-oriented design techniques to cleanly separate language specification from language processing.

It should be noted that the ModelCC approach is not bound to any particular lexical or syntactic analysis technique. ModelCC models do not need to comply with the constraints imposed by particular parsing algorithms.

A fully-functional "proof of concept" implementation of ModelCC is soon to be released at the www.modelcc.org website.

In the future, ModelCC will incorporate a wider variety of parsing techniques so that it will be able to automatically determine the most efficient parsing technique that does not incur in ambiguities for processing a particular language.

ModelCC will also be extended in order to support multiple concrete syntaxes (for a single abstract syntax).

Finally, we also plan to study the use of ModelCC in different application domains, including model induction, natural language processing, text mining applications, data integration, and information extraction.

Acknowledgements. Work partially supported by research project TIN2009-08296.

References

1. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D.: Compilers: Principles, Techniques, and Tools, 2nd edn. Addison Wesley, Reading (2006)
2. Aho, A.V., Ullman, J.D.: The Theory of Parsing, Translation, and Compiling, Volume I: Parsing & Volume II: Compiling. Prentice Hall, Englewood Cliffs (1972)
3. Chomsky, N.: Three models for the description of language. IRE Transactions on Information Theory 2, 113–123 (1956)
4. Crescenzi, V., Mecca, G.: Automatic information extraction from large websites. Journal of the ACM 51, 731–779 (2004)

5. Earley, J.: An efficient context-free parsing algorithm. Communications of the ACM 26, 57–61 (1983)
6. Ehrig, H., Taentzer, G.: Graphical representation and graph transformation. ACM Computing Surveys 31(9) (1999)
7. Fowler, M.: Domain-Specific Languages. Addison-Wesley Signature Series (Fowler) (2010)
8. Hudak, P.: Building domain-specific embedded languages. ACM Computing Surveys 28, 196 (1996)
9. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition, 2nd edn. Prentice Hall, Englewood Cliffs (2009)
10. Kasami, T., Torii, K.: A syntax-analysis procedure for unambiguous context-free grammars. Journal of the ACM 16, 423–431 (1969)
11. Kats, L.C.L., Visser, E., Wachsmuth, G.: Pure and declarative syntax definition: paradise lost and regained. In: Proceedings of the ACM International Conference on Object Oriented Programming Systems Languages and Applications (OOPSLA 2010), pp. 918–932 (2010)
12. Klein, D., Manning, C.D.: Corpus-based induction of syntactic structure: Models of dependency and constituency. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL 2004), pp. 479–486 (2004)
13. Kleppe, A.: Towards the generation of a text-based IDE from a language metamodel. In: Akehurst, D.H., Vogel, R., Paige, R.F. (eds.) ECMDA-FA. LNCS, vol. 4530, pp. 114–129. Springer, Heidelberg (2007)
14. Knuth, D.E.: On the translation of languages from left to right. Information and Control 8, 607–639 (1965)
15. Levine, J.R., Mason, T., Brown, D.: Lex & yacc, 2nd edn. O'Reilly, Sebastopol (1992)
16. McManis, C.: Looking for lex and yacc for java? you don't know jack. JavaWorld (1996), http://www.javaworld.com/javaworld/jw-12-1996/jw-12-jack.html
17. Mernik, M., Heering, J., Sloane, A.M.: When and how to develop domain-specific languages. ACM Computing Surveys 37, 316–344 (2005)
18. Parr, T.J., Quong, R.W.: Antlr: A predicated-ll(k) parser generator. Software Practice and Experience 25, 789–810 (1995)
19. Porubän, J., Forgáč, M., Sabo, M.: Annotation based parser generator. In: Proceedings of the International Multiconference on Computer Science and Information Technology, vol. 4, pp. 705–712. IEEE Computer Society Press, Los Alamitos (2009)
20. Rosenkrantz, D.J., Stearns, R.E.: Properties of deterministic top-down grammars. Information and Control 17, 226–256 (1970)
21. Schmidt, D.C.: Model-driven engineering. IEEE Computer 39, 25–31 (2006)
22. Tan, P.-N., Kumar, V.: Introduction to Data Mining. Addison Wesley, Reading (2006)
23. Turmo, J., Ageno, A., Cataà, N.: Adaptive information extraction. ACM Computing Surveys 38(4) (2006)
24. Younger, D.H.: Recognition and parsing of context-free languages in time n^3. Information and Control 10, 189–208 (1967)

Typical Testors Generation Based on an Evolutionary Algorithm

German Diaz-Sanchez[1], Ivan Piza-Davila[2], Guillermo Sanchez-Diaz[3],
Miguel Mora-Gonzalez[1], Oscar Reyes-Cardenas[3], Abraham Cardenas-Tristan[3],
and Carlos Aguirre-Salado[3]

[1] Universidad de Guadalajara, Centro Universitario de los Lagos,
Av. Enrique Diaz de Leon 1144, Lagos de Moreno, Jal. Mexico, C.P. 47460
[2] Instituto Tecnologico y de Estudios Superiores de Occidente,
Periferico Sur Manuel Gomez Morin 8585, Tlaquepaque, Jal. Mexico, C.P. 45604
[3] Universidad Autonoma de San Luis Potosi, Facultad de Ingenieria
Dr. Manuel Nava 8, San Luis Potosi, SLP, Mexico, C.P. 78290
guillermo.sanchez@uaslp.mx

Abstract. Typical testors are useful for both feature selection and feature relevance determination in supervised classification problems. However, reported algorithms that address the problem of finding the set of all typical testors have exponential complexity. In this paper, we propose to adapt an evolutionary method, the Hill-Climbing algorithm, with an acceleration operator in mutation process, to address this problem in polinomial time. Experimental results with the method proposed are presented and compared, in efficiency, with other methods, namely, Genetic Algorithms (GA) and Univariate Marginal Distribution Algorithm (UMDA).

Keywords: Hill climbing, feature selection, optimization, typical testors.

1 Introduction

Feature selection is a significant task in supervised classification and other pattern recognition areas. It identifies those features that provide relevant information for the classification process. The problem of feature subset selection, has been treated in several ways including: metaheuristics [10] and multi-objective point of view [6], etc. Nevertheless, results found so far are not conclusive.

In Logical Combinatorial Pattern Recognition approach [5], feature selection is addressed using Testor Theory [4]. Yu. I. Zhuravlev [3, 18] introduced the concept of testor (also known as non-reducible descriptors) to pattern recognition problems. He defined a testor as a set of features that does not confuse objects descriptions belonging to different classes. This concept has been extended and generalized in several ways [4, 17].

In text mining, typical testors have been widely used for text categorization [8] and document summarization [7].

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 58–65, 2011.

The computation of all typical testors requires exponential time [16]. In general, two approaches have been developed to address this problem: a) generate the whole set of typical testors (LEX [13], fast-CT_EXT [11]); and b) find a subset of typical testors (GA [12], UMDA[1], AGHPIA [15]). Nevertheless, these global search heuristics become too slow as the number of features grows because the goal of these techniques focuses on finding the entire typical-testor set as global optimum. However, we consider that every typical testor can be treated as a local optimum.

Thus, this paper proposes to adapt a local-search evolutionary technique, the Hill Climbing algorithm, and to, incorporate an acceleration operator at the mutation step, to find a subset of the entire set of typical testors. The key idea in the method proposed is to iteratively generate typical testors as the algorithm explores the space search and explotes promising regions. The classic concept of testor, in which classes are assumed to be both hard and disjointed, is used. The comparison criteria used for all features are Boolean, regardless of the feature type (qualitative or quantitative). The similarity function used for comparing objects demands similarity in all features. These concepts are formalized in the following section.

2 Background

Let TM be a training matrix containing m objects, described in terms of n features $R = \{x_1, \cdots, x_n\}$ and distributed into c classes $\{k_1, \cdots, k_c\}$. Each feature $x_i \in R$ takes values in a set L_i, $i = 1, \cdots, n$.

A comparison criterion of dissimilarity $C_i : L_i \times L_i \rightarrow \{0, 1\}$ is associated to each x_i (0=similar, 1=dissimilar). Applying these comparison criteria for all possible pairs of objects belonging to different classes in TM, a Boolean dissimilarity matrix, denoted by DM, is built.

Let p and q be two rows of DM. We say that p is a subrow of q if: $\forall_j [q_j = 0 \Rightarrow p_j = 0]$ and $\exists_i [p_i = 0 \Rightarrow q_i = 1]$. A row p of DM is called basic if no row in DM is a subrow of p. The submatrix of DM containing all its basic rows (without repetitions) is called a basic matrix (BM).

The typical-testor set of a TM can bee obtained using DM or BM. A theorem showed in [9] proves that the set of all typical testors generated using either DM or BM is the same. Commonly, algorithms use BM instead of DM due to a substantial reduction of rows.

The characterization of a typical testor working with the basic matrix is then presented. The following concepts were taken from [9, 4, 18].

Proposition 1. *A feature subset $T = \{x_{i_1}, \cdots, x_{i_s}\}$ is a testor of TM, if and only if a whole row of zeros does not appear in the remaining submatrix of BM, after eliminating all columns corresponding to the features in $R \backslash T$.*

Remark 1. This means that T is a testor, if in each row of BM, in the columns $i_1, \cdots, i_s$ of BM there is at least a 1 (there are not rows of zeros in these columns of BM).

Proposition 2. *Let $T = \{x_{i_1}, \cdots, x_{i_s}\}$ a testor of TM. T is a typical testor of TM if there is no proper subset of T that meets a testor.*

Remark 2. Thus, each typical testor is of minimal length. That means that each typical testor cannot be reduced any more.

In terms of BM, proposition 2 can be rewritten as follows:

Proposition 3. *Let $T = \{x_{i_1}, \cdots, x_{i_s}\}$ a testor of TM. T is a typical testor of TM if and only if, for each column $i_1, \cdots, i_s$ of BM, there is at least a row in BM, such that in the column i_j, $j \in \{1, \cdots, s\}$ has a 1, and in the remaining columns i_p, $p \neq j$, $p \in \{1, \cdots, s\}$ are all zeros.*

Proof. If for each column $i_1, \cdots, i_s$ of BM, there is at least a row with only a 1 in a column i_j, $j \in \{1, \cdots, s\}$, and remaining columns are all zeros, then if any column i_j is removed of BM, a row of zeros appears in BM in the columns $i_1, \cdots, i_s$, and then T does not meet a testor. Thus, T is typical testor.

2.1 Hill Climbing Algorithm

The Hill-Climbing algorithm [14] is a local-search stochastic method which, in general, uses a bit string to represent either a set of prototypes or, in some experiments, a collection of features. Hill-Climbing can be considered as an evolutionary strategy with one individual which was intended to solve complex optimization problems arising from engineering design problems [2].

Consider the set: $P(R) = \{\emptyset, \{x_1\}, \cdots, \{x_n\}, \{x_1, x_2\}, \cdots, \{x_{n-1}, x_n\}, \cdots, \{x_1, \ldots, x_n\}\}$, where $P(R)$ is the power set of feature set R, and n is the cardinality of R. Now, consider the following set: $SS(R) = P(R) \setminus \{\emptyset\}$, where $SS(R)$ is the entire search space of the set R. Then, $SS(R)$ contains all the possible combinations of features that can be formed in R.

Let BM be a Basic Matrix obtained from a Training Matrix TM, and m_{BM} be the number of rows of BM. And let $Z = \{x_{i_1}, \cdots, x_{i_s}\}$, $Z \subseteq R$ and $Z \in SS(R)$.

We want to obtain a set Z such that minimizes the absolute value of performance index

$$J(Z) = 1 - (\sum_{p=1}^{m_{BM}} zr_p + \frac{1}{(\sum_{q=i_1}^{i_s} or_q) + 1}) \tag{1}$$

Where zr_p are the rows of zeros that contains BM in the columns $i_1, \cdots, i_s$, and or_q is the number of columns $i_1, \cdots, i_s$ of BM, such that for each column there is not a row in BM, such that the column i_j, $j \in \{1, \cdots, s\}$ is 1, and the remaining columns i_l, $l \neq j$, $l \in \{1, \cdots, s\}$ have all zeros.

Remark 3. Notice that for any feature subset Z, $-m_{BM} \leq J(Z) < 1$. If the performance index $J(Z)$ reaches the value 0, then Z is a typical testor (Z meets propositions 1 and 3). If $J(Z)$ is a positive value, then Z just a testor, but it is not typical testor (Z only meets proposition 1). Otherwise, if $J(Z)$ is negative, Z is not a testor (Z does not meets proposition 1).

Considering this problem of feature selection as a problem of location of zeros, the hill climbing algorithm is designed to obtain the feature subsets Z, such that the performance index $J(Z)$ proposed in this paper reaches a cero (i.e. to find a feature subset $Z \subseteq R$ and $Z \in SS(R)$, such that $J(Z) = 0$).

3 The Proposed Hill Climbing Algorithm for Generated Typical Testors

3.1 The Acceleration Operation

The proposed Hill-Climbing algorithm incorporates an acceleration operator at the mutation step. This operator improves the exploration capability of the mutation, being able to find a feature subset $Z = \{x_{i_1}, \cdots, x_{i_s}\}$ which meets the typical testor property, with a lower number of computations. The accelerator operator is independent to the mutation operator because the latter can be performed without the accelerator operator proposed, as is done in simple Hill Climbing algorithm.

This acceleration operator is applied differently. It depends on the performance index found and based on the behavior of the combination of feature subset, according to the following rules:

Rule 1. If $J(Z) = 0$ (Z is a typical testor), then one feature x_j, $j = i_1, \cdots, i_s$ is removed of Z. And, one feature x_p, $p \neq j$, $p = i_1, \cdots, i_s$ is added to Z.

Rule 2. If $J(Z) > 0$ (Z is a testor), then k_t-features x_j, $j = i_1, \cdots, i_s$ are removed of Z.

Rule 3. If $J(Z) < 0$ (Z is not a testor), then k_{nt}-features x_j, $j = i_1, \cdots, i_s$ are added to Z.

Remark 4. According to different experiments with several algorithms, we could observe that in many cases, two different typical testors, could be equated to perform a permutation of two features x_i, x_j, $i \neq j$ as follows: if $x_i = 1$ and $x_j = 0$ then set $x_i = 0$ and $x_j = 1$. This reasoning is applied to Rule 1.

Remark 5. If a feature subset Z is a testor, then Z does not satisfy proposition 3. This means that Z can be reduced, and some features can be removed from Z. In Rule 2, this reasoning is applied.

Remark 6. Finally, if a feature subset Z is not a testor, then Z does not satisfy proposition 1. Thus, Z needs more features to satisfy property 1, and some features can be added to Z. This reasoning is applied to Rule 3

The hill climbing algorithm has two parameters used to calculate the number of features to remove or add in Z, which are the mutation probability for a non testor and the mutation propability for a testor, respectively. Besides, the stop condition handled as the maximum number of iterations to be peerformed by the algorithm, or a number of typical testors to find. The algorithm is designed as follows:

Input: BM (basic matrix); Iter (number of iterations); NumTT (number of typical testors to find); p_t (mutation probability for a testor); p_{nt} (mutation probability for a non testor)

Ouput: TT (list of typical testor subset found)

1. *Prototypes representation and initialization.* A feature combination Z are encoded in an n-dimensional binary array as: $A = [a_1, \cdots, a_n]$, where each $a_j = 1$ means that feature x_j is present in Z. Otherwise, if $a_j = 0$ indicates the absence of feature x_j in Z.
   The performance index $J(Z)$ will be handled as the fitness value $F(A)$. Start from an empty list of typical testors TT; $Iter \leftarrow 1$.
2. *Initialization of array.* The components a_j of array A, are generated randomly. Call this array best-evaluated and calculate the fitness value $F(A)$ (i.e. the belonging performance index $J(Z)$ is obtained). If $F(A) = 0$ then, add A to the list TT.
3. *Mutation.* First, the values of mutated array are assigned as $A_{mut}(a_i) = A(a_i)$, $i = 1, \cdots, n$. Second, the value of some components a_j of mutated array are randomly mutates using a Uniform random variable, according to the rules defined below in the acceleration operator, using a procedure as follows: $Mutate(A_{mut}, F(A), p_t, p_{nt})$. Probabilities p_t, p_{nt}, are used according the value of $F(A)$.
4. *Fitness calculation.* Compute the Fitness of the mutated array A_{mut}, as $F(A_{mut})$. If $F(A) = 0$, verify if A is not in the list TT, to add it to the list.
5. *Compare the fitness obtained.* If $abs(F(A_{mut})) < abs(F(A))$, where $abs(F)$ indicates the absolute value of F, or if $F(A_{mut}) = 0$, then set the best-evaluated array to the mutated array $(A(a_i) = A_{mut}(a_i), i = 1, \cdots, n)$.
6. *Stop condition.* If the maximum number of iterations has been performed (if $Iter > MaxIter$), or typical testor number has been reached, then return the list of typical testors TT. Otherwise, go to step 3.

4 Experiments

The first experiment focuses on a comparison among the performance based on the number of evaluations to compute a specific number of typical testors found by a Genetic Algorithm [12], which denoted as GA. Besides, the Univariate Marginal Distribution Algorithm, published in UMDA [1], which denoted as UMDA, and in addition, a simple Hill Climbing algorithm denoted as HC, that does not incorporate the acceleration operator in the mutation step. And finally, the Hill Climbing algorithm proposed in this paper, which incorporates the acceleration operator in the mutation step, which denoted as HCTT. All experiments were conducted in a PC, with a Pentium IV 2Ghz processor, and 1 Mbyte of RAM.

Remark 7. This experiment is intended to compare the number of evaluations required by each algorithm to find a fixed amount of typical testors, as carried out in [12] and [1]. An evaluation involves all the required steps to determine whether a feature combination satises the property of being either a testor, a typical testor, or not a testor. The execution time of the algorithms is not included due to hardware variations.

Please note that we do not make comparisons with the GA published in [15], because the authors did not provide the algorithm to make comparisons with the proposal Hill Climbing algorithm. Besides, the only dataset presented in [15], has 29 features, which is regarded as relatively small Basic Matrix. The number of rows of Basic Matrix above is great. However, it is a factor that multiplies the complexity time of the algorithm, increasing their complexity time in a polynomial way. In contrast, the number of features grows exponentially.

A collection of four basic matrices described in [12] and [1] was tested. For this case, the set of parameters was: $p_t = 0.2$ and $p_{nt} = 0.01$. Experimentally, the best results were obtained with these parameters, after testing with different values from them. The results are shown in table 1. In this table, EV represents the number of evaluations carried out by the algorithm. The dimensions of the matrices are expressed as $rows \times columns$. The number of typical testors to find by compared algorithms is denoted as TTF.

Table 2 shows the comparison between HCTT and the deterministic algorithm fast-CT_EXT [11]. For this case, a collection of six matrices described in [12] and [1], and a new basic matrix with a considerable number of features were tested. For the first five matrices, the number of all typical testor found is know, because fast-CT_EXT calculates this set in a relative small time. For the remaining two matrices, the number of all typical testor found is unknow, because fast-CT_EXT works more than two days and does not finish. TIME, TTF and EV are handled in the same way as in Table 1.

We carried out 1 000 000 and 10 000 000 iterations respectively, to verify the computational complexity growth factor, as well as the proportion of typical testors found, when the number of iterations carried out by the algorithm is increased.

4.1 Discussion

In the first experiment, the execution time of HCTT ranged from 2 to 13 seconds. In all cases, the number of evaluations required by the proposed algorithm (which

Table 1. Comparison of GA, UMDA, simple HC and the HCTT performance

Matrices	**TTF**	EV-GA	EV-UMDA	EV-HC	EV-HCTT
1215x105	**105**	22 500 000	336 700	718 356	8 933
269x42	**318**	5 000 000	89 800	138 564	11 036
40x42	**655**	1 400 000	142 500	210 879	30 813
209x47	**1967**	5 000 000	706 900	558 530	80 066

Table 2. Comparison of fast-CT_EXT and the HCTT performance

	fast-CT_EXT		EV-HCTT = 1 000 000		EV-HCTT = 10 000 000	
Matrices	TTF	TIME	TTF	TIME	TTF	TIME
40x42	8 963	0	2 991	106	5 387	1 147
80x42	32 277	2	5 669	117	11 035	1 024
110x42	65 299	6	8 200	127	19 849	1 286
269x42	302 066	120	11 335	174	38 407	1 837
209x47	184 920	72	7 820	149	20 658	1 620
215x105	?	> 2 days	11 166	809	79 467	9 252
300x300	?	> 2 days	3	552	54	5 575

can be considered as a constant-time process) is significantly lower than that from the compared algorithms.

Table 2, shows that deterministic algorithms that compute typical testors are not suitable when matrices with a great number of features are processed (for example, hyperspectral images consisting by 256 bands). Otherwise, the proposed hill climbing algorithm was developed to process data sets with a great number of features in training matrix (with 60 features or more).

5 Conclusions

A new Hill Climbing algorithm that incorporates an acceleration operation for generating typical testor of a training matrix is introduced. This acceleration operator had a powerful effect in reducing the number of computations on this process. The superior performance of the proposed algorithm over the Genetic algorithm reported in [12], the UMDA published in [1], and a simple Hill Climbing shown in this paper is experimentally demonstrated. The Hill Climbing algorithm with the acceleration operator generates the same number of typical testors as the reported heuristics, but with a fewer number of evaluations and with significantly less time.

References

[1] Alba, E., Santana, R., Ochoa, A., Lazo, M.: Finding Typical Testors By Using an Evolutionary Strategy. In: Proc. of V Iberoamerican Workshop on Pattern Recognition, Lisbon, Portugal, pp. 267–278 (2000)

[2] De Jong, K.: Evolutionary computation, vol. 1, pp. 52–56. John Wiley & Sons, Inc., Chichester (2009)

[3] Dmitriev, A., Zhuravlev, I., Krendeliev, F.: About mathematical principles and phenomena classification. Diskretni Analiz (Rusia) 7, 3–15 (1966)

[4] Lazo-Cortes, M., Ruiz-Shulcloper, J., Alba-Cabrera, E.: An Overview of the evolution of the concept of testor. Pattern Recognition 34(4), 753–762 (2001)

[5] Martinez-Trinidad, J., Guzman-Arenas, A.: The Logical Combinatorial approach for pattern recognition. An overview through selected Works. Pattern Recognition 4, 741–751 (2001)

[6] Mierswa, I., Michael, W.: Information Preserving Multi-Objective Feature Selection for Unsupervised Learning. In: Proc. of the Genetic and Evolutionary Computation Conference, pp. 1545–1552. ACM Press, New York (2006)

[7] Pons-Porrata, A., Ruiz-Shulcloper, J., Berlanga-Llavori, R.: A Method for the Automatic Summarization of Topic-Based Clusters of Documents. In: Sanfeliu, A., Ruiz-Shulcloper, J. (eds.) CIARP 2003. LNCS, vol. 2905, pp. 596–603. Springer, Heidelberg (2003)

[8] Pons-Porrata, A., Gil-García, R.J., Berlanga-Llavori, R.: Using Typical Testors for Feature Selection in Text Categorization. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 643–652. Springer, Heidelberg (2007)

[9] Ruiz-Shulcloper, J., Guzman-Arenas, A., Martinez-Trinidad, J.: Logical combinatorial pattern recognition approach. Centro de Investigacion en Computacion, IPN, Mexico City, Mexico (1999)

[10] Saeys, Y., Degroeve, S., Van de Peer, Y.: Digging into acceptor splice site prediction: An iterative feature selection approach. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 386–397. Springer, Heidelberg (2004)

[11] Sanchez-Diaz, G., Piza-Davila, I., Lazo-Cortes, M., Mora-Gonzalez, M., Salinas-Luna, J.: A Fast Implementation of the CT_EXT Algorithm for the Testor Property Identification (6438). In: Sidorov, G., Hernández Aguirre, A., Reyes García, C.A. (eds.) MICAI 2010, Part II. LNCS (LNAI), vol. 6438, pp. 92–103. Springer, Heidelberg (2010)

[12] Sanchez-Diaz, G., Lazo-Cortes, M., Fuentes-Chavez, O.: Genetic algorithm for calculating typical testors of minimal cost. In: Proc. of the Iberoamerican Symposium on Pattern Recognition (SIARP 1999), pp. 207–213 (1999)

[13] Santiesteban-Alganza, Y., Pons-Porrata, A.: LEX: A new algorithm for calculating typical testors. Revista Ciencias Matematicas (Cuba) 21(1), 85–95 (2003)

[14] Schuetze, O., Lara, A., Coello, C.: Evolutionary continuation methods for optimization problems. In: Proc. of the Genetic and Evolutionary Computation Conference, pp. 651–658 (2009)

[15] Torres, D., Ponce-de-León, E., Torres, A., Ochoa, A., Díaz, E.: Hybridization of evolutionary mechanisms for feature subset selection in unsupervised learning. In: Aguirre, A.H., Borja, R.M., Garciá, C.A.R. (eds.) MICAI 2009. LNCS (LNAI), vol. 5845, pp. 610–621. Springer, Heidelberg (2009)

[16] Valev, V., Asaithambi, A.: On computational complexity of non-reducible descriptors. In: Proc. of the IEEE Int. Conf. on Information Reuse and Integration, pp. 208–211 (2003)

[17] Valev, V., Sankur, B.: Generalized non-reducible descriptors. Pattern Recognition 37(9), 1809–1815 (2004)

[18] Valev, V., Zhuravlev, Y.: Integer-valued problems of transforming the training tables in k-valued code in pattern recognition problems. Pattern Recognition 24(4), 283–288 (1991)

SVM Approach to Classifying Lesions in USG Images with the Use of the Gabor Decomposition

Marcin Ciecholewski

Institute of Computer Science, Jagiellonian University,
ul. Łojasiewicza 6, 30-348 Kraków, Poland
marcin.ciecholewski@uj.edu.pl

Abstract. The article presents the application of the support vector machines (SVM) method to recognise gallbladder lesions such as lithiasis and polyps in USG images. USG images of the gallbladder were first processed by the histogram normalisation transformation to improve their contrast, and the gallbladder shape was segmented using active contour models. Then the background area of uneven contrast was eliminated from images. To extract features from the images to be classified, the Gabor decomposition was applied to a plane presented in a log-polar system. In the best case, the SVM achieved the accuracy of 82% for all lesions, 85.7% for lithiasis and 74.3% for polyps.

1 Introduction

10–15% of the adult population of Europe and the US is estimated to have gallstones in their gallbladders [13]. Another widespread disease is the presence of polyps in the gallbladder, including cancerous ones, which are found in 5% of the global population [10].

This data supports the need to constantly improve the effectiveness of diagnosing diseases of this organ and thus contribute to eliminating diagnostic errors. All the more so, as for the gallbladder there are no ready, practical solutions to help the physician in their work. Consequently, in this publication the Support Vector Machines (SVM) method was used for recognising gallstones and polyps in USG images. Fig. 1(a) shows an image without lesions, while Fig. 1 (b) shows an image with polyp. In Fig. 1(c) a lithiasis is visible.

The SVM method is very capable of generalising the knowledge it gains [16,17] which leads to its increasingly frequent use. SVMs are good in generalising the knowledge gained, even for multi-dimensional data with relatively little training data. In addition, SVM performance is good even without any *a priori* knowledge of the problem under consideration: for instance, in many applications where learning machines are recognising images, the same free permutation of pixels in all images does not change the training results for SVMs [1]. The traditional approach, where artificial neural networks are used for classifying, can frequently yield results not good enough, due to overtaining, which has a negative impact on the ability of the learning machine to generalise the knowledge

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 66–75, 2011.

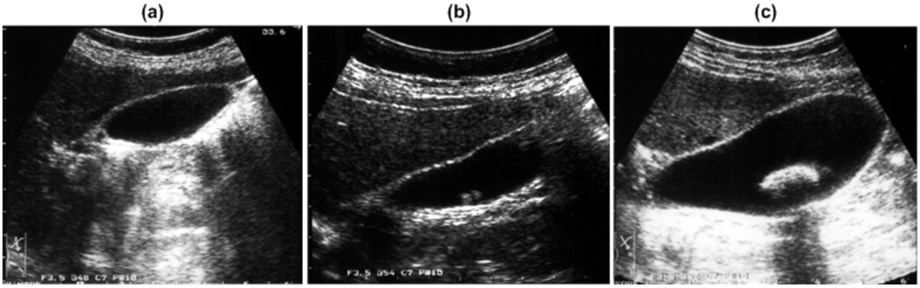


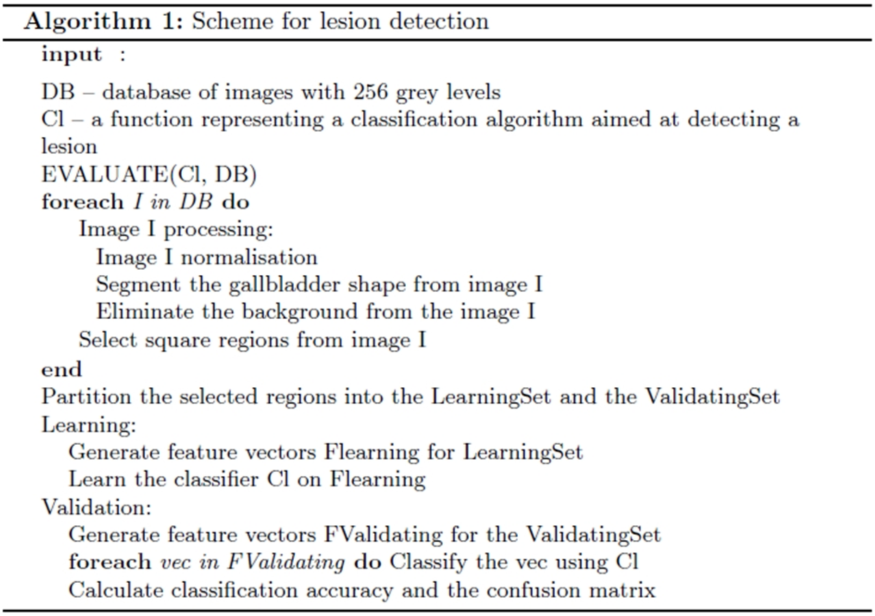
Fig. 1. Examples USG images of the gallbladder. (a) An image of a gallbladder free of lesions. (b) An image showing a polyp inside the gallbladder. (c) An image with visible cholecystolithiasis (gallstone).

gained [9]. An SVM classifier has already been used to analyse medical images, e.g. SPECT (Single Proton Emission Computed Tomography) images to distinguish Alzheimer disease lesions from healthy cases (free of lesions). The results achieved by the SVM classifier used are as follows: over 90% accuracy in publication [14] and 84.4% sensitivity with 90.9% specificity, as presented in article [6]. The SVM results described in publication [14] were achieved for a linear kernel function and a contiguous linear SVM classifier [6], while in this publication the best results, e.g. the accuracy of 85.7%, were obtained for the Gaussian kernel function.

This article is structured as follows. Section 2 describes the designed system model which uses support vector machines for classifying lesions like lithiasis and polyps. Section 3 shows the processing of input USG gallbladder images. Section 4 presents the classification method based on support vector machines. Section 5 is about using the modified bank of Gabor filters to extract image features to be classified. The following section assesses the SVM algorithm based on experiments completed and contains selected research results. The last section provides a summary.

2 System Description

The design of a system enabling lesions of the gallbladder such as lithiasis and polyps to be detected is based on a machine learning method using the SVM classifier. The general structure is presented in Algorithm 1. The methods presented in Algorithm 1 comprise several steps. First, every USG image is pre-processed by segmenting the shape of the gallbladder and eliminating the area of uneven contrast background. The gallbladder shape is segmented using active contour models [3,4]. In the second step, a set of rectangular areas is selected from every available image from the database of gallbladder USG images. Selected areas are partitioned into patterns designated for learning and validating. Then, a set of features is distinguished for every pattern. Features from the learning set are designated for training the algorithm which will then be able to take a binary decision about the image features supplied to it. After the learning process, the classifier is validated using features of the image from the validating set in order to evaluate the accuracy of the classifier. In particular, a confusion matrix with the dimensions 2 × 2: $A_{i,j}$ is calculated. The element $a_{i,j}$ of this matrix represents the number of samples belonging to class i, provided that the classification algorithm decides that it belongs to class j, $(i, j \in \{1 - lesion, 0 - nonlesion\})$. As finding a lesion is treated as a positive diagnosis, while the lack of lesions as a negativeone, the confusion matrix may contain four possible values determining the classifier's behaviour. These are as follows: true positive ratio specified as the $\frac{a_{1.1}}{a_{1.0}+a_{1.1}}$, true negative ratio $\frac{a_{0.0}}{a_{0.0}+a_{0.1}}$, false positive ratio $\frac{a_{0.1}}{a_{0.0}+a_{0.1}}$ and false negative ratio $\frac{a_{1.0}}{a_{1.0}+a_{1.1}}$. The true positive ratio is also referred to as the sensitivity of the classifier, while the true negative as its specificity. Finally, the overall accuracy, or the success rate of the classifier can be quantified as $\frac{a_{1.1}+a_{0.0}}{a_{0.0}+a_{0.1}+a_{1.0}+a_{1.1}}$.



```
Algorithm 1: Scheme for lesion detection
input :
DB – database of images with 256 grey levels
Cl – a function representing a classification algorithm aimed at detecting a
lesion
EVALUATE(Cl, DB)
foreach I in DB do
    Image I processing:
      Image I normalisation
      Segment the gallbladder shape from image I
      Eliminate the background from the image I
    Select square regions from image I
end
Partition the selected regions into the LearningSet and the ValidatingSet
Learning:
   Generate feature vectors Flearning for LearningSet
   Learn the classifier Cl on Flearning
Validation:
   Generate feature vectors FValidating for the ValidatingSet
   foreach vec in FValidating do Classify the vec using Cl
   Calculate classification accuracy and the confusion matrix
```

3 Analysed Data

An image database obtained from the Image Diagnostics Department of the Gdańsk Regional Specialist Hospital, Poland, was used in the research on USG image analysis. The database of images used in this research contained 800 images, including 600 images without lesions and 200 images containing them, specifically 90 images showing polyps and 110 images depicting gallbladder stones. USG images of the gallbladder were processed by the histogram normalisation transformation to improve their contrast, and the gallbladder shape

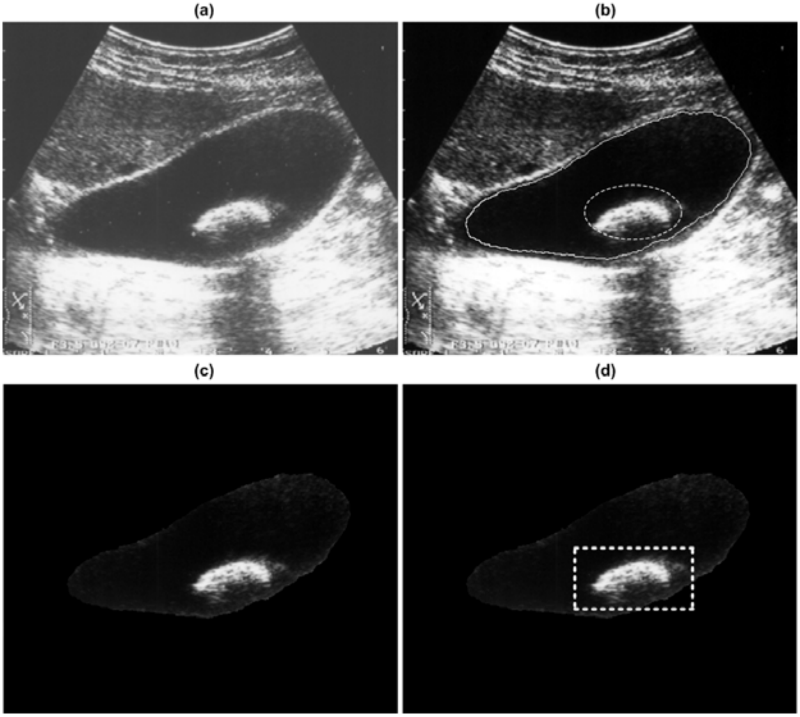


Fig. 2. Pre–processing of the input USG image of the gallbladder. (a) Input image obtained from the USG image database. (b) An image with improved contrast after the histogram normalisation transformation and the edge marked using the active contour method. The dashed line shows manually initiated contour inside the gallbladder shape. (c) A USG gallbladder image with the background eliminated. (d) The marked rectangular image fragment with the gallstone in the centre of the region.

was segmented using active contour models [3,4]. Then the background area of uneven contrast was eliminated from images. The entire 800 database of gallbladder images was processed in this fashion. For illustration, Figure 2(a) shows an unprocessed USG image depicting lithiasis from the database provided, while Figure 2(b) is an image after the histogram normalisation transformation, with the approximate edge marked. The contour was initiated manually inside the gallbladder shape and on the outside of the single lesion or several lesions present, as illustrated (dashed line) in Figure 2(b). Figure 2(c) is an image of a gallbladder from which the uneven background of the image was eliminated by setting the black colour to pixels located outside the contour. For images showing lesions, a rectangular bounding region containing the disease was captured. Then, the centre of the lesion was approximated, after which the rectangular region was cropped so as to contain the lesion at its centre. An example is shown in Fig. 2(d). These transformations generated a set of rectangular regions of various dimensions, every one of which contained a lesion in the centre. The above procedure made it possible to obtain samples of image regions containing lesions. To obtain samples without lesions, an region equal in size to that obtained for a lesion was delineated in images from healthy individuals. As there are more images without lesions, for every sample with a lesion there were 3 samples identified with the same area free of lesions, coming from different patients. The set of 200 samples with lesions, containing 90 of polyps and 110 of stones and 600 samples from images without lesions was divided into two sets:

- The training set 300 of samples, containing 200 samples without lesions and 100 samples with lesions (60 of stones and 40 of polyps).
- The validation set 500 of samples, containing 400 samples without lesions and 100 samples with lesions (50 of stones and 50 of polyps).

The number of cases with lesions in the validating set represents 25% of all samples in that set. The SVM classifier handles rectangular input image areas of a specific length and uses responses of Gabor filters as an input features.

4 Support Vector Classification

Let's consider data in the form of vectors from space R^n belonging to two classes, to which we respectively assign numbers from set $\{-1, 1\}$:

$$D = \{(x_1, y_1), \dots, (x_n, y_n), x_i \in R^n, y_i \in \{-1, 1\}, i = 1 \dots m\} \tag{1}$$

The support vector machine algorithm separates classes of input patterns with the hyperplane for which the distance between the nearest vectors of these two classes given by relationship (1) is the greatest. This hyperplane is called the optimal separating hyperplane. It is defined as follows: $H : f(x) =< w, x > +b$. Where x is a vector from the R^n space, w is the vector perpendicular to hyperplane H, and the value $\frac{b}{\|w\|}$ determines the offset from the beginning of the coordinate system. Finding the optimal hyperplane for which the distance of the nearest vectors from two different classes is the greatest is equivalent to minimising the square of the following norm [9]. : $\frac{1}{2}\|w\|^2$. The case of linear data separation can be generalised to data separation with a non–linear function. For this purpose, data vectors from the R^n space are mapped to a new Hilbert space H using a certain mapping function $\phi : R^n \rightarrow H$. As the data appears only in the form of scalar products, the form of the ϕ function does not have to be given explicitly, the following K kernel function can be used

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_i) \tag{2}$$

Points belonging to two classes (1) cannot be linearly separable. In such cases, one of the solutions may be to slacken the condition that there must be no points between pattern classes. For this purpose, slack variables $\xi \geq 0$ are introduced. The constraining conditions will then take the following form:

$$y_i \cdot [< w, x_i > +b \geq 1 - \xi_i], \quad \xi_i \geq 0, \quad i = 1 \dots n \tag{3}$$

where $y_i \in \{-1, 1\}$ denotes the appropriate labels of classes of input patterns x_i in accordance with the relationship (1). As a result, the SVC classification is equivalent to a minimisation problem with the following objective function: $\frac{1}{2}\|w\|^2 + C\Big[\sum_{i=1}^{n} \xi_i\Big]^m$ where $m = 1$ is usually assumed. Constant C valuates the amount of the penalty for the wrong classification of training data. Using the Lagrange multipliers technique the optimisation problem formulated above can be transformed as follows:

$$L_D(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad \text{and} \quad L_D(\alpha) \to min$$
$$\text{under the conditions:} \quad 0 \le \alpha_i \le C \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{4}$$

In the above relationship, vector $\alpha = [\alpha_1, \alpha_2, \dots \alpha_n]$ defines non-negative Lagrange coefficients. It should be noted that separating the pattern classes makes it possible to use the kernel function $K(.,.)$. This is possible, because mapping ϕ satisfies equation (2). Non-negative Lagrange coefficients which constitute solutions to equation (6) may be used in the following decision-making function:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \quad \text{whereas:} \quad b = y_i - \sum_{j=1}^{n} \alpha_j y_j K(x_j, x_i) \tag{5}$$

The solution to equation (4) can be determined using general quadratic programming methods. In addition, dedicated heuristic methods [12] have been developed which effectively solve many classes of problems.

5 Log-Polar Sampling of the Gabor Decomposition

Gabor filters are very useful to extract features from images. These filters were originally proposed in [7] as an elementary Gabor function, and were then extended in [8] into a two–dimensional operator used in analysing digital images. A two–dimensional Gabor filter is defined as:

$$G(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-1/2(x^2/\sigma_x^2 + y^2/\sigma_y^2)} e^{j2\pi v_0 x} \tag{6}$$

This filter is a sinusoidal wave plane modulated by a Gaussian operator and is sensitive to image details which correspond to frequencies close to v_0 on the Fourier plane. Parameters σ_x and σ_y represent widths along the X and the Y axes, respectively. As the wave vector of this filter is parallel to the X axis, this filter is sensitive to parallel details in the image. However, to create a filter sensitive to image details displaced by a $\phi \neq 0$ angle, it is enough to rotate the original filter presented in equation (6). In publication [15], modified Gabor filters cast for log-polar coordinates are used:

$$G(r, \phi) = Ae^{-(r-r_0)^2/2\sigma_r^2} e^{-(\phi-\phi_0)^2/2\sigma_\phi^2} \tag{7}$$

where:

$$r = log\sqrt{v^2 + \mu^2} \quad \text{and} \quad \phi = arctan(\frac{\mu}{v}) \tag{8}$$

The parameters σ_r and σ_ϕ control a radius-axis width and an angle-axis width of the Gaussian function. This approach is useful in building filter banks for various orientations of the ϕ and central frequencies v_0, especially as filters defined by equation (7) do not overlap at low frequencies. To the contrary, the structure based on [7] requires values to be carefully selected for filters with a low frequency. Finally, publication [15] proposes a log-polar spatial grid to sample the response of a bank of Gabor filters. This grid consists of points located on several circles with logarithmically spaced radii (see Fig.3). To calculate the feature vector for a given point of an X image, this image is filtered with the bank of modified Gabor filters, and the magnitudes of responses are sampled using a grid with the central point X.

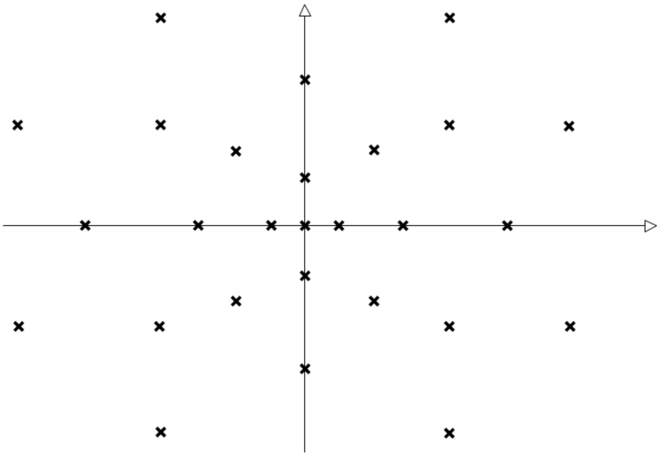


Fig. 3. The grid used to sample the responses of a Gabor filter bank, as proposed in publication [15]. To sample responses for a set point **p**, the centre of the grid is located in point **p**. Then, magnitudes of complex responses for the filter bank are calculated in every point of the grid. The result is a vector whose coordinates are the magnitudes of filter responses collected from all grid points, in a defined order.

6 SVM Algorithm Assessment Based on Completed Experiments and Selected Research Results

This section contains the research results and an assessment of the SVM algorithm based on the experiments completed. The tests were carried out on data sets presented in Section 3.

6.1 Grid Sampling Parameters on a Log–Polar Scale

Image feature vectors were extracted using a log–polar grid of 51 points. These points formed parts of six circles with logarithmically spaced radii from a 5–40 point range. The Gabor filter bank used in the sampled grid consisted of 20 filters, 85×85 points in size. These filters were arranged into four logarithmically-spaced frequency channels and five uniformly spaced orientation channels. The lowest normalised frequency found in the filter bank was $\frac{1}{7}$, and the highest $\frac{1}{2}$. The orientation channels covered the entire spectrum, e.g. from 0 to $\frac{4}{5}\pi$.

6.2 The Stage of Learning Data Sets and Selected Experiment Results

The SVM classifier was evaluated independently for three kernel functions:

- linear kernel: $k(x, y) =< x, y >$
- polynomial kernel of order 4: $k(x, y) = (\sigma < x, y > +\gamma)^4$
- Gaussian RBF kernel: $k(x, y) = e^{-\|x-y\|/2\sigma^2}$

In order to select the most suitable values of the misclassification penalty for wrong classification and the parameters of kernel functions, the specificity parameter was evaluated for every kernel function, using a learning set and employing the following values of the misclassification penalty for wrong classifications: $C \in \{1, 2, \ldots, 60\}$. For the polynomial and Gaussian kernel functions, the following range of parameter values was assessed: $\sigma, \gamma \in \{0.1, 0.2, \ldots, 15.0\}$. Then, for every kernel function, the value of the misclassification penalty for wrong classification was selected for which the sensitivity factor was the highest. The misclassification penalty values so obtained were used in further research. The results from testing the selection of parameters are summarised in Table 1.

Table 1. Values of kernel function parameters and the misclassification penalties used in the SVM classifier

Kernel function	Par. of the kernel function	Misc. penalty
Linear	–	$C = 10.0$
Polynomial	$\sigma = 0.7, \gamma = 0.5$	$C = 4.0$
Gaussian function RBF	$\sigma = 2.0$	$C = 8.0$

In order to determine the total accuracy, sensitivity and specificity of the classification, the classifier was assessed for data from the validating set. Results for various combinations of kernel functions are presented in Table 2. The experiments conducted show that the greatest sensitivity, specificity and accuracy were achieved if the Gaussian RBF kernel function was used. In order to evaluate the operation of the SVM classifier when recognising individual lesions, two

Table 2. Results of the SVM classifier in different configurations for a validation sets

Kernel function	**Sensitivity(%)**	**Specificity(%)**	**Accuracy(%)**
Linear	62%	64.3%	63.4%
Polynomial	68%	65.6%	67.3%
Gaussian function RBF	80.2%	81.3%	82%
Test 1. Samples containing 100 lesions and 400 free of lesions			
Linear	65.4%	68.2%	67.1%
Polynomial	72.5%	70.7%	73%
Gaussian function RBF	83.7%	82.1%	85.7%
Test 2. Samples containing 50 lithiasis and 400 free of lesions			
Linear	58.2%	60.3%	61.5%
Polynomial	65.2%	62.4%	64.1%
Gaussian function RBF	71%	72.7%	74.3%
Test 3. Samples containing 50 polyps and 400 free of lesions			

additional validations were conducted. The results of these validations are presented in Table 2, respectively, Test 2 and Test 3. The SVM classifier exhibited clearly greater accuracy, specificity and sensitivity when recognising stones than polyps. This can be attributed to the nature of the polyps presented in USG images which are highly non–uniform and not as well localized as gallstones. In the best case, the algorithm achieved the accuracy of 85.7% for lithiasis and of 74.3% when classifying polyps.

7 Summary and Further Research Directions

Experiments conducted for the defined training sets and the three applied kernel functions showed that the SVM classifier was more sensitive when recognising lithiasis than polyps. In classifying lithiasis and polyps, the best sensitivity and accuracy was obtained for the Gaussian kernel function. Classifier accuracy and sensitivity reached 85.7% and 83.7%, respectively, for lithiasis, and dropped to 74.3% and 71% for polyps. The reason for this drop in accuracy and sensitivity is the much poorer recognition of polyps by the SVM classifier. The research completed shows that the image features used are the main reasons limiting the further improvement of recognition accuracy. Further research will focus on developing specialised methods for analysing USG gallbladder images, making it possible to extract image features and thus improve the accuracy of recognising lesions, particularly polyps.

Acknowledgements. This research was financed with state budget funds for science for 2009-2012 as research project of the Ministry of Science and Higher Education: N N519 406837.

References

1. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 955–974 (1998)
2. Burges, C.J.C., Crisp, D.: Uniqueness of the SVM solution. In: Advances in Neural Information Processing Systems, vol. 12, pp. 223–229. MIT Press, Cambridge (1999)
3. Ciecholewski, M.: Gallbladder Segmentation in 2-D Ultrasound Images Using Deformable Contour Methods. In: Torra, V., Narukawa, Y., Daumas, M. (eds.) MDAI 2010. LNCS (LNAI), vol. 6408, pp. 163–174. Springer, Heidelberg (2010)
4. Ciecholewski, M.: Gallbladder Boundary Segmentation from Ultrasound Images Using Active Contour Model. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 63–69. Springer, Heidelberg (2010)
5. Ciecholewski, M.: Support Vector Machine Approach to Cardiac SPECT Diagnosis. In: Koroutchev, K. (ed.) IWCIA 2011. LNCS, vol. 6636, pp. 432–443. Springer, Heidelberg (2011)
6. Fung, G., Stoeckel, J.: SVM feature selection for classification of SPECT images of Alzheimer's disease using spatial information. Knowledge and Information Systems 11(2), 243–258 (2007)
7. Gabor, D.: Theory of communications. J. Int. Electr. Eng. 93, 427–457 (1946)
8. Granlund, G.: In search of a general picture processing operator. Comp. Graph. Image Proc. 8, 155–173 (1978)
9. Gunn, S.: Support vector machines for classification and regression. Technical Report. Dept. of Electronics and Computer Science. University of Southampton, Southampton, U.K (1998)
10. Myers, R.P., Shaffer, E.A., Beck, P.L.: Gallbladder polyps: Epidemiology, natural history and management. Can. J. Gastroenterol. 16(3), 187–194 (2002)
11. Osuna, E., Freund, R., Girosi, F.: An improved training algorithm for support vector machines. In: Principe, J., Gile, L., Morgan, N., Wilson, E. (eds.) Neural Networks for Signal Processing VII, pp. 276–285. IEEE, New York (1998)
12. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C.J., Smola, A.J. (eds.) Advances in Kernel Methods – Support Vector Learning, pp. 185–220. MIT Press, Cambridge (1999)
13. Portincasa, P., Moschetta, A., Palasciano, G.: Cholesterol gallstone disease. Lancet 368, 230–239 (2006)
14. Salas-Gonzalez, D., Górriz, J.M., Ramírez, J., Segovia, F., Chaves, R., López, M., Illán, I.A., Padilla, P.: Selecting regions of interest in SPECT images using wilcoxon test for the diagnosis of alzheimer's disease. In: Graña Romay, M., Corchado, E., Garcia Sebastian, M.T. (eds.) HAIS 2010. LNCS, vol. 6076, pp. 446–451. Springer, Heidelberg (2010)
15. Smeraldi, F., Carmona, O., Bigün, J.: Saccadic search with Gabor features applied to eye detection and real-time head tracking. Image Vision Comp. 18(4), 323–329 (2000)
16. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
17. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, Inc., New York (1998)

Online Labelling Strategies for Growing Neural Gas

Oliver Beyer and Philipp Cimiano

Semantic Computing Group, CITEC, Bielefeld University
obeyer@cit-ec.uni-bielefeld.de
http://www.sc.cit-ec.uni-bielefeld.de

Abstract. Growing neural gas (GNG) has been successfully applied to unsupervised learning problems. However, GNG-inspired approaches can also be applied to classification problems, provided they are extended with an appropriate labelling function. Most approaches along these lines have so far relied on strategies which label neurons a posteriori, after the training has been completed. As a consequence, such approaches require the training data to be stored until the labelling phase, which runs directly counter to the online nature of GNG. Thus, in order to restore the online property of classification approaches based on GNG, we present an approach in which the labelling is performed online. This online labelling strategy better matches the online nature of GNG where only neurons – but no explicit training examples – are stored. As the main contribution, we show that online labelling strategies do not deteriorate the performance compared to offline labelling strategies.

Keywords: artificial neural networks, growing neural gas, clustering, classification, labeling.

1 Introduction

Self-organising approaches such as self-organising maps (SOM) [5], learning vector quantization (LVQ) [6], or neural gas (NG) [11] are often successfully used in unsupervised learning tasks, clustering in particular. The advantage of such neurally-inspired clustering approaches lies in their ability to learn the representation of a feature space without supervision. A further interesting property is the fact that they typically perform dimensionality reduction. Typical applications of these learning paradigms can be found in areas such as text mining [3,12,7] or pattern recognition [10]. Growing neural gas (GNG) [2] represents an extension of the NG algorithm in which the number of neurons is not fixed a priori as in NG, but grows over time. This is especially interesting in such clustering tasks where the number of clusters is previously unknown.

In order to apply self-organising approaches to classification problems, two extensions are necessary: i) a function which assigns labels to neurons and ii) a function that performs the prediction on unseen data points. So far, mainly offline labelling strategies which require the explicit storage of labelled training data have been considered for the first case [3,12,8,9,10]. They perform the

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 76–83, 2011.

assignment of labels to neurons a posteriori, after the training phase has been ended. There are several disadvantages connected to these strategies. For example, offline labelling strategies assume that there is a definite end of the training phase after which the labelling can be performed. This is of course not given in life-long learning scenarios in which training and prediction are interleaved. Using offline labelling strategies, we are not able to perform predictions for unseen examples in every iteration step, which is the crucial characteristic of online classification approaches. In the area of cognitive systems engineering, the online nature of learning processes is crucial in order to learn complex behaviours incrementally and continuously [14]. Another disadvantage of offline labelling strategies is the fact that they directly run counter to the online nature of GNG, as training examples are stored explicitly.

In this paper we thus present and evaluate several strategies allowing to perform the labelling on-the-fly, thus extending GNG to an online classification algorithm. We compare these strategies to several offline strategies that have been proposed in the literature and examine in particular whether an online labelling strategy can compete in terms of performance, i.e. classification accuracy, with an a posteriori labelling strategy. In fact, we show on three data sets (one artificial and two standard data sets) that an online labelling strategy does not perform significantly worse compared to an offline labelling strategy.

We offer the following contributions in particular:

- We systematically evaluate different offline labelling strategies for GNG in a classification task.
- We extend GNG by an on-the-fly labelling step that allows us to extend GNG to an online classification algorithm.
- We present and systematically analyse various online labelling strategies and compare them to the offline labelling strategies, showing that they do not deteriorate the performance of a classification approach based on GNG.

The paper is structured as follows: in Section 2 we discuss how GNG can be extended to a classification algorithm and what labelling functions are typically used in the literature. In Section 3 we present our extension of GNG to an online classification algorithm, relying on a novel set of labelling functions which perform labelling on-the-fly. In Section 4, we present our experimental results obtained on the basis of three data sets. We discuss related work and conclude in Section 5.

2 Classification with GNG

In this article we rely on the original version of GNG introduced by Fritzke [2] which is an extension of NG [11]. The algorithm is depicted in Figure 1 (without step 4). The GNG algorithm initially starts with a small network of two neurons in step 1. In steps 2-8, a winner neuron and its neighbouring neurons are determined and adapted according to the presented input example (stimulus). In steps 9 and 10, neurons are removed and inserted into the network according to a fixed set of parameters. The algorithm stops when a predefined criterion is met, e.g. when the network has reached a certain size.

In order to apply GNG to a classification task, most approaches in the literature extend the algorithm by two functions. A neuron labelling function $l : N \rightarrow C$ where C is the set of class labels, and a prediction function $l : D \rightarrow C$ where D is the input space. We analyse the following offline neuron labelling functions as proposed by Lau et al. [9]. They are offline in the sense that they assume that the pairs (d, l_d) with $d \in D_{train} \subseteq D$ and $l_d \in C$ seen in the training phase are explicitly stored:

- *Minimal-distance method (min-dist):* According to this strategy, neuron n_i adopts the label l_d of the closest data point $d \in D_{train}$:

$$l(n_i) = l_d = l(\arg \min_{d \in D_{train}} |n_i - d|^2)$$

- *Average-distance method (avg-dist):* According to this strategy, we assign to neuron n_i the label of the category c that minimises the average distance to all data points labelled with category c:

$$l(n_i) = \arg\min_c \sum_{k=1}^{|D(c)|} \frac{|n_i - d_k|^2}{|D(c)|}$$

  where $D(c) = \{d \in D_{train} \mid l(d) = c\}$ is the set of all examples labelled with c.

- *Voronoi method (voronoi):* According to this strategy, we label neuron n_i with that category c having the highest overlap (in terms of data points labelled with c) with the data points in the voronoi cell for n_i. We denote the set of data points in the voronoi cell for n_i as $v(n_i) = \{d \in D_{train} \mid \forall n_j, j \neq i : |n_j - d|^2 \geq |n_i - d|^2\}$ within the topological map.

$$l(n_i) = \arg\max_c |D(c) \cap v(n_i)|$$

In addition to the neuron labelling strategy, we need to define prediction functions that assign labels to unseen examples. These prediction strategies are inspired by linkage strategies typically used in cluster analysis (see [4,1,13]):

- *Single-linkage:* In this prediction strategy a new data point d_{new} is labelled with category c of the neuron n that minimises the distance to this new example:

$$l(d_{new}) = \arg\min_c (\arg \min_{n \in N(c)} |n - d_{new}|^2)$$

  where $N(c) = \{n \in N \mid l(n) = c\}$ is the set of all neurons labelled with category c according to one of the above mentioned neuron labelling function.

- *Average-linkage:* In this strategy, example d_{new} adopts the label of category c having the minimal average distance to the example:

$$l(d_{new}) = \arg\min_c (\sum_{k=1}^{|N(c)|} \frac{|n_k - d_{new}|^2}{|N(c)|})$$

- *Complete-linkage:* In this prediction strategy a new data point d_{new} is labelled with category c of the neuron n that minimises the maximal distance to this new example:

$$l(d_{new}) = \arg\min_c (\arg \max_{n \in N(c)} |n - d_{new}|^2)$$

3 Online Labelling Strategies for GNG

In order to extend GNG to an online classification algorithm, we extend the basic GNG by a step in which the label of the presented stimulus is assigned to the winner neuron during the learning process. We denote the winner neuron for data point d by $w(d)$. All prediction strategies are local in the sense that they do not consider any neighbouring neurons besides the winner neuron $w(d)$. As the labelling is performed on-the-fly, the label assigned to a neuron can change over time, so that the labelling function is dependent on the number of examples the network has seen and has the following form: $l : N \times T \rightarrow C$. We will simply write $l_t(n_i)$ to denote the label assigned to neuron n_i after having seen t data points.

- *Relabelling method (relabel)*: According to this very simple strategy, the winner neuron $w(d)$ adopts the label of d:

$$l_t(n_i) = l_d, \ \textit{where}\ n_i = w(d)$$

- *Frequency-based method (freq)*: We assume that each neuron stores information about how often a data point of a certain category has been assigned to n_i after t examples have been presented to the network as $freq_t(c, n_i)$. A neuron is labelled by the category which maximises this frequency, i.e.

$$l_t(n_i) = \arg\max_c freq_t(c, n_i)$$

- *Limited-distance method (limit)*: According to this strategy, the winner neuron $n_i = w(d)$ adopts the category label l_d of the data point d if the distance between them is lower than the adaptive distance $\theta_t(n_i)$ of the neuron n_i. In case of a smaller distance, $\theta_t(n_i)$ will be updated with the new distance.

$$l_t(n_i) = \begin{cases} l_d, & \textit{if}\ |n_i - d|^2 \leq \theta_t(n_i) \\ l_{t-1}(n_i), & \textit{else} \end{cases}$$

As these labelling strategies do not guarantee that every neuron in the network is actually labelled, we need to extend the prediction strategy to handle unlabelled neurons. For the presented prediction strategies we simply ignore unlabelled neurons during the prediction state.

4 Experiments and Results

We evaluate the labelling strategies in dependance of the mentioned prediction methods on three classification data sets: an artificial data set generated following a gaussian distribution, the ORL face database[1] and the image segmentation data set of the UCI machine learning database[2]:

[1] Samaria, F.S.: Parameterisation of a stochastic model for human face identification, *In Proc. of the IEEE Workshop on Applications of Computer Vision 1994*, 138-142

[2] Blake, C.L., Merz, C.J.: UCI repository of machine learning databases, 1998

Online labelling for growing neural gas (OGNG)

```
1. Start with two units $i$ and $j$ at random positions in the input space.
2. Present an input vector $x \in R^n$ from the input set or according to input distribution.
3. Find the nearest unit $s_1$ and the second nearest unit $s_2$.
4. **Assign the label of $x$ to $s_1$ according to the present labelling strategy.**
5. Increment the age of all edges emanating from $s_1$.
6. Update the local error variable by adding the squared distance between $w_{s_1}$ and $x$.
7. Move $s_1$ and all its topological neighbours (i.e. all the nodes connected to $s_1$ $\Delta error(s_1) = |w_{s_1} - x|^2$ by an edge) towards $x$ by fractions of $e_b$ and $e_n$ of the distance:
       $\Delta w_{s_1} = e_b(x - w_{s_1})$
       $\Delta w_n = e_n(x - w_n)$
   for all direct neighbours of $s_1$.
8. If $s_1$ and $s_2$ are connected by an edge, set the age of the edge to 0 (refresh). If there is no such edge, create one.
9. Remove edges with their age larger than $a_{max}$. If this results in nodes having no emanating edges, remove them as well.
10. If the number of input vectors presented or generated so far is an integer or multiple of a parameter $\lambda$, insert $a$ new node $r$ as follows:
    Determine unit $q$ with the largest error.
    Among the neighbours of $q$, find node $f$ with the largest error.
    Insert $a$ new node $r$ halfway between $q$ and $f$ as follows:
        $w_r = \frac{w_q + w_f}{2}$
    Create edges between $r$ and $q$, and $r$ and $f$. Remove the edge between $q$ and $f$.
    Decrease the error variable of $q$ and $f$ by multiplying them with a constant $\alpha$. Set the error $r$ with the new error variable of $q$.
11. Decrease all error variables of all nodes $i$ by a factor $\beta$.
12. If the stopping criterion is not met, go back to step (2). (For our experiments, the stopping criterion has been set to be the maximum network size.)
```

Fig. 1. GNG algorithm with extension for online labelling

Artificial data set (ART): The first data set is a two dimensional gaussian mixture distribution with 6 classes located at [0,6], [-2,2], [2,2], [0,-6], [-2,-2], [2,-2]. The data points of each class are gaussian distributed with the standard derivation of 1.
ORL face database (ORL): The second data set is the ORL face database which contains 400 frontal images of humans, performing different gestures. The data set consists of 40 individuals showing 10 gestures each. We downscaled each image from 92 × 112 to 46 × 56 and applied a principal component analysis (PCA) to reduce the number of dimensions from 2576 to 60, corresponding to 86.65% of the total variance.
Image Segmentation data set (SEG): The image segmentation data set consists of 2310 instances from 7 randomly selected outdoor images (brick-face, sky, foliage, cement, window, path, grass). Each instance includes 19 attributes that describe a 3 × 3 region within one of the images.

In order to compare the different labelling strategies to each other, we chose a fixed set of parameters for GNG and used this set in all of our experiments. This set was empirically determined on a trial-and-error basis through preliminary experiments. The GNG parameters are set as follows: insertion parameter $\lambda =$

300; maximum age $a_{max} = 120$; adaptation parameter for winner $e_b = 0.2$; adaptation parameter for neighbourhood $e_n = 0.006$; error variable decrease $\alpha = 0.5$; error variable decrease $\beta = 0.995$

For our experiments we randomly sampled 10 training/test sets out of our data and averaged the accuracy for each labelling and prediction setting. Thereby, we trained each GNG classifier with 4 labelled examples of each category. We validated the accuracy of the labelling methods and prediction strategies on test. Our results are shown in Tables 1 and 2. Both tables show the classification accuracy for various configurations of labelling methods (min-dist, avg-dist, voronoi, relabel, freq, limit) and prediction strategies (single-linkage, average-linkage and complete-linkage) averaged over the three different data sets using 4 training samples per class. We evaluated the accuracy of each labelling method combined with three prediction strategies (rows of the tables). Therefore, we consider the results of 54 experiments overall[3]. The results license the following conclusions:

- **Comparison of offline labelling strategies:** According to Table 1, there is no labelling method which significantly outperforms the others. Comparing the accuracy results averaged over all prediction strategies, the *voronoi method* is the most effective labelling method as it provides the highest accuracy with 77.59%, followed by the *min-dist method* with 76.27% and the *avg-dist method* with 74.28%. Concerning the prediction strategies, the *single-linkage prediction* strategy shows best results averaged over all methods with 81.41%, followed by the *average-linkage prediction* strategy with an accuracy of 77.65%. The *complete-linkage* yielded the worst results with an averaged accuracy of 69.07%.
- **Comparison of online labelling strategies:** According to Table 2, all three online labelling strategies are almost equal in their classification performance. The *limit method* performs slightly better than the other two methods and achieves an accuracy of 78.15%, followed by the *freq method* with an accuracy of 78.09% and the *relabel method* with an accuracy of 77.88%. As for the offline labelling strategies, here it is also the case that the *single-linkage prediction* is the best choice with an accuracy of 83.30%, followed by the *average-linkage prediction* with an accuracy of 80.90% and the *complete-linkage prediction* with an accuracy of 69.88%.
- **Online vs. offline labelling strategies:** Comparing the averaged accuracy of all labelling methods of Table 1 and 2, the results show that there is no significant difference between them in terms of performance. The online labelling methods even provide a slightly higher accuracy.
- **Impact of memory:** Strategies relying on some sort of memory (e.g. storing the frequency of seen labels as in the *freq method*), do not perform significantly better than a simple context-free (or memory-less) method (*relabel method*) performing decisions on the basis of new data points only. This shows that the implementation of a label memory does not enhance the classifiers performance.

[3] The results of all 54 experiments can be found at `http://www.sc.cit-ec.uni-bielefeld.de/people/obeyer`

Table 1. Classification accuracy for the offline labelling strategies (min-dist, avg-dist, voronoi) combined with the prediction strategies (single-linkage, average-linkage, complete-linkage) averaged over the three data sets (ART, ORL, SEG) trained with 4 labelled data points of each category (best prediction strategy and labelling method results are marked).

Prediction strategies	Min-dist method	Avg-dist method	Voronoi method	*Average*
Single-linkage	81.93	78.92	83.39	**81.41**
Average-linkage	77.15	76.35	79.46	77.65
Complete-linkage	69.72	67.56	69.92	69.07
Average	76.27	74.28	**77.59**	

Table 2. Classification accuracy for the online labelling strategies (relabel, freq, limit) combined with the prediction strategies (single-linkage, average-linkage, complete-linkage) averaged over the three data sets (ART, ORL, SEG) trained with 4 labelled data points of each category (best prediction strategy and labelling method results are marked).

Prediction strategies	Relabel method	Freq method	Limit method	*Average*
Single-Linkage	83.25	83.25	83.39	**83.30**
Average-Linkage	80.46	81.12	81.13	80.90
Complete-Linkage	69.92	69.79	69.93	69.88
Average	77.88	78.05	**78.15**	

5 Related Work and Conclusion

In this paper we have presented, analysed and compared different online labelling strategies in order to extend the growing neural gas (GNG) algorithm to an online classification approach. While GNG is essentially an unsupervised algorithm, previous approaches have presented extensions of GNG for classification tasks. Such extensions typically rely on a suitable labelling function that assigns labels to neurons as well as a prediction function that assigns labels to unseen examples. In this line, we have experimentally compared different offline and online labelling strategies inspired by previous research. To our knowledge, there has been no systematic investigation and comparison of different offline strategies so far, a gap we have intended to fill. The question of how GNG can be extended to an online classification algorithm has also not been addressed previously. In most cases, offline strategies have been considered that perform the labelling after the training phase has ended and the network has stabilised to some extent as in the WEBSOM [7,8] and LabelSOM [12] approaches. In both of these approaches, the label assignment is essentially determined by the distance of the labelled training data point to the neurons of the already trained network. Such offline labelling strategies contradict the online nature of GNG, whose interesting properties are that the network grows over time and only neurons, but no explicit examples, need to be stored in the network. In this sense, an important question we have addressed in this paper is whether GNG can be extended to a classification algorithm without affecting its online nature or degrading performance considerably. Our research has shown that this is indeed possible. Online labelling functions where the label of a neuron can change over time and is computed when a new example is assigned to the neuron in question do not perform worse than offline labelling strategies.

Acknowledgement. This project has been funded by the Excellence Initiative of the Deutsche Forschungsgemeinschaft (DFG).

References

1. Anderberg, M.R.: Cluster analysis for applications. Academic Press, New York (1973)
2. Fritzke, B.: A growing neural gas network learns topologies. In: Proceedings of the International Conference on Advances in Neural Information Processing Systems (NIPS) 1995, pp. 625–632 (1995)
3. Hyotyniemi, H.: Text document classification with self-organizing maps. In: Proc. of the Finnish Artificial Intelligence Conf. (STeP) 1996, pp. 64–72 (1996)
4. Johnson, S.C.: Hierarchical clustering schemes. Psychometrika 32(3), 241–254 (1967)
5. Kohonen, T.: Self-organized formation of topologically correct feature maps. Biological Cybernetics 43(1), 59–69 (1982)
6. Kohonen, T.: Learning vector quantization. The Handbook of Brain Theory and Neural Networks, pp. 537–540. MIT Press, Cambridge (1995)
7. Kohonen, T., Kaski, S., Lagus, K.: Self organization of a massive document collection. IEEE Transactions on Neural Networks 11(3), 574–585 (2000)
8. Lagus, K., Kaski, S.: Keyword selection method for characterizing text document maps. In: Proc. of the Int. Conf. on Artificial Neural Networks (ICANN), pp. 371–376 (1999)
9. Lau, K., Yin, H., Hubbard, S.: Kernel self-organising maps for classification. Neurocomputing 69(16-18), 2033–2040 (2006)
10. Lefebvre, G., Garcia, C.: A probabilistic self-organizing map for facial recognition. In: Proc. of the Int. Conf. on Pattern Recognition (ICPR) 2008, pp. 1–4 (2008)
11. Martinetz, T., Schluten, K.: A "neural-gas" network learns topologies. In: In Proc. of the Int. Conf. on Artificial Neural Networks (ICANN) 1991, pp. 397–402 (1991)
12. Rauber, A.: LabelSOM: On the labeling of self-organizing maps. In: Proc. of the Int. Joint Conf. on Neural Networks (IJCNN) 1999, pp. 3527–3532 (1999)
13. Sneath, P.H.A., Sokal, R.R.: Numerical taxonomy: The principles and practice of numerical classification, 1st edn. W H Freeman & Co. (Sd), New York (1973)
14. Steil, J.J., Wersing, H.: Recent trends in online learning for cognitive robotics. In: Proc. of the European Symposium on Artificial Neural Networks (ESANN) 2006, pp. 77–87 (2006)

Multi-layer Topology Preserving Mapping for *K*-Means Clustering

Ying Wu[1], Thomas K. Doyle[1], and Colin Fyfe[2]

[1] Coastal and Marine Research Centre, ERI, University College Cork
Glucksman Marine Facility, Naval Base, Haulbowline, Ireland
{y.wu,t.doyle}@ucc.ie

[2] Applied Computational Intelligence Research Unit,
The University of the West of Scotland, Scotland
colin.fyfe@uws.ac.uk

Abstract. In this paper, we investigate the multi-layer topology preserving mapping for K-means. We present a Multi-layer Topology Preserving Mapping (MTPM) based on the idea of deep architectures. We demonstrate that the MTPM output can be used to discover the number of clusters for K-means and initialize the prototypes of K-means more reasonably. Also, K-means clusters the data based on the discovered underlying structure of the data by the MTPM. The standard wine data set is used to test our algorithm. We finally analyse a real biological data set with no prior clustering information available.

1 Introduction

Clustering is an active area in data mining and machine learning research, which can be considered as one of the most well-known and important unsupervised learning problems. The K-means algorithm [14] is one of the most popular partitional clustering algorithms that attempts to find K clusters which minimize the mean squared quantization error [4]. It is robust and very simple to implement and thus tends to be one of the first algorithms used on a new data set.

The cross entropy method has been well introduced in [5,15]. In [17,16], we have investigated an on-line cross entropy method for unsupervised data exploration. In this paper, we follow a similar idea using a deep architecture in which we present a multi-layer topology preserving mapping for K-means clustering. We demonstrate that by applying the multi-layer topology preserving mapping: (1) It is easy to determine the number of clusters of the data set for K-means. (2) The K-means algorithm clusters the data points based on the underlying structure of the data set discovered by MTPM, by which the accuracy of K-means clustering can be improved. (3) The algorithm in this paper can set the initial values to the prototypes more reasonably.

2 Topology Preserving Manifolds

A topographic mapping captures the underlying structure in the data set, such that points which are mapped close to one another have some common feature

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 84–91, 2011.

while points that are mapped far from one another do not share this feature. The most common topographic mapping is Kohonen's self-organizing map (SOM) [13]. The Generative Topographic Mapping (GTM) [3] is a mixture of experts model which treats the data as having been generated by a set of latent points where the mapping is *non-linear*. Fyfe [8] has derived an alternative topology preserving model, called the *Topographic Products of Experts* (ToPoE), based on a product of experts [10], which is closely related to the GTM.

We follow [3,8] to create a latent space of points $\mathbf{x}_1, \ldots, \mathbf{x}_K$ which lie equidistantly on a line or at the corners of a grid. To allow non-linear modeling, we define a set of M basis functions, $\phi_1(), \ldots, \phi_M()$, with centres μ_j in latent space. Thus we have a matrix Φ where $\phi_{kj} = \phi_j(\mathbf{x}_k)$, each row of which is the response of the basis functions to one latent point, or, alternatively each column of which is the response of one of the basis functions to the set of latent points. Typically, the basis function is a squared exponential. These latent points are then mapped to a set of points $\mathbf{m}_1, \ldots, \mathbf{m}_K$ in data space where $\mathbf{m}_j = (\Phi_j \mathbf{W})^T$, through a set of weights, $\mathbf{W}$. The matrix $\mathbf{W}$ is $M \times D$ and is the sole parameter which we change during training. We have

$$\mathbf{m}_k = \sum_{j=1}^{M} \mathbf{w}_j \phi_j(\mathbf{x}_k) = \sum_{j=1}^{M} \mathbf{w}_j \exp(-\beta||\mu_j - \mathbf{x}_k||^2), \forall k \in \{1, \ldots, K\}. \quad (1)$$

where $\phi_j(), j = 1, \ldots, M$ are the M basis functions, and $\mathbf{w}_j$ is the weight from the j^{th} basis function to the data space. The algorithm is summarized as:

1. Randomly select a data point, $\mathbf{t}$.
2. Find the closest prototype, say $\mathbf{m}_{k*}$, to $\mathbf{t}$.
3. Generate T samples from the Gaussian distribution, $\mathcal{N}(\mathbf{m}_{k*}, \beta_{k*}^2 I)$. Call the samples, $\mathbf{y}_{k*,1}, ..., \mathbf{y}_{k*,T}$. We note that we are using $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K$ to perform two conceptually separate functions, as prototypes or means to which the data will be quantized and as centres of Gaussian distributions from which samples will be drawn.
4. Evaluate the samples using $S(\mathbf{y}) = \exp(-\gamma \parallel \mathbf{y} - \mathbf{t} \parallel^2)$ as the performance function.
5. Sort the samples using $p(1)$ as the worst to $p(T)$ as the best. i.e. we will use this to identify the r elite samples.
6. Update the parameters

$$\mathbf{w} \leftarrow \mathbf{w} + \eta(\frac{1}{r} \sum_{j=T-r}^{T} \mathbf{y}_{k*,p(j)} - \mathbf{m}_{k*})\phi(\mathbf{x}_{k*}) \quad (2)$$

$$\beta_{k*}^2 = \beta_{k*}^2 + \eta_0(\frac{1}{r} \sum_{j=T-r}^{T} (\mathbf{y}_{k*,p(j)} - \mathbf{m}_{k*})(\mathbf{y}_{k*,p(j)} - \mathbf{m}_{k*})^T) \quad (3)$$

   where η, η_0 are the learning rates with typically $\eta = 10\eta_0$.
7. Update the prototypes' positions using (1).

3 Multi-layer Topology Preserving Mapping for *K*-Means Clustering

Deep architectures are compositions of many layers of adaptive non-linear components, which allow representations of data in a more compact form than shallow architectures. Bengio and LeCun [2] have demonstrated that deep architectures are often more efficient for solving complicated problems in terms of number of computational components and parameters. A greedy, layer-wise unsupervised learning algorithm [1,11,12] has been introduced to provide an initial configuration of the parameters with which a gradient-based supervised (backpropagation) learning algorithm is initialized, which results in a very much more efficient learning machine. The idea behind the deep architecture is that simpler models are learned sequentially and each model in the sequence receives a different representation of the data. In this section, we extend the topology preserving mapping with cross entropy to a multi-layer topology preserving mapping (MTPM), where we follow the similar idea and demonstrate the deep architecture in unsupervised clustering. We create a multi-layer topology preserving mapping model, where we perform the same topology preserving mapping algorithm in each layer to capture more accurately the more abstract underlying data structure. The K-means algorithm is performed in each layer using the projection of the latent points in that layer. The prototypes of K-means from 2^{nd} layer are initialized by the clustered output of the previous layer.

Specifically, we create a q-dimensional latent space in each layer with a regular array of points, $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_K)$ that have the structure of lying equidistantly on a line or on a grid. These latent points are nonlinearly mapped to points, $(\mathbf{m}_1, \ldots, \mathbf{m}_K)$ in the input space through a set of basis function, which forms a set of reference vectors, $\mathbf{m}_i = \mathbf{W}\phi(\mathbf{x}_i)$. Each of the reference vector then forms the centre of the Gaussian distribution in the input space and we can represent the distribution of the data points in input space in terms of a smaller q-dimensional nonlinear manifold. Thus we have a higher-level representation of the data points in the input space by the projection of the data points in the latent space. The latent space representation of each data point is

$$\mathbf{t}_n^{latent} = \sum_{i=1}^{K} r_{ni}\mathbf{x}_i \tag{4}$$

$$r_{ni} = \frac{\exp(-\gamma d_{ni}^2)}{\sum_k \exp(-\gamma d_{nk}^2)} \tag{5}$$

where r_{ni} is the responsibility of the i^{th} latent point for the n^{th} data point and $d_{pq} = ||\mathbf{t}_p - \mathbf{m}_q||$, is the Euclidean distance between the p^{th} data point and the projection of the q^{th} latent point. The projection of the data points in the latent space of this layer then becomes the data points in the input space of the next layer. Therefore, the topology preserving mapping in each layer performs a non-linear transformation on its input vectors and produces as output the vectors that will be used as input for the topology preserving mapping in the next layer and the projection of the data points in higher layers may represent the underlying structure of the data set more clearly. We develop a new way to visualize the experimental results, where we can identify the possible number of clusters, K, for K-means: we calculate the sum of the responsibility vectors over the data points, $R = \sum_{n=1}^{N} \mathbf{r}_n$, and then form a responsibility grid using R for each latent point. Since the latent points are fixed in (4), the data points belonging to the same cluster will have similar responsibility vectors, and thus the area where one cluster is located in the responsibility grid will become 'hot' with higher density of data points.

In l^{th} layer, we perform the K-means algorithm to cluster the output and assign the labels, $L_c^l, c = 1...K$, to the data points in the latent space, $\mathbf{t}^{latent}$, where the data points are thus partitioned into K clusters. When the labeled $\mathbf{t}^{latent}$ is transferred to the $(l+1)^{th}$ layer, the prototypes of K-means are initialized to the mean values of the data points with the same labels as (6),

$$p_c^{l+1} = \frac{1}{N_c^l} \sum \mathbf{t}_{n'}^{latent}, n' \in L_c^l \tag{6}$$

where N_c^l is the number of data points in cluster L_c^l in layer l. Since the projection of the data points in a higher layer represents the underlying structure of the data more clearly and abstractly, the K-means is performed based on the discovered underlying data structure and the prototypes of K-means are initialized more reasonably and accurately. Finally, we use the K-means result in the topmost layer as the final clustering result.

To demonstrate our multi-layer topology preserving mapping for K-means, we create a deep architecture model with four layers, in each layer of which we use a 2-dimensional grid of latent points: we use a 21×21 grid of latent points being passed through a 5×5 set of basis vectors. We begin by illustrating the algorithm on the well-known wine data set from the UCI Repository of machine learning databases. It has 178 samples, 13 features and 3 classes. Because some of the features are scaled up to 1500 and others lie between 0 and 1, we preprocessed the data by normalizing all features between -1 and 1.

A plot of the responsibility grid and the projection of the data points in the latent space in each layer is shown in Figure 1. We can see that although the data points have been mapped into the latent space accurately in the first layer, there are only scattered hot areas in the responsibility grid. We can also see that the higher level the layer is in, the hotter the areas corresponding to different clusters are and in the fourth layer, we can clearly see that three clusters have been identified. We evaluate the performance of the K-means in each layer by the quantization error [4] as listed in Table 1. We can see that the quantization error is decreased in the higher layer, which corresponds to what is shown in Figure 1 that the data points belonging to the same cluster in higher layer become much closer than those in the first layer. We compare our algorithm with PCA+K-means. In this case, the quantization error with PCA+K-means is 144.9405. We find that the MTPM+K-means classifies the data points more accurately. In this case, 65.10% data points are classified correctly by PCA+ K-means and 88.20% data points are classified correctly by MTPM+ K-means in the 4^{th} layer.

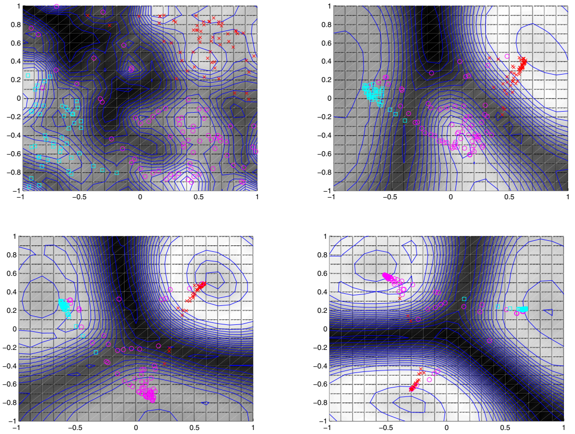


Fig. 1. Plot of the responsibility grid and the projection of the data points in the latent space in the first four layers with correct labels. Each contour line represents a constant value of R. The red cross, cyan square and magenta cycle correspond to the data points for different clusters. Top left: the 1^{st} layer. Top right: the 2^{nd} layer. Bottom left: the 3^{rd} layer. Bottom right: the 4^{th} layer.

Therefore, we consider the MTPM model can capture the underlying structure of the data set more clearly and accurately in the higher layer and the performance of K-means is improved in the higher layer.

Table 1. The quantization errors of K-means in eight layers

Layer	Quantization Error
1^{st}	30.6165
2^{nd}	6.0467
3^{rd}	4.7441
4^{st}	3.2859
5^{st}	1.8430
6^{st}	1.2887
7^{st}	0.8292
8^{st}	0.4887

4 Biological Data Set

In this section, we apply our algorithm to a real biological data set. The data used describes the diving behaviour of a leatherback sea turtle (*Dermochelys coriacea*) that was satellite tagged in Irish waters. Specifically, a Satellite Relay Data Logger (SRDL) was attached to a male leatherback sea turtle on 29th June 2006, in Dingle, Ireland ($52.24°N$, $10.30°W$) [6]. In total, 807 time/dive profiles were extracted from this real turtle data set for analysis. Each time/depth profile, as shown in the left of Figure 2 is made up of 5 time/depth pairs (critical points) which best represent the actual dive (i.e. as the dive profile is made up of many hundreds or thousands of time/depth pairs [one collected every 4 seconds] an algorithm onboard the SRDL summarises the profile into the 5 most significant pairs [7,9]). This process of reduction retains essential information (dive shape) but dramatically reduces the amount of information that is to be sent via a satellite system (the maximum number of bits allowed in a single Argos message is 256). In addition to the critical points, total dive duration is also transmitted. Apart from these dive profiles, no other prior knowledge is known about the data set.

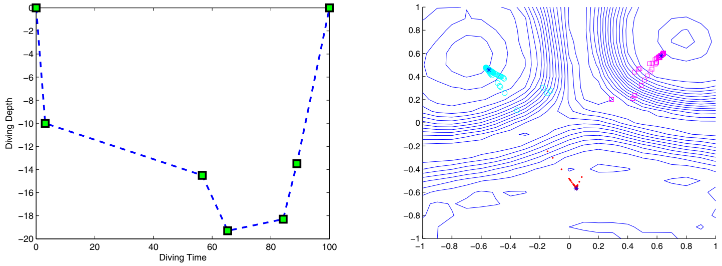


Fig. 2. Left: An example of a dive profile with 5 'critical points'. Right: Clustered dive profiles in in the responsibility grid.

From the data set we have original raw data values: dive duration, depth of critical point (D1,. . . , D5) and time of critical point (T1,. . . , T5) as shown in the left of Figure 2. From this raw data we generated some derived data values or descriptors to help classify the data set. These descriptors are: total square (area) of the dive profile, time interval proportion (the proportion of time intervals between two critical points to the total duration), square of interval shape, interval square proportion (the proportion of the square in a time interval) and depth difference (change in depth during one time interval).

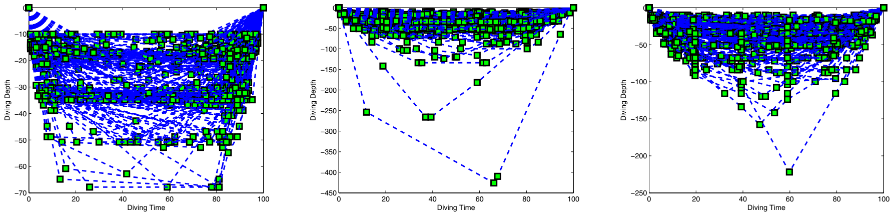


Fig. 3. Plot of the dive profiles. The green 'square' point represent a critical diving point of one dive profile. Left: the dive profiles in 1^{st} cluster. Middle: the dive profiles in 2^{nd} cluster. Right: the dive profiles in 3^{rd} cluster.

Figure 2, right, shows that the algorithm consistently (29 times out 30) identified three possible clusters (on one occasion 4 clusters were identified). When the three clusters are graphed as 2D plots in Figure 3 it can be seen that the dive profiles within each cluster have common features that differentiate them from the dive profiles within the other clusters. For example, the individual dive profiles plotted in the left of Figure 3 all have what is described in biology as a 'square profile' (i.e. dive shape is typically square, so initial large change in depth followed by little change over next 2 or 3 critical points). Dive profiles from the right of Figure 3 have more of a 'v' shape profile, so similar rate of change in depth between all critical points. Further investigation of this type of clustering analysis may provide novel insights on animal diving behaviour, particularly if such clusters are further refined using other descriptors of dive profiles such as: time of day, % bottom time, day length, location (latitude and longitude) and sea temperature.

5 Conclusion

In this paper, we have investigated multi-layer topology preserving mapping for K-means clustering, as an illustration of considering deep architectures for unsupervised clustering. We demonstrated that with multi-layer topology preserving mapping, the number of clusters in the data can be identified more easily and accurately. The data points in higher layer are gathered with higher density in the areas where clusters are located and the performance of K-means algorithm can be improved, where the quantization error is decreased and it tends to classify the data with higher accuracy. Also, since the projection of the data points in a higher layer represents the underlying structure of the data more clearly and abstractly, we consider it is more reasonable and accurate to initialize the K-means prototypes based on the underlying data structure discovered in the lower layers. Two real data sets have been analysed by applying our algorithm. We have used the wine data set, a standard data set with prior knowledge of the number of classes, to demonstrate the performance of K-means can be improved. Then we have analysed a real biological data set, a data set without prior knowledge of clusters. Useful clustering information has been extracted, which leads to further biological investigation.

References

1. Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems, vol. 19, pp. 153–160. MIT Press, Cambridge (2007)
2. Bengio, Y., LeCun, Y.: Large-Scale Kernel Machines. In: Scaling Learning Algorithms towards AI. MIT Press, Cambridge (2007)
3. Bishop, C.M., Svensen, M., Williams, C.K.I.: GTM: The generative topographic mapping. Neural Computation 10, 215–234 (1998)

4. Bottou, L., Bengio, Y.: Convergence properties of the k-means algorithms. In: Advances in Neural Information Processing Systems, vol. 7, pp. 585–592. MIT Press, Cambridge (1995)
5. de Boer, P.-T., Kroese, D.P., Mannor, S., Rubenstein, R.Y.: A tutorial on the cross-entropy method. Annals of Operations Research 134(1), 19–67 (2004)
6. Doyle, T.K., Houghton, J.D.R., O'Suilleabhain, P.F., Hobson, V.J., Marnell, F., Davenport, J., Hays, G.C.: Leatherback turtles satellite tagged in european waters. Endangered Species Research 4, 23–31 (2008)
7. Fedak, M., Lovell, P., McConnell, B., Hunter, C.: Overcoming the constraints of long range radio telemetry from animals: Getting more useful data from smaller packages. Integrative and Comparative Biology 42(1), 3–10 (2002)
8. Fyfe, C.: Two topographic maps for data visualization. Data Mining and Kownledge Discovery 14, 207–224 (2007)
9. Hays, G.C., Houghton, J.D.R., Isaacs, C., King, R.S., Lloyd, C., Lovell, P.: First records of oceanic dive profiles for leatherback turtles, dermochelys coriacea, indicate behavioural plasticity associated with long-distance migration. Animal Behaviour 67(4), 733–743 (2004)
10. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Technical Report 2000-004, Gatsby Computational Neuroscience Unit. University College, London (2000)
11. Hinton, G.E., Osindero, S., Teh, Y.: A fast learning algorithm for deep belief nets. Neural Computation 16, 1527–1554 (2006)
12. Hinton, G.E., Salakhutdinov, R.R.: Reducing the demensionality of data with neural networks. Science 313, 504–507 (2006)
13. Kohonen, T.: Self-organising maps. Springer, Heidelberg (1995)
14. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Le Cam, L.M., Neyman, J. (eds.) Proc. of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)
15. Rubinstein, R.Y.: Optimization of computer simulation models with rare events. European Journal of Operations Reasearch 99, 89–112 (1997)
16. Wu, Y., Fyfe, C.: The on-line cross entropy method for unsupervised data exploration. WSEAS Transactions on Mathematics 6(12), 865–877 (2007)
17. Wu, Y., Fyfe, C.: Topology preserving mappings using cross entropy adaptation. In: International Conference on Artificial Intelligence, Knowledge Engineering and Data Bases (2008)

On the Extraction and Classification of Hand Outlines

Luke Davis[1], Barry-John Theobald[1], Andoni Toms[2], and Anthony Bagnall[1]

[1] School of Computing Sciences, University of East Anglia, Norwich, UK
{luke.davis,b.theobald,anthony.bagnall}@uea.ac.uk
http://www.uea.ac.uk/cmp

[2] Radiology Academy, Norfolk and Norwich University Hospital, Norwich, UK
andoni.toms@nnuh.nhs.uk
http://www.nnuh.nhs.uk/nra

Abstract. We examine alternative ways of finding the outline of a hand from an X-ray as the first stage in segmenting the image into constituent bones. We assess a variety of algorithms including contouring, which has not previously been used in this context. We introduce a novel ensemble algorithm for combining outlines based on dynamic time warping (DTW). Our goal is to minimise the human intervention required, hence we investigate alternative ways of training a classifier to determine whether an outline is in fact correct or not. We evaluate outlining and classification on a set of 1370 images. We conclude that ensembling improves performance of all outlining algorithms, that the contouring algorithm used with the ensemble performs the best of those assessed, and that the most effective classifier of hand outlines assessed is a random forest applied to outlines transformed into principal components.

Keywords: Hand segmentation, outline classification, outline ensemble.

1 Introduction

This research is part of a wider project to build predictive models of bone age using hand radiograph images. Bone age assessment typically involves estimating the expected age of a patient from a radiograph by quantifying the development of the bones of the non dominant hand. It is used to check that a child's bones are developing at an acceptable rate and to monitor whether certain treatments are affecting a patient's skeletal development. Currently this task is performed manually by scoring each bone using a system such as Tanner and Whitehouse (TW) [10]. It is a time consuming and often inaccurate procedure. We are attempting to develop a fully automated system that will quickly produce an accurate estimate of bone age based on predictive models constructed from a wide range of images. Broadly speaking, bone age is estimated by the shape and position of the bones in the hand. Hence our approach involves segmenting the image to find the location of individual bones, then extracting features to build a regression model. The unsupervised extraction of bones is not trivial, and we

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 92–99, 2011.

believe that by first extracting the outline of the hand we will get a better segmentation of the bones as the hand outline gives clearly defined landmarks upon which to begin the extraction. This paper addresses two problems associated with the task of extracting the outline of a hand from an image: firstly, we assess how to extract the outline; and secondly we evaluate how to determine whether a given outline is in fact a correct outline of a hand. We assess four candidates for extracting the hand outline: Otsu thresholding [9]; Canny edge detection [2]; active appearance models (AAM) [3]; and contouring [5]. These are briefly described in Section 2. Despite the fact that the first three have previously been used in this context [1,6,11] our experience is that the variability in intensity across images, low contrast between background, flesh and bone, and variability in hand size and shape mean that the extraction of the outline is non-trivial and that none of the algorithms assessed are consistent enough for our requirements. Hence we define an ensemble method that combines the outlines formed from a range of transformed images (see Section 2), which we find creates much better outlines (as assessed in Section 4). However, none of the algorithms we evaluate work perfectly. Figure 1 shows two examples of incorrect outlines. In these examples the algorithm has found the internal outline of the metacarpals or phalanges. We have observed several other types of error, such as over extended regions or cropping of individual fingers.

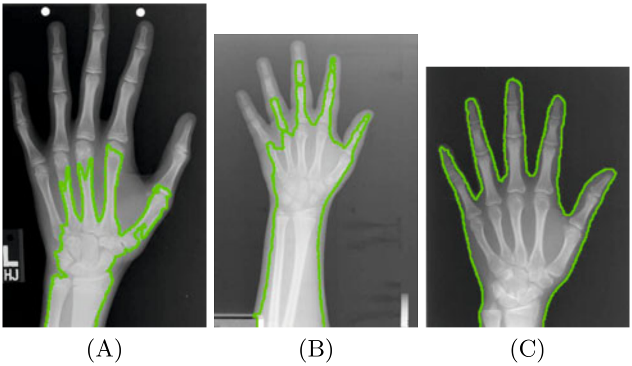


Fig. 1. (A) and (B) Two examples of incorrectly located hand outlines, and (C) a hand outline correctly segmented

Clearly, an incorrectly segmented outline will compromise any subsequent steps of bone segmentation and age modelling. Since our aim is to minimise the human intervention needed to progress from image to age estimate, we require an automated means of classifying whether an outline is in fact correct. We constructed a training set of 1000 images and a test set of 370 images. Each image was automatically segmented (Section 2) and manually labelled as correct or incorrect depending on the quality of the segmentation. Features were then extracted from the training images and a range of classification algorithms (Section 3) were evaluated using the testing set. The results are presented and analysed in Section 4. In Section 5 we conclude from our findings and describe how this prototype system could be improved.

The contributions of this paper can be summarised as follows: firstly, we propose using contouring to find hand outlines in radiographs and compare with other methods used in this problem domain. To our knowledge, contouring has not been used in this context before and our experiments indicate it has great potential; secondly, we propose a new way of combining image outlines through an ensemble technique with diversity achieved through rescaling and a novel voting scheme that uses dynamic time warping; thirdly, we describe a 'time series' classification problem that will be embedded in an automated process to detect whether an outline is in fact correct or not.

2 Outlining a Hand

There has been extensive research on the segmentation of hand radiographs [1,12,11,6]. The majority of this work concentrates on the direct segmentation of the bones and uses the problem of finding the outline merely as a motivational example. However, we consider the seemingly easier problem of finding the hand outline as the most sensible first step; once we have obtained an outline which we are confident is correct, the position of the bones is highly constrained and thus much easier to detect. We have applied four commonly used algorithms for outlining.

Otsu Thresholding [9] segments an image into foreground and background regions by forming a histogram of the relative frequency of intensities across the image and finding the partition that minimises the within group variance. This approach has been applied to hand radiograph segmentation, where the hand outline is extracted from the thresholded image using a simple tracing algorithm [1,12].

Canny Edge Detector [2] is a multistage algorithm that combines differential filtering, non-maximal filtering and thresholding with hysteresis. It has been used previously in the context of hand radiograph segmentation [6,8]. To summarise, the Canny algorithm:

1. smoothes the image with a Gaussian filter;
2. estimates the gradient magnitude and direction at each pixel and quantises the gradient directions to be one $\{0, 45, 90, 135\}$ degrees;
3. performs non-maximal suppression by switching off candidate pixels that are not locally maximum in the direction of the gradient;
4. identifies definite edge pixels as those with a gradient magnitude above a global high value threshold and switching off pixels that have a gradient magnitude below a global low threshold; and
5. checks the pixels with gradient magnitude between the high and low thresholds to determine if there is a path that connects them to a definite edge pixel. Those that are connected to a definite edge form the edge, otherwise they do not.

The Canny edge detector identifies all edges in an image, and we assume that the longest edge represents the hand outline.

Active Appearance Model (AAM) [3] is an extremely effective and popular algorithm for the semi-supervised segmentation of images. AAMs are a parameterised, statistical representation of the shape and appearance variation in a set of training images that have been manually labelled with landmark points. The AAM algorithm can be summarised as follows:

1. manually identify a set of key landmark points on a set of training images;
2. for the shape model, extract the landmark points then normalise for translation, rotation and scale;
3. transform with a principal component analysis and use to form the distribution model for the data;
4. for the appearance model, shape normalise the images by warping from the manually labelled key points onto the mean shape; and
5. transform with a principal component analysis and use to form the appearance model for the data;

Once the model has been built, a further iterative algorithm is used to fit the outline to new instances. We have used the Inverse Compositional AAM [7].

Contouring algorithms treat the image as a surface and attempt to trace outlines as areas of common intensity. In summary:

1. pre-determine the number of contour levels;
2. for each contour level, find the pixels that lie on this level (i.e. have intensity within the range defined by this level); and
3. for each pixel on this level, connect it to its neighbour if it also is on this level and continue until either none of the neighbours belong to this contour or the trace has returned to the starting point.

The output from the contouring algorithm is a set of contours, and we take the largest contour to represent the outline of the hand. Contour algorithms typically are used in weather analysis [5] and to the best of our knowledge have not be used in this context.

2.1 Ensemble Techniques

The main problems with forming the outline of a hand are that image intensities vary greatly from image to image and also within an image. This is because radiographs vary from machine to machine, the bulb deteriorates over time altering the intensity distribution and the energy emitted from the bulb is non-uniform across the bulb. To counteract this, we rescale an image $\mathbf{I}$ using a power transform $\mathbf{I}^{\gamma} + c$, where c is a constant that rescales the intensities back in the range 0–255. The optimal value for γ is image dependent, hence our approach is to perform the transform with a range of values and use each power transformed image as an input to our outlining algorithm. We then ensemble this set of images and choose a single best outline. Unlike traditional classification ensembles, we cannot simply vote on an outline. Instead we measure the quality of each outline against a set of idealised outlines from [4]. We do this by first mapping the outline onto a 1-D series by computing the Euclidean distance of each pixel along the contour to the first pixel. We then measure the Dynamic Time Warping distance between every candidate outline to every idealised image. The outline with the minimum median warping distance to the idealised images is chosen as the output of the ensemble.

3 Classification of an Outline

None of the approaches described in Section 2 is 100% robust against the sources of variation described previously. Our goal is to minimise the requirement for human intervention in bone age assessment, thus we need a classifier to identify whether the output from the ensemble is a true representation of the hand outline. To train a classifier we first produced 1000 hand outlines using a mixture of the methods described in Section 2. Three human subjects manually labelled the training data as correct or incorrect. Since our priority at this stage is to make sure we do not progress with an incorrectly labelled image, an outline is labelled as correct only if all three human subjects classify it as correct.

The outlines were then mapped onto a one dimensional series based on the Euclidean distance of each pixel along the contour to the first point, and these series were smoothed using a median filter and z-normalised (across the series) to remove the scaling effect of age variation. Clearly, the length of outlines will vary. To simplify the classification the outlines were resampled to ensure each was the same length as the shortest series (2709 attributes). The number of attributes and the obvious dependencies between them make it appropriate to transform the data prior to classification. We have experimented with Fourier transforms (FFT) and principal component transforms (PCA) (details in Section 4). We conducted our classification experiments using the WEKA implementation of Random Forest, C4.5, Naive Bayes, SVM and kNN (with k set through cross validation).

4 Results

There are four stages to our experimentation; firstly, we evaluate classifiers on our training set of outlines and choose a subset of classifiers to use in testing; secondly, we apply our outlining algorithms to 370 test images; thirdly, we assess the outline outputs with the classifiers; and finally, we manually assess the outlines and comment on the suitability of the classifiers.

The training set has 638 positive cases and 362 negative cases. The raw (normalised) data has 2709 attributes. For the Fourier transform data set we retain the first 500 Fourier terms. We tried two forms of PCA: the first, PCA1, found the components on the whole data set. The second performed the transform on the positive cases only, then used the components to define features for both the positive and negative cases. For both PCA methods we retained the components that explain 95% of the variation (10 and 14 components respectively).

For PCA2 we also formed a data set of all components. Thus we evaluate each classifier with five training data sets: normalised, PCA1 (95%), PCA2 (95%), PCA2 (100%) and FFT (500).

We performed a ten fold cross validation for each classifier using the default WEKA parameters for all classifiers except for *k*nn, where we selected *k* for each fold separately through a further cross validation on each training subset. The mean classification accuracy and standard deviation between folds are shown in Table 1. We observe that Random Forest achieves the highest overall accuracy of 93.77% using all components of PCA2. A classification accuracy of over 90% is sufficient at this point in the development cycle, hence we continue to use a Random Forest classifier trained on PCA2 (100%).

Table 1. Ten fold cross validation accuracy (with standard deviation) for five classifiers (columns) on five separate data sets (rows)

	Naive Bayes	*k*-nn	C4.5	SVM	Random Forest
Normalised	82.97%(2.9)	86.19%(3.0)	85.24%(3.9)	88.88%(2.9)	88.20%(3.4)
PCA1 (95%)	82.43%(0.4)	86.79%(4.6)	84.89%(0.4)	86.32%5.2)	87.89%(3.3)
PCA2 (95%)	84.02%(3.2)	85.56%(3.1)	84.87%(3.1)	84.51%(3.3)	87.10%(3.1)
PCA2 (100%)	84.79%(2.4)	80.42%(2.8)	85.85%(3.6)	86.79%(2.9)	93.77%(2.1)
FFT (500)	80.33%(3.6)	80.32%(3.4)	76.23%(3.8)	74.86%(3.2)	79.42%(4.0)

The second stage of the experimentation involves forming hand outlines on 370 separate testing images. We used the four methods described in Section 2, then run ensembles of the Canny, Otsu and contour algorithms on rescaled images. The AAM is trained on a manually labelled set of 30 images. We then train a Random Forest classifier on the full training set and use the resulting classifier to classify the outlines as hands or not. Table 2 shows the percentage of each outlining algorithm that the Random Forest classifier determines are actual hand outlines.

Table 2. Percentage of outlines the Random Forest classifier classified as a correct outline of a hand

AAM	74.05%		
Contour (ensemble)	67.57%	Contour	24.05%
Otsu (ensemble)	47.84%	Otsu	11.82%
Canny (ensemble)	0%	Canny	0%

The results in Table 2 suggest that the AAM technique is the best performing outlining scheme. However, investigation of the output indicates that whilst the AAM algorithm more frequently finds a hand shape, it is frequently not the correct outline. Figure 2 gives two examples of this phenomena. This is caused by the fitting procedure used by AAM, where the fit is constrained to always be a valid example in the sense of the training data.

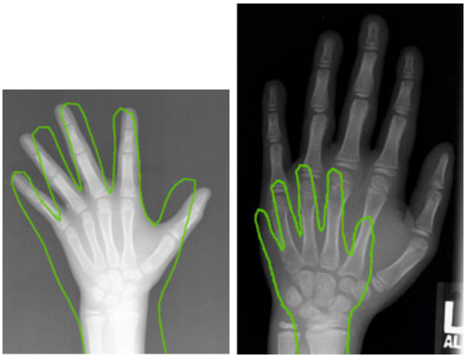


Fig. 2. Two examples of AAM finding an incorrect outline

This error is caused by the model becoming stuck in a local optimum. A manual inspection of the AAM outlines indicates that in fact the AAM outlined only 187 test images correctly (50.54%) and the classifier made 121 false positive classifications. Since we are primarily concerned with minimising false positives, this presents a serious problem. We could increase the training set size for AAM and no doubt improve performance, but there is a strong likelihood that errors of this nature will still occur. Alternatively we could alter our classification scheme to use a measure derived from the image rather than the outline. However, this over complicates our design and given the contour ensemble performance we deem it unnecessary at this stage; a visual inspection of the contour (ensemble) outlines reveals that it in fact correctly found 320 outlines (86.46%). The classifier is over cautious in labelling these, but it makes very few false positive classifications (16). Experimentation with other classifiers revealed a similar pattern. Hence our primary conclusion from these experiments is that the contour ensemble is the most appropriate outlining algorithm for hand images. Our ensemble algorithm improved the performance of both the contour and the Otsu algorithms, thus demonstrating the utility of the approach. The random forest classified all of the Canny outlines as incorrect either with or without the ensemble. A visual inspection revealed this was an overly pessimistic scoring but that nevertheless the Canny performed poorly. We believe this is caused by two factors; firstly, Canny is an edge detector rather than an outline detector. Whilst a human may classify the Canny output as good since it broadly looks correct, it does not generally form a continuous outline. Secondly, whilst the intensity difference between hand and the background is obvious to the human eye, the actual intensity differentials at the boundaries are not particularly great.

5 Conclusions and Future Work

This paper describes a novel ensemble algorithm for outlining radiographs and a classification scheme to automatically detect whether an outline is correct. We have successfully applied this ensemble algorithm to a contouring outliner that can extract correct outlines from over 80% of images. We conclude that AAM is the only other contender in terms of accuracy, but is not suitable for this project because the types of mistake it makes are hard to detect automatically.

Whilst the Random Forest classifier performs adequately, we believe we could construct a better classifier. We will experiment with alternative transformations and classification schemes to try to improve performance. The contour ensemble algorithm performs well, but we believe we could also improve performance by setting contour levels dynamically.

In the wider context of this project, the next task is to extract individual bones from an outlined hand image. We will concentrate on extracting the phalanges then attempting to recreate the Tanner-Whitehouse classification scheme.

References

1. Bielecki, A., Korkosz, M., Zielinski, B.: Hand radiographs preprocessing, image representation in the finger regions and joint space width measurements for image interpretation. Pattern Recognition 41(12), 3786–3798 (2008)
2. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(6), 679–698 (1986)
3. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 681–685 (2001)
4. Gilsanz, V., Ratib, O.: Hand bone age: a digital atlas of skeletal maturity. Springer, Heidelberg (2005)
5. Husar, R.B., Gillani, N.V., Husar, J.D., Paley, C.C., Turcu, P.N.: Long-range transport of pollutants observed through visibility contour maps, weather maps and trajectory analysis. In: Preprint Volume: Third Symposium on Turbulence, Diffusion and Air Pollution, pp. 344–347. American Meteorological Society, Reno (1976)
6. Lehmann, T.M., Beier, D., Thies, C., Seidl, T.: Segmentation of medical images combining local, regional, global, and hierarchical distances into a bottom-up region merging scheme. In: Proc. SPIE, vol. 5747, pp. 546–555. Citeseer (2005)
7. Matthews, I., Baker, S.: Active appearance models revisited. International Journal of Computer Vision 60(2), 135–164 (2004)
8. Muñoz-Moreno, E., Cárdenes, R., De Luis-García, R., Martín-Fernández, M.Á., Alberola-López, C.: Automatic detection of landmarks for image registration applied to bone age assessment. In: Proceedings of the 5th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, pp. 117–122. World Scientific and Engineering Academy and Society (WSEAS) (2005)
9. Otsu, N.: A threshold selection method from gray-level histograms. Automatica 11, 285–296 (1975)
10. Tanner, J.M., Whitehouse, R.H., Cameron, N., Marshall, W.A., Healy, M.J.R., Goldstein, N.H.: Assessment of Skeletal Maturity and Prediction of Adult Height (TW3 Method). WB Saunders (2001)
11. Thodberg, H.H.: Hands-on experience with active appearance models. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, vol. 4684, pp. 495–506 (2002)
12. Zielinski, B.: Hand Radiograph Analysis and Joint Space Location Improvement for Image Interpretation. Schedae Informaticae 17(1), 45–61 (2009)

Clustering-Based Leaders' Selection in Multi-Objective Particle Swarm Optimisation

Noura Al Moubayed, Andrei Petrovski, and John McCall

Robert Gordon University, Aberdeen,
St. Andrew Street, AB25 1HG, Aberdeen, UK
{noura.al-moubayed,a.pertovski,j.mccall}@rgu.ac.uk

Abstract. Clustering-based Leaders' Selection (CLS) is a novel approach for leaders selection in multi-objective particle swarm optimisation. Both objective and solution spaces are clustered. An indirect mapping between clusters in both spaces is defined to recognize regions with potentially better solutions. A leaders archive is built which contains representative particles of selected clusters in the objective and solution spaces. The results of applying CLS integrated with OMOPSO on seven standard multi-objective problems, show that clustering based leaders selection OMOPSO (OMOPSO/C) is highly competitive compared to the original algorithm.

Keywords: Evolutionary Computation, Multi-Objective Particle Swarm Optimisation, Leaders' Selection, Density Based Spatial Clustering, Principal Component Analysis, Domination, OMOPSO.

1 Introduction

Multi-objective optimisation (MOO) is the process of finding feasible solutions to a multi-objective problem (MOP) and evolving these solutions gradually to enhance their quality. MOP is an optimisation problem with more than one conflicting objectives, solving such problems usually results in trade-off solutions among the different objectives [1]. Particle Swarm Optimisation (PSO) (and multi-objective PSO (MOPSO)) [1] is an efficient and stable algorithm which demonstrated applications in various domains [2,3].

The majority of MOPSOs use a leaders archive of a fixed size to store the trade-off solutions found so far through the optimisation process. As the selected leaders highly influence the optimisation process, maintaining the archive and selecting the leaders are main challenges for MOPSO. To update the archive a domination concept is mainly used, where only non-dominated solutions are stored. MOPSO aims at minimizing the distance between the solutions in the archive and the actual Pareto Front (PF) while maximizing the diversity of these solutions in the objective space. To tackle this problem some techniques are proposed, e.g., adaptive grid [4] , niche count [5], ϵ-dominance [6], nearest neighbour [7] and clustering [8]. These techniques maintain the quantity and the diversity of the solutions in the objective space without taking into account the

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 99–106, 2011.

diversity of these solutions in the solution space, which might result in discarding potentially important regions in the solution space.

The behavior of the particles is defined by the leaders, which results in the swarm gradually converging into groups. This justifies the clustering of the swarm to obtain the leaders. Clustering is proposed in [8] and applied only on the non-dominated solutions in the objective space where the solutions closest to the cluster centroids are added to the archive. The authors in [9] proposed a similar clustering method for genetic algorithms.

CLS proposed in this paper goes further by clustering the particles in the solution space as well as in the objective space aiming at maintaining the diversity and covering the potential regions in both objective and solution spaces, and then defining a corresponding relation between clusters in both spaces. The proposed leaders archive considers the dominated as well as the non-dominated solutions.

2 Multi-Objective Particle Swarm Optimisation

The challenge of MOO is that an improvement in one objective often happens at the expense of deteriorating the performance with respect to other objectives. The goal therefore is to find the entire set of trade-off solutions that satisfy the conflicting objectives. The objectives of MOP can be represented as a vector $F(X) = \{f_1(X), f_2(X), \ldots, f_m(X)\}$ where $X \in \Omega$, $\Omega \subset R^n$ and m is the number of objectives. When minimizing $F(X)$, a domination relationship is defined between these solutions as follows: let $X, Y \in \Omega$, $X \succ Y$ if and only if $f_i(X) \leq f_i(Y)$ for all $i = \{1, 2, ..., m\}$, and there is at least one j for which $f_j(X) < f_j(Y)$. X is a Pareto optimal solution if there is no other solution $Y \in \Omega$ such that $Y \succ X$. The image of the Pareto optimal set in the objective space $F(X^*)$ is called the Pareto Front (PF)[1].

In MOPSO, each particle represents a potential solution and exchanges positional information with the global and local leader(s) and its own personal memory. These information are used to move the particle in the search space [1]. Every particle is characterised by its position and velocity. The position is the point in the solution space, whereas the velocity represents the positional change. Each particle uses the position of the selected leader and its personal movement trajectory to update its velocity and position. To solve MOP using MOPSO, four distinct approaches are mainly used: Aggregation approach, where the objectives are merged into a single objective using an aggregation function (e.g., SDMOPSO [10]). Lexicographic ordering approach, in which each objective is independently optimised starting from the objective with the highest priority. Sub-swarms approach divides the swarm into several sub-swarms, each solves the MOP using only one objective, at a certain stage the sub-swarms exchange particles in order to produce trade-off solution (e.g., VEPSO [11]). In Pareto-based approach, the dominance concept is used in order to obtain the set of non-dominated particles (i.e. archive), the swarm leader is then selected from this set (e.g., OMOPSO [12]).

3 The CLS Approach

Clustering based leaders selection uses Density Based Spatial Clustering (DBSC) [13] to assign each individual into one solution space cluster and one objective space cluster where the individual is represented in the solution space by its decision variable vector and in the objective space by its objective values. Should two individuals fall in the same cluster in the solution space, it does not necessarily mean they belong to the same cluster in the objective space and vice versa. DBSC is chosen for its ability to discover a dynamic number of clusters with arbitrary shapes and a variable number of individuals in each cluster so that CLS is agnostic to the number of clusters. Large number of decision variables can be a serious challenge for clustering especially with low number of samples. Principal component analysis (PCA) [14] is employed to reduce the dimensionality when necessary.

3.1 Density Based Spatial Clustering

Spatial clustering uses similarity in spatial attributes between objects in the space where close dense regions are merged to form one cluster [15]. DBSC [13] is a spatial clustering method with few parameters to tune. The number of clusters generated by DBSC is dynamic, it is estimated by the algorithm based on the locations of the objects in the space [15], resulting in clusters with arbitrary shapes. DBSC starts by randomly selecting a point n from the space. A neighborhood of n contains all the points in the space located in a circle surrounding n with a predefined radius $\epsilon : dist(n, m) \geq \epsilon$. The neighborhood $N_{\epsilon}(n)$ is only created if it contains a minimum number of points $MinPts$. This is repeated for every point in the current cluster to check whether the cluster can be expanded. Expanding a cluster is done by adding all the reachable points by the cluster's neighborhood. When the cluster can no longer be expanded, another point is selected and the same process is repeated. The algorithm terminates when every point belongs to a cluster or labeled as noise (i.e. not reachable by any cluster). $MinPts$ and ϵ are the only required parameters.

3.2 The Algorithm

CLS aims at covering all the potentially good regions in both the objective and the solution spaces and maintaining a good level of diversity in both of them. This is done by clustering the particles in the objective and solution spaces and incorporating the diversity information using ϵ dominance. PCA is applied to reduce dimensionality if necessary and then DBSC is applied to implement the clustering.

The particle location in the solution space is defined by its variable vector, while the objective vector defines the particle's location in the objective space. After applying DBSC in both spaces, two sets of clusters are created and then every particle (except noise) is assigned to a cluster in each space. This defines an indirect mapping between the two spaces. Should two particles belong to the same cluster in one space, it does not necessarily mean they belong to same cluster in the other space. Each particle is marked by an index in the population, and each cluster in both spaces has a unique index as well. These added information facilitate mapping the individual into the clusters. Fig. 1 demonstrates this indirect relation.

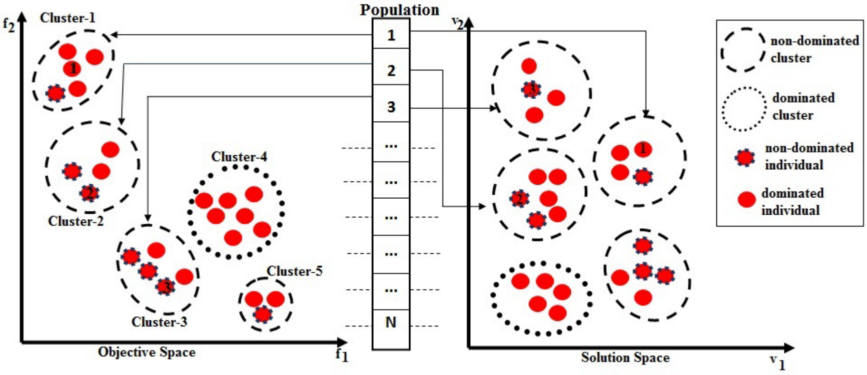


Fig. 1. Example of the algorithm at work while mapping clusters

A non-dominated cluster is a cluster in the objective space that contains at least one non-dominated individual. CLS exploits the relation between the non-dominated clusters in the objective space and the related clusters in the solution space. Following is a formal representation of the algorithm.

Let O be the set of clusters in the objective space and V the set of clusters in the solution space. A cluster $o \in O$ is called non-dominated cluster if and only if $\exists a \in o$ and $a \in PF$. The set of non-dominated clusters is then called $O^{'}$. Then we can define the function:

$$\phi(c) = \{v : v \in V and\ \exists X_a \in v : F_a \in c\} \tag{1}$$

where $c \in O$ and X_a is the image of an individual a in the solution space and F_a is the image of a in the objective space. For a set of clusters C in the objective space, the function $\Phi(C)$ is defined as:

$$\Phi(C) = \{\phi(c) : c \in C\} \tag{2}$$

The selection function Ψ is defined as follows:

$$\Psi : A \rightarrow \{0, 1\} \tag{3}$$

where A is the current population, and

$$\Psi(a) = \begin{cases} 1 & : F_a \in PF \\ 1 & : F_a \notin PF, F_a \in o, o \in O^{'}, \phi(o) \notin \Gamma \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\Gamma \subseteq V$ is the set of clusters selected so far. $\Gamma = \{c \in V : \exists\, a, X_a \in c, \Psi(a) = 1\}$.

Algorithm 1 outlines CLS approach to build the leaders' archive, where Apply-PCA applies PCA in both spaces when necessary. Apply-DBSC creates the clusters as discussed before. Continue-Original-Algorithm skips the building of the archive when no clusters can be found and continues with MOPSO, this might happens in early stages of the optimisation process where the particles are sparse due to the initialization. $\Lambda(a) \in V : X_a \in \Lambda(a)$.

Algorithm 1. CLS

```
[$X_a^*$, $F_a^*$]=Apply-PCA($X_a, F_a$)
[O,V]=Apply-DBSC($X_a^*, F_a^*$)
if $|O| = 0$ or $|V| = 0$ then
    Continue-Original-Algorithm
else
    for all $o \in O^{'}$ do
        for all $F_a \in o$ do
            if $\Psi(a) = 1$ then
                add a to leaders-archive
                add $\Lambda(a)$ to $\Gamma$
            end if
        end for
    end for
end if
```

The leaders archive is characterized by a limited size. ϵ-dominance is used to retain the diversity when the archive reaches its maximum size, taking into account that at least one particle from each non-dominated cluster in the objective space as well as one from the related clusters in the solution space must be maintained in the archive. Maintaining the clusters' representatives may not exceed the maximum size of the archive as the size of the archive is set to the population size and the maximum number of clusters possible is smaller than half the population size (i.e. each cluster contains at least 2 solutions). If the archive is not full yet, additional non-dominated solutions are added regardless of their corresponding clusters.

4 Experiments and Results

To verify our technique, CLS is used to substitute the archiving technique in OMOPSO to form OMOPSO/C. OMOPSO is chosen as it is an established MOPSO method. OMOPSO/C is tested on several standard test problems defined in the test suite [16]. The selected test problems are ZDT1-4, ZDT6, Viennet2, and Viennet3. These were chosen as they cover diverse MOPs with convex, concave, connected and disconnected PFs. ZDT1-ZDT4 and ZDT6 are two-objective problem with 30 decision variables. Viennet2 and Viennet3 are three-objective problems with 2 decision variables. OMOPSO/C uses PCA when solving ZDT1-4 and ZDT6. OMOPSO and OMOPSO/C use turbulence probability of 0.5. OMOPSO/C parameters are set experimentally, ϵ is set to 0.1 when solving 2-objective problems and 0.01 when solving 3-objective problems, and $MinPts$ is set to 3. Each algorithm is run 20 times for each test problem. For the 2-objective problems each algorithm uses 300 iteration per run, and 200 particles per generation. For the 3-objective problems the corresponding values of 400 iterations and 400 particles were used. OMOPSO parameters are set as recommended in [12].

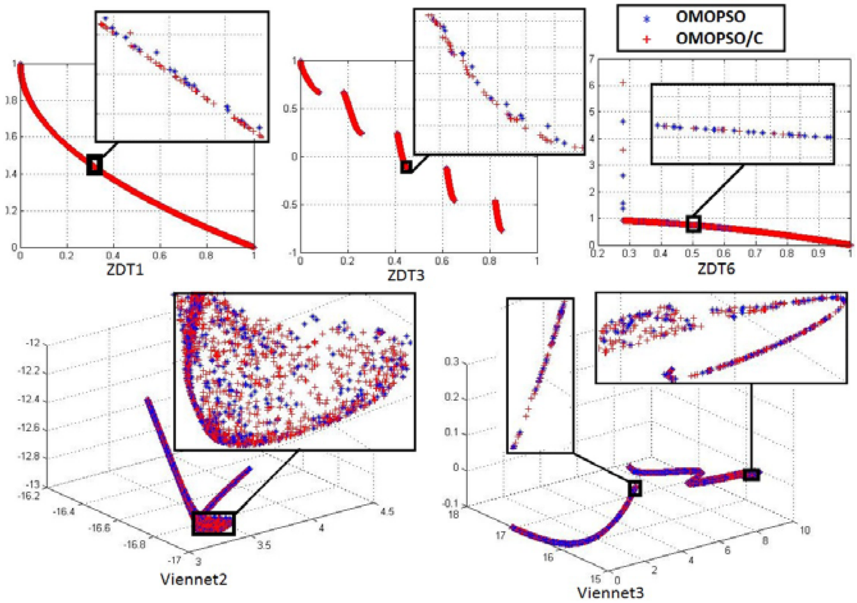


Fig. 2. PF approximations produced by OMOPSO and OMOPSO/C

Three indicators are used for measuring the convergence and diversity of the solutions produced by both methods[17].

Inverted generational distance (IGD) measures the uniformity of distribution of the obtained solutions in terms of dispersion and extension. The average distance is calculated for each point of the actual PF (PF_{True}) A and the nearest point of the approximation PF (PF_{approx}) B.

$$IGD(A, B) = (\sum_{a \in A} (\min_{b \in B} \parallel F(a) - F(b) \parallel^2))^{1/2} / |A| \tag{5}$$

Average generational distance (GD) measures only the closeness of the obtained solutions to PF, using Eq.5 but by assigning A to PF_{approx} and B to PF_{True}. Smaller IGD and GD means better PF_{approx}.

Set Coverage (SC) measure calculates the percentage of solutions in B that are dominated by at least one solution in A, where A and B are two approximation of the PF.

$$SC(A,B) = (|b \in B, \exists a \in A : a \succ b|)/|B| \tag{6}$$

Due to limited space only 5 approximated PF produced by OMOPSO and OMOPSO/C are demonstrated in Fig. 2. Table 1 presents the results of the three measures. The two methods seem to be performing similarly in general but with some advantage of OMOPSO/C in some problems. IGD results demonstrate better spread and closeness to PF for OMOPSO/C over OMOPSO in all problems except ZDT4. GD results indicate that OMOPSO/C is closer to the actual PF than OMOPSO for all MOPs. SC results show OMOPSO/C solutions to dominate solutions produced by OMOPSO for ZDT1-ZDT3 with similar dominance for the other problems.

Table 1. The Results of Applying The Three Measures Where A is PF_{approx} Obtained using OMOPSO and B is PF_{approx} Obtained Using OMOPSO/C

Problem:	**ZDT1**	**ZDT2**	**ZDT3**	**ZDT4**	**ZDT6**	**Viennet2**	**Viennet3**	**Avg.**
Inverted Generational Distance								
OMOPSO	2.01E-5	1.97E-5	4.22E-5	0.05	1.92E-5	7.33E-5	4.14E-5	0.0072
OMOPSO/C	1.04E-5	1.41E-5	1.94E-5	0.06	1.15E-5	3.87E-5	3.76E-5	0.0086
Average Generational Distance								
OMOPSO	3.48E-5	1.42E-5	5.73E-5	0.09	6.37E-4	5.24E-5	1.50E-5	0.0130
OMOPSO/C	1.82E-5	9.54E-6	4.27E-5	0.07	3.04E-4	2.68E-5	1.09E-5	0.0101
Set Coverage								
SC(A,B)	0%	0%	47%	2%	0%	73%	57%	25.57%
SC(B,A)	63%	46%	64%	1%	4%	80%	56%	44.85%

5 Conclusions

This work introduces a new leaders' selection technique within a multi-objective particle swarm optimisation. The method is based on simultaneous clustering in the solution and objective spaces and then mapping the two cluster sets to create a leaders' archive. The proposed approach, is then tested on 5 2-dimensional MOPs and 2 3-dimensional ones.

CLS is a general technique, different clustering algorithms may be used instead of DBSC, PCA can be substituted by any other dimensionality reduction technique, and the mapping definition can also be replaced by any other mapping function (e.g. one that incorporates prior knowledge about the MOP). Clusters in the objective space may also be ranked and a decision maker can be involved in directing the optimisation processes toward regions according to this prior knowledge. Eliminating the worst ranked clusters and re-sampling new solutions in regions of interest is another approach that worth further investigation.

References

1. Reyes-Sierra, M., Coello, C.A.C.: Multi-objective particle swarm optimizers: A survey of the state-of-the-art. Int. J. Comp. Intel. Res. 2(3), 287–308 (2006)
2. Wang, Z., Durst, G.L., Eberhart, R.C., Boyd, D.B., Ben Miled, Z.: Particle swarm optimization and neural network application for qsar. In: Int. Par. Dist. Proc. Sym., vol. 10, p.194 (2004)
3. Al Moubayed, N., Petrovski, A., McCall, J.: Multi-objective optimisation of cancer chemotherapy using smart pso with decomposition. In: 3rd IEEE Sym. Comp. Intel. IEEE, Los Alamitos (2011)
4. Knowles, J., Corne, D.W.: Approximating the nondominated front using the pareto archived evolution strategy. Evo. Comp. 8, 149–172 (2000)
5. Deb, K., Goldberg, D.E.: An investigation of niche and species formation in genetic function optimization. In: 3rd Int. Conf. Gen. Alg. Morgan Kaufmann Publishers Inc., San Francisco (1989)
6. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining convergence and diversity in evolutionary multiobjective optimization. Evo. Comp. 10(3), 263–282 (2002)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: Nsga-ii. IEEE Trans. Evo. Comp. 6(2), 181–197 (2002)
8. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Trans. Evo. Comp. 4(3), 257–271 (1999)
9. Wang, Y., Dang, C., Li, H., Han, L., Wei, J.: A clustering multi-objective evolutionary algorithm based on orthogonal and uniform design. In: Proc. Eleventh Conf. Congress on Evo. Comp., CEC 2009, pp. 2927–2933. IEEE Press, Los Alamitos (2009)
10. Al Moubayed, N., Petrovski, A., McCall, J.: A novel smart multi-objective particle swarm optimisation using decomposition. In: Schaefer, R., Cotta, C., Kołodziej, J., Rudolph, G. (eds.) PPSN XI. LNCS, vol. 6239, pp. 1–10. Springer, Heidelberg (2010)
11. Parsopoulos, K.E., Tasoulis, D.K., Vrahatis, M.N.: Multiobjective optimization using parallel vector evaluated particle swarm optimization. In: Proc. IASTED. ACTA Press (2004)
12. Reyes-Sierra, M., Coello, C.A.C.: Improving PSO-based multi-objective optimization using crowding, mutation and ϵ-dominance. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 505–519. Springer, Heidelberg (2005)
13. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proc. 2nd Int. Conf. Know. Disc. Data Min. (1996)
14. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
15. Basu, S., Banerjee, A., Mooney, R.: Semi-supervised clustering by seeding. In: Proc. 19th Int. Conf. Machine Learning (2002)
16. Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, Norwell (2002)
17. El-Ghazali, T.: Metaheuristics: from design to implementation. John Wiley & Sons, Chichester (2009)

Categorization of Wikipedia Articles with Spectral Clustering

Julian Szymański

Department of Computer Systems Architecture,
Gdańsk University of Technology, Poland
julian.szymanski@eti.pg.gda.pl

Abstract. The article reports application of clustering algorithms for creating hierarchical groups within Wikipedia articles. We evaluate three spectral clustering algorithms based on datasets constructed with usage of Wikipedia categories. Selected algorithm has been implemented in the system that categorize Wikipedia search results in the fly.

1 Introduction

Documents categorization based on similarity is one of the main goals of automatic knowledge organization. This approach finds many applications especially in information retrieval domain [1] where introducing structure of similarities allows to improve searching for relevant information. In the experiments presented here we evaluate spectral clustering methods used for finding groups of similar documents and organize them into hierarchy. We test three algorithms on datasets constructed from Wikipedia articles. Human made categories of these articles have been used as referential structures, which allows us to construct relevance sets used for evaluation. After evaluation we selected the best clustering method and it has been implemented in practical application that organizes Wikipedia search results into clusters that represent groups of conceptually similar articles instead of their ranked list.

2 Spectral Clustering

One of the important groups of clustering algorithms is spectral clustering. The method is based on cutting the graph of object's similarities using methods of spectral graph theory [2]. In recent years this theory has been strongly developed, especially in direction of graph clustering algorithms where the most well known are: Shi–Malik (2000), Kannan–Vempala–Vetta (2000), Jordan–Ng–Weiss (2002) and Meila–Shi (2000).

If we consider clustering in terms of graph theory we can introduce measure that describe partition of graph nodes into two subsets. If we have weighed graph $G = (V, E)$, it's partition into two sets of nodes (A and B): $A \cap B = \emptyset$ and $A \cup B = V$ can be specified with cutset number [3] defined as:

$$Cut(A, B) = \sum_{u \in A, v \in B} w(u, v). \tag{1}$$

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 108–115, 2011.

Representing the objects that are to be clustered with graph nodes and the w weights of the graph edges with objects similarities the partitioning problem is reduced to finding optimal cutset. The problem is computationally polynomial [3]. Sometimes cutset gives the results that are different than intuitive nodes partition, so other measures are introduced:

Normalized cut (NCut) ([2]), that increases its value for clusters that have nodes with small sum of edge weights.

$$NCut(A,B) = \frac{Cut(A,B)}{Vol(A)} + \frac{Cut(A,B)}{Vol(B)}, \tag{2}$$

where

$$Vol(A) = \sum_{u \in A} \sum_{v \in V} w(u,v) \tag{3}$$

Multiway Normalized Cut ($MNCut$) ([4]) promotes stronger relations between elements in one cluster. It also allows to describe more precisely more separated clusters.

$$MNCut(\Delta) = \sum_{i=1}^{K} (1 - \frac{Cut(C_i, C_i)}{VolC_i}) \tag{4}$$

It is known the spectral clustering methods give high quality partitions [4] and have good computational complexity [3] [5]. There are many variants of spectral algorithms. Mainly they differ in the way of eigenvectors calculation and usage. What is common – they treat source objects as graph nodes and then they are mapped into points in n-dimensional space. This space is constructed using spectral analysis and there is performed essential clustering. We can select three main steps of spectral algorithms ([6]):

1. Preprocessing and data normalization
   At this stage the data are preprocessed into their computational representation. If we are clustering the text documents typically Vector Space Model (VSM) [7] is used. In this representation weighting with Term Frequency and Inverse Document Frequency (TFIDF) [8] allows to calculate similarities between documents. In our experiments we use cosine distance which is known to be the suitable similarity measure for sparse vectors [9].
2. Spectral mapping
   This stage distinguished spectral approach. Using the data from step 1 the typically Laplacian matrix is built and then appropriate number of its eigenvectors is calculated.
3. Clustering
   The objects represented with spectral mapping are divided into two or more sets. Sometimes it is enough to find appropriate cut of the n-element, sorted collection which divide this collection into two clusters. In other methods this step is more complicated and performs partitioning in new representation space (provided by spectral mapping) using standard clustering algorithm eg. k-Means [10].

In our experiments we test three spectral clustering algorithms:

1. Shi–Malik [11] (SM), is realized in following steps:
   (a) Calculate eigenvectors of similarity Laplacian graph.
   (b) Sort elements of the dataset according to second smallest eigenvector value, which is denoted as x_1, x_2, ..., x_n,
   (c) Calculate the partition $\{\{x_1, x_2..., x_i\}, \{x_{i+1}, x_{i+2}, ..., x_n\}\}$ $(1 \leq i \leq n-1)$ having the smallest $NCut$.
   (d) If given partition has $NCut$ value smaller than given a priori value (that means it is better) then this method in each of the divided sets is run again, otherwise the algorithm stops.

2. Kann–Vempala–Vett algorithm [6] (KVV) is a heuristic method that finds graph cut which minimizes two parameter quality function, called conductance. Considering partition $(S, \bar{S})$ of graph $G = (V, E)$, where $\bar{S} = V \setminus S$ is a function defined with formula

   $$\phi(S) = \frac{Cut(S, \bar{S})}{min(Vol(S), Vol(\bar{S}))} \tag{5}$$

   In comparison to previous one the algorithm instead of Laplacian matrix operates on normalized similarity matrix and uses second biggest (instead second smallest) eigenvector.

   The algorithm goes as follows:

   (a) Normalization performed by dividing each of row elements by the sum of elements in each row.
   (b) For each of clusters C_i created so far, create matrix of similarity $\mathbf{W}_i$ from the nodes the cluster.
   (c) Normalize each matrix $\mathbf{W}_i$ by inserting at the position on diagonal value that complete sum of the elements in this row to 1.
   (d) Calculate second biggest eigenvector of $\mathbf{v}_2^i$ of matrix $\mathbf{W}_i$.
   (e) For each C_i sort its elements according to value of respective coordinate of $\mathbf{v}_2^i$. We denote as x_1^i, x_2^i, ..., $x_{n_i}^i$ sequence of ordered objects form cluster C_i.
   (f) For each C_i find cut in the form of $\left\{\left\{x_1^i, x_2^i..., x_j^i\right\}, \left\{x_{j+1}^i, x_{j+2}^i, ..., x_{n_i}^i\right\}\right\}$ which has smallest conductance.
   (g) Find cluster C_i with the smallest conductance for a given cut and divide it according this cut and replace cluster C_i with two new clusters.
   (h) If given number of clusters has not been reached go to 2.

3. Jordan–Ng–Weiss algorithm [4] (JNW) is a partitioning algorithm – in one iteration it creates flat clusters, given by a priori K parameter. This parameter denotes also the number of used eigenvectors.
   The algorithm performs following steps:
   (a) Calculate Laplacian of similarity matrix
   (b) Calculate K biggest eigenvectors of Laplacian matrix $\mathbf{v}_1$, $\mathbf{v}_2$, ..., $\mathbf{v}_K$.

(c) Perform spectral mapping from set V (original nodes) to $\mathbb{R}^K$:
$map(i) = [\mathbf{v}_1(i)\ \ \mathbf{v}_2(i)\ \ ...\ \ \mathbf{v}_K(i)]^T$ $(1 \leq i \leq n)$.
(d) Perform clustering of the points $map(i)$ in space $\mathbb{R}^K$.
(e) Return result of partitioning of entrance elements into K clusters.

Because JNW is a partitioning approach, the last step is not so simple as in previous algorithms. It requires usage of other clustering method to divide n elements in K-dimensional space. The authors of the algorithm used one of the simplest approaches of k-means.

3 Experiments

In our experiments we compare three algorithms in application to text data. We find Wikipedia[1] dumps very useful to evaluate our approach for documents categorization. These data are easy to preprocess as well as cover wide spectrum of human knowledge thus provide very varied set of documents. What is the most beneficial Wikipedia dumps provide also human made categories that can be used during evaluation process.

3.1 The Data

For our experiments we construct eight test data packages. Each package has been constructed with articles from selected Wikipedia super categories and their sub categories retrieved to selected level. The details of each package are shown in Table 1.

Table 1. Descriptions: n – number of nodes (in brackets — unisolated), l – non zero elements in neighborhood matrix, N – number of upercategories , k – number of all categories, d – depth of category tree), P – number of categories to Wikipedia root category

n.o.	Name	n	l	N	k	d	P	Comment
1	Z01	575 (575)	67099	2	18	1	3	2 distant categories: Distance_Education i Science_Experiments.
2	Z02	1157 (1156)	323901	5	35	1	3	5 distant categories: Caligraphy, General_Economics, Military_logistics, Evolution, Analytic_number_theory.
3	Z03	3905 (3903)	2260919	8	102	1	3	8 distant categories: Geometric_Topology, Epistemology, Rights, Aztec, Navigation, Clothing_companies, Protests, Biological_Evolution.
4	Z04	3827 (3826)	3195963	2	204	6	4	Two distant categories at the same hierarchy level: Criticism_of_journalism i Corporate_crime.
5	Z05	3647 (3644)	1682361	6	213	6	5	6 distant categories at high level of abstraction: DIY_Culture, Emergence, Military_transportation, Formal_languages, Geology_of_Autralia, Computer-aided_design.
6	Z06	4750 (4747)	2568378	9	289	6	5	9 distant categories at high level of abstraction: DIY_Culture, Emergence, Military_transportation, Formal_languages, Geology_of_Autralia, Computer-aided_design, Special_functions, History_of_ceramics, Musical_thatre_companies.
7	Z07	4701 (4701)	4230139	2	298	6	4	2 neighboring categories at high abstraction level: Computer_law i Prosecution.
8	Z08	5717 (5716)	11288283	4	893	6	4	4 neighboring categories at high abstraction level: Impact_events, Droughts, Volcanic_events, Storm.

[1] http://download.wikimedia.org

3.2 Results

There are many approaches to cluster validation. In our experiments we evaluate results according to external criteria which is known to be harder task than evaluate them according to internal criteria [12]. Our validation we made using standard clustering quality measures (described below) that have been compared to relevance set formed by Wikipedia categories. To make evaluation easier, on each of hierarchy levels we took parameter K (the number of clusters) from referential set.

External cluster validation criteria measure the similarity of the structure of the clusters provided by the algorithm to a priori known structure that is expected to be achieved [13]. Validated cluster structure we denote as $C = \{C_1, ..., C_K\}$, and reference set as $P = \{P_1, ..., P_s\}$. Our source set of objects (articles) is denoted as X, and its cardinality with N. Unordered pairs $\{x_i, x_j\}$ of X elements may fall into four cases:

(a) elements x_i and x_j that belong to the same cluster as well in C as in P,
(b) elements x_i and x_j that belong to the same cluster in C, but not in P,
(c) elements x_i and x_j that belong to the same cluster in P, but not in C,
(d) elements x_i and x_j that belong to different cluster both in C and in P.

The symbols a, b, c and d denote numbers of elements in respectively cases. Note they correspond to values in confusion matrix (respectively TP FN FP TN) [14].

Additionally to perform validation two matrices $\mathbf{X}$ i $\mathbf{Y}$ are defined. The first one describes clusters from C, the second from P. Value one at the position (i, j) denotes the pair of elements (x_i, x_j) that belong to different clusters in the structure.

To evaluate our results we use standard clustering quality measures:

- Rand statistics $R = \frac{a+d}{M}$. R value depends on number of objects pairs having their mutual position (same / other cluster) the same in cluster and validation structure. R is in the range $[0, 1]$ and is greater while two compared structures are more similar (1 is when they are identical)
- Jaccard coefficient $J = \frac{a}{a+b+c}$ is similar to R but numerator and denominator are decreased by d value. Similarly to R maximum of Jaccard coefficient is 1 when $b + c = 0$ and it denotes situation when two structures are the same.
- Fowlkes–Mallows index $FM = \frac{a}{\sqrt{(a+b)(a+c)}}$. The FM value grows when a increases and when b and c decrease. The maximum $FM = 1$ when $b = c = 0$.
- Hubert statistics $\Gamma = \frac{1}{M} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} X(i, j)Y(i, j)$. The Γ value grows when the number of elements having same relative position in validated and referential structure increases. $\Gamma_{max} = 1$ when these two structures form N one-element clusters.

In Figure 1 we show metrics values of external validation criteria formed with methods KVV, JNW and SM. Each figure corresponds to the one test package (presented in Table 1), colors denote lines for quality metrics: Rand statistics (R) – blue, Jaccard index (J) – red, Fowlkes–Mallows index (FM) – green and Hubert statistics (Γ) – orange. Different clustering algorithms have been denoted with different line patterns.

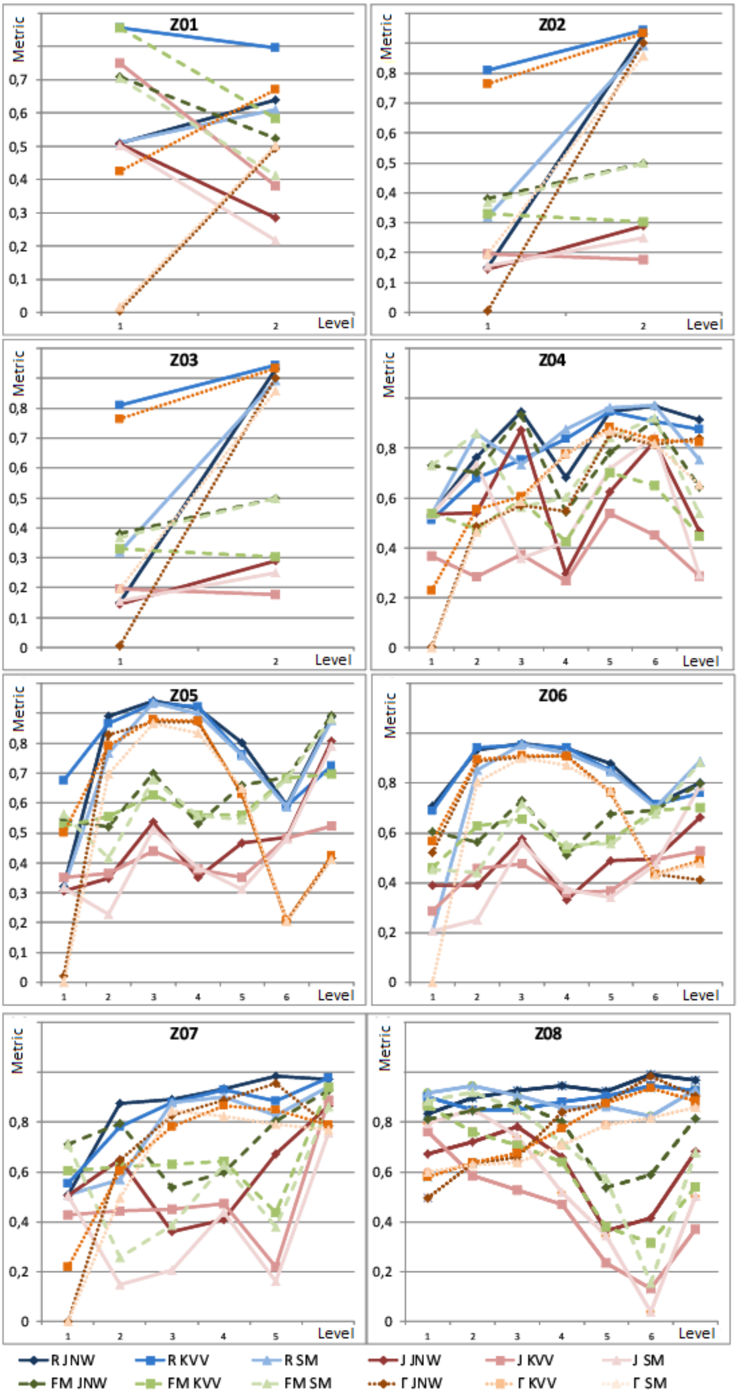


Fig. 1. Results of cluster validation in packages Z01 - Z08

4 Discussion

What can be seen from graphs presented in Figure 1 is strong correlation of Rand (R) with Hubert (Γ) statistics as well as Fowlkes–Mallows (FM) with Jaccard index (J). It is caused by high value of a parameter which a denotes number of pairs in one cluster in validated and referential structure. The second parameter that has high influence on metrics is d which denotes a number of objects that were assigned to different clusters in referential and in validation set. Metrics J and FM does not use parameter d, statistics R and Γ involve it in dominator and nominator which cause similarity of these measures.

In some areas we can see differences between metric pairs J and FM as well as R and Γ. Eg. for test package Z04 at level 4 J value for KVV algorithm is low while Γ is high. It is because the structure of the metrics J does not use d parameter, but Γ does. In comparison at 3rd level of hierarchy in this test d value grows (from 3.652.328 to 3.819.940), but a value decreases (from 939.463 to 374.220). Because of the increased d value Γ also increases, while the decrease of a caused the decrease of J value. Similarly there can be substantiated decrease Γ while FM grows (eg. at level 6 of test package Z06). In this case we can see the increase of a and decrease of d. It suggests that on level 6 in referential structure there are fewer categories than at level 5. Indeed number of categories at level 5 is 895 and at level 6 – 503.

The highest values of R and J parameters have been obtained with JNW algorithm. Only one case, when other methods gave better results, is a case described above (level 4 in package Z04). Also FM measure pointed out JNW algorithm as the best. The highest values of Γ in packages Z07 and Z08 we obtain using JNW algorithm, the others have been obtained using KVV.

From the above observations we may conclude the best measure for cluster validation is Rand statistics. It has been less dependent on cluster structure changes, at succeeding hierarchy levels in test packages. It is especially important when a parameter is increasing (growing number of small clusters) in correlation with growing d which causes metric decreases.

In most of the cases the best clusters have been achieved using JNW algorithm, thus we used it in our practical application.

4.1 Practical Application and Future Work

Based on JNW algorithm we have created a prototype system named WikiClusterSearch. It automatically organizes the results of searching Wikipedia for a given keyword. In the system user may specify a searched phrase and the articles containing it are organized into clusters in the fly. It allows to present directions in which user may continue his or her search. For the efficiency reasons for now only Polish Wikipedia is supported (English one is approximately 5 times bigger). WikiClusterSearch (WCS) has demonstrated the proposed approach can be used to obtain a good quality hierarchy of clusters. The system is available on line under http://swn.eti.pg.gda.pl/UniversalSearch.

In future we plan to develop the JNW approach and use clustering algorithm based on densities [15] instead of k-means. We also plan to implement our system for English Wikipedia what requires to improve its architecture. The long term goal is to join the method of retrieving the information based on clusters of Wikipedia categories with classifier [16] that allows to categorize linear search results returned by search engine into these categories.

We also plan to perform experiments on large scale clustering - it is on the whole Wikipedia. The experiments using well tuned clustering algorithm will allow to improve category system of Wikipedia finding, missing and wrong assignments articles to categories.

Acknowledgements. The work has been supported by the Polish Ministry of Science and Higher Education under research grant N519 432 338.

References

1. Manning, C., Raghavan, P., Schütze, H., Corporation, E.: Introduction to information retrieval, vol. 1. Cambridge University Press, Cambridge (2008)
2. Cvetkovic, D., Doob, M., Sachs, H.: Spectra of Graphs–Theory and Applications, III revised and enlarged edition. Johan Ambrosius Bart. Verlag, Heidelberg (1995)
3. Vazirani, V.: Algorytmy aproksymacyjne. WNT Warszawa (2005)
4. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems, vol. 2, pp. 849–856 (2002)
5. Kannan, R., Vetta, A.: On clusterings: Good, bad and spectral. Journal of the ACM (JACM) 51, 497–515 (2004)
6. Verma, D., Meila, M.: A comparison of spectral clustering algorithms. University of Washington, Tech. Rep. UW-CSE-03-05-01 (2003)
7. Salton, G., Wong, A., Yang, C.: A vector space model for automatic indexing. Communications of the ACM 18, 613–620 (1975)
8. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management 24, 513–523 (1988)
9. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: KDD Workshop on Text Mining, vol. 400, pp. 525–526. Citeseer (2000)
10. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S.: Constrained k-means clustering with background knowledge. In: Proceedings of the Eighteenth International Conference on Machine Learning, vol. 577, p. 584. Citeseer (2001)
11. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
12. Eldridge, S., Ashby, D., Bennett, C., Wakelin, M., Feder, G.: Internal and external validity of cluster randomised trials: systematic review of recent trials. Bmj 336, 876 (2008)
13. Yeung, K., Haynor, D., Ruzzo, W.: Validating clustering for gene expression data. Bioinformatics 17, 309 (2001)
14. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proceedings of the Fifteenth International Conference on Machine Learning, vol. 445. Citeseer (1998)
15. Kriegel, H., Pfeifle, M.: Density-based clustering of uncertain data. In: Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, vol. 677. ACM, New York (2005)
16. Szymański, J.: Towards automatic classification of wikipedia content. In: Fyfe, C., Tino, P., Charles, D., Garcia-Osorio, C., Yin, H. (eds.) IDEAL 2010. LNCS, vol. 6283, pp. 102–109. Springer, Heidelberg (2010)

Is Genetic Programming "Human-Competitive"? The Case of Experimental Double Auction Markets

Chen Shu-Heng and Shih Kuo-Chuan

AIECON research center, Department of Economics, National Chengchi University,
Taipei, Taiwan
{chen.shuheng,melvinshih}@gmail.com

Abstract. In this paper, the performance of human subjects is compared with genetic programming in trading. Within a kind of double auction market, we compare the learning performance between human subjects and autonomous agents whose trading behavior is driven by genetic programming (GP). To this end, a learning index based upon the optimal solution to a double auction market problem, characterized as integer programming, is developed, and criteria tailor-made for humans are proposed to evaluate the performance of both human subjects and software agents. It is found that GP robots generally fail to discover the best strategy, which is a two-stage procrastination strategy, but some human subjects are able to do so. An analysis from the point of view of cognitive psychology further shows that the minority who were able to find this best strategy tend to have higher working memory capacities than the majority who failed to do so. Therefore, even though GP can outperform most human subjects, it is not "human-competitive" from a higher standard.

Keywords: Experimental Markets, Double Auctions, Genetic Programming, Human-Competitiveness, Working Memory Capacity.

1 Introduction and Motivation

The Double Auction market (DA) has a long history in experimental economics [1]. It serves as an important foundation for public policy and market mechanism designs [2]. It also plays an important role in the understanding of the strategic behavior of individual agents [3] [4] [5] [6]. Recently, it has been studied from a psychology viewpoint to observe the effects of cognitive capacity on traders' and the market's performance [7] [8]. It has been intensively simulated by agent-based models using software agents [9].

Chen and Yu [9] proposed a series of agent-based simulations on the double auction markets, using one kind of machine learning, namely, genetic programming (GP), in their simulation. It was found that GP agents are smart in terms of market timing. Specifically, they attempted to postpone their participation in the market transaction so as to avoid early competition and become a *monopsonist* at a later stage. This strategy is then called the *theory of optimal procrastination* to signify the best way to trade under a specific market environment. As explained in Koza *et al.*

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 116–126, 2011.

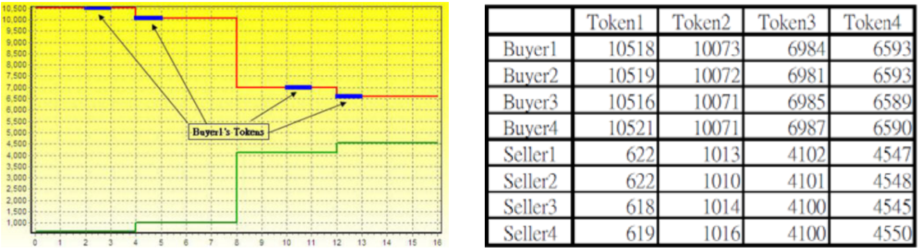
[10], eight criteria have been described in the field of artificial intelligence, machine learning, and GP; based on those criteria, GP in 36 instances are shown to be human-competitive. In this vein, we ask the question: can we have an additional instance of showing the human-competitiveness of GP? In other words, is GP human-competitive in the double auction markets?

To rigorously compare GP agents with human subjects, we need to observe whether the agent has learned the theory of optimal procrastination and when. In addition, if the agent failed to learn the theory, we need to know how far he is away from there. Only after we can answer these questions, can a comparison of the performance of human subjects and software agents be meaningfully conducted. The earning that the agent obtained in an experiment can certainly tell us something about whether the agent has learned, but it is not precise enough to indicate when he learned and how far away he was if he did not learn. In this paper, we therefore present a learning index which is able to dynamically characterize the agent's learning status.

The rest of the paper is organized as follows. Section 2 describes the design of the double auction market experiments. Section 3 is the main contribution of the paper. The work here is divided into three parts. Section 3.1 first transforms the double auction experiments into an integer programming problem, and the optimal solution to the problem serves as the benchmark on which the performance comparisons of GP and human subjects are based. Section 3.2 then develops a learning index to evaluate the learning performance of human subjects during the experiments. The essence of this proposed learning index is to separate very different learning dynamics of human subjects. Section 3.3 further develops the accident-tolerance criteria, which are tailor-made for humans. Using these criteria, we are able to identify whether the human subjects have learned and when. Section 4 applies these criteria to classify the human subjects into two groups, the group that learned and the group that did not, and then study the contribution of cognitive capacity to this difference. Section 5 presents the concluding remarks.

2 Human Experiments

Based on the experiment settings of Chen and Yu [9], four experimental double auction markets are designed. Each market is characterized by a supply-demand curve which indicates the value of tokens assigned to 4 buyers and 4 sellers. This assigned token value controls the buyer's highest willingness to pay and the seller's minimum acceptable price. An example (Market IV) is given in Figure 1. In order to make deals, buyers and sellers can submit bids or asks at each step; however, only the buyer with the highest bid (current bid) and the seller with the lowest ask (current ask) are qualified to engage in transactions. If there are two or more qualified buyers or sellers, one of them will be determined randomly. Of course, a transaction can be made only if the current bid is greater than or equal to the current ask, and in this case the transaction price will be the average of the two. The gain from the trade is then simply the difference between the transaction price and the value of the respective token. The process lasts for 25 steps unless all possible deals are finished earlier. To give human subjects the opportunity to learn from their experience, a duration of 30 periods is set for each experiment.



	Token1	Token2	Token3	Token4
Buyer1	10518	10073	6984	6593
Buyer2	10519	10072	6981	6593
Buyer3	10516	10071	6985	6589
Buyer4	10521	10071	6987	6590
Seller1	622	1013	4102	4547
Seller2	622	1010	4101	4548
Seller3	618	1014	4100	4545
Seller4	619	1016	4100	4550

Fig. 1. The Supply-Demand Curve for Market IV

In our experiments, one human subject is matched with seven software agents. All human subjects are assigned the same role, specifically, Buyer One. Other buyers and sellers are taken up by software agents that are designed using the truth-teller strategy, which is to bid and ask at the given token value. This design is equivalent to Chen and Yu [9] except that Buyer One in their markets is a GP robot rather than a human subject. The fixed design makes it easier to trace and analyze the performance of either the human subject or the software agent.

A series of DA experiments was conducted in the year 2010. All subjects were college or graduate students of universities in Taipei. Furthermore, they had formerly taken part in another series of DA experiments [8] and were recruited from the experimental subject database of the AIECON Research Center. Moreover, the experiment's environment was developed using JAVA programs. Each human subject participated in the DA experiments in a computer laboratory. Before the experiments, a 90-minute tutorial was given to ensure that all subjects fully understand how the programs and the market mechanisms operated. However, they were not told that their opponents (the software agents) were truth tellers. To entice the subjects to do their best, the subjects were paid a fixed attendance fee and, depending on their market performance, some additional amounts. A total of 165 subjects participated in these experiments.

3 Performance Measurement

3.1 The DA Market as a Combinatorial Optimization Problem

The experimental design can be modeled as a *constrained combinatorial optimization problem* [11] or an integer programming problem, i.e., to maximize Buyer One's consumer's surplus or transaction gains (Equation (1)) subject to the 20 constraints (Equations (2) to (21)). Notations of the variables in these equations are given in Table 1. Many of the variables are Boolean. Basically, we use these Boolean variables and the inequalities and predicates derived to represent the trading mechanism (time flow, transaction schedule, trading sequence of tokens, correlations among bids, asks and transactions) associated with the double auction market in general and the market specifically comprised of truth tellers. Many of the constraints are applied to buyers and sellers symmetrically. To save space, we combine these symmetries into one constraint, but use the symbol || to separate them.

Objective Function.

$$max\ \sum_{j}^{np}\sum_{k}^{nt}[\left(btv_{1,k}-DV_{j}\right)*BW_{1,k,j}] \tag{1}$$

Constraints.

$$\forall b,j,k\ BW_{b,k,j}\leq BT_{b,k,j}\quad \|\quad \forall s,j,k\ SW_{s,k,j}\leq AT_{s,k,j} \tag{2}$$

$$\forall j\ \sum_{s}^{ns}\sum_{k}^{nt}SW_{s,k,j}=\sum_{b}^{nb}\sum_{k}^{nt}BW_{b,k,j} \tag{3}$$

$$\forall j\ \sum_{b}^{nb}\sum_{k}^{nt}BW_{b,k,j}\leq 1\quad \|\quad \forall j\ \sum_{s}^{ns}\sum_{k}^{nt}SW_{s,k,j}\leq 1 \tag{4}$$

$$\forall b,k\ \sum_{j}^{np}BW_{b,k,j}\leq 1\quad \|\quad \forall s,k\ \sum_{j}^{np}SW_{s,k,j}\leq 1 \tag{5}$$

$$\forall b\ \sum_{j}^{np}\sum_{k}^{nt}BW_{b,k,j}\leq nt\quad \|\quad \forall s\ \sum_{j}^{np}\sum_{k}^{nt}SW_{s,k,j}\leq nt \tag{6}$$

$$\forall b\in 2..4,j\ \ BV_{b,j}=\sum_{k}^{nt}\left(BT_{b,k,j}*btv_{b,k}\right) \tag{7}$$

$$\forall s,j\ \ AV_{s,j}=\sum_{k}^{nt}(AT_{s,k,j}*stv_{s,k}) \tag{8}$$

$$\forall b,j\ \sum_{k}^{nt}BT_{b,k,j}\leq 1\quad \|\quad \forall s,j\ \sum_{k}^{nt}AT_{s,k,j}\leq 1 \tag{9}$$

$$\forall b\ BT_{b,1,1}=1\quad \|\quad \forall s\ AT_{s,1,1}=1 \tag{10}$$

$$\begin{gathered}\forall b,j\in 2..np\ BT_{b,1,j}=BT_{b,1,j-1}-BW_{b,1,j-1}\quad \|\\ \forall s,j\in 2..np\ AT_{s,1,j}=AT_{s,1,j-1}-SW_{s,1,j-1}\end{gathered} \tag{11}$$

$$\begin{gathered}\forall b,k\in 2..nt,j\in 2..np\ BT_{b,k,j}=BT_{s,k,j-1}+BW_{s,k-1,j-1}\quad \|\\ \forall s,k\in 2..nt,j\in 2..np\ AT_{s,k,j}=AT_{s,k,j-1}+SW_{s,k-1,j-1}\end{gathered} \tag{12}$$

$$\forall s\ \sum_{j}^{np}SW_{s,nt+1,j}=0 \tag{13}$$

$$\forall j\ \sum_{b}^{nb}BB_{b,j}=1\quad \|\quad \forall j\ \sum_{s}^{ns}BS_{s,j}=1 \tag{14}$$

$$\forall b,j\ BB_{b,j}\geq \sum_{k}^{nt}BW_{b,k,j}\quad \|\quad \forall s,j\ BS_{s,j}\geq \sum_{k}^{nt}SW_{s,k,j} \tag{15}$$

$$\forall j\ BBV_{j}=\sum_{b}^{nb}(BB_{b,j}\times BV_{b,j})\quad \|\quad \forall j\ BAV_{j}=\sum_{s}^{ns}(BS_{s,j}\times AV_{s,j}) \tag{16}$$

$$\forall b,j\ BBV_{j}\geq BV_{b,j}\quad \|\quad \forall s,j\ BAV_{j}\leq AV_{s,j} \tag{17}$$

$$\forall j\ \sum_{b}^{nb}\sum_{k}^{nt}BW_{b,k,j}=D_{j} \tag{18}$$

$$\forall j\ (BBV_{j}-BAV_{j})\times D_{j}\geq 0 \tag{19}$$

$$\forall j\ (BBV_{j}-BAV_{j})<D_{j}\times bm \tag{20}$$

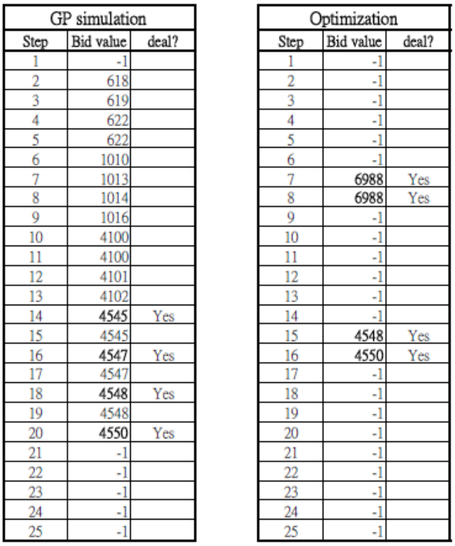
$$\forall j\ 2\times DV_{j}=(BBV_{j}+BAV_{j}) \tag{21}$$

Table 1. Description of Variables

b	Index for buyers	nb	number of buyers
s	Index for sellers	ns	number of sellers
j	Index for steps	np	number of steps
k	Index for tokens	nt	number of tokens
$btv_{b,k}$	value of token k for buyer b	$stv_{s,k}$	value of token k for seller s
bm	A very big number		
$BW_{b,k,j}$	The judgment of the for buyer b for token k at step j [0,1]	$SW_{s,k,j}$	The judgment of the winner for seller s for token k at step j [0,1]
$BT_{b,k,j}$	The judgment of the bidded token for buyer b for token k at step j [0,1]	$AT_{s,k,j}$	The judgment of the asked token for seller s for token k at step j [0,1]
$BV_{b,j}$	The bidded value of buyer b at step j	$AV_{s,j}$	The asked value of seller s at step j
$BB_{b,j}$	The judgment of the best buyer for buyer b at step j [0,1]	$BS_{s,j}$	The judgment of the best seller for seller s at step j [0,1]
BBV_j	The best bid value at step j	BAV_j	The best ask value at step j
D_j	The judgment of whether a deal is made at step j [0,1]	DV_j	The deal value at step j

The best trading strategy as a solution for the problem (1)-(21) can be obtained by applying the branch-and-bound method to solve the formulated integer programming problem. Taking Market IV as an example, we solve the constrained combinatorial optimization problem and present the unique optimal strategy in the right panel of Figure 2. The last column of the panel articulates the best market timing. The time to enter the market is denoted by “Yes”. The bid is shown in the second column of this panel. A value of “-1” means “to pass” (no action). As we can now read from this example, the optimal time to enter the market is at step 7, 8, 15, and 16 with a bid of 6,988, 6,988, 4,548, and 4,550 respectively. In this way, Buyer One can earn a maximum profit of 17,067. According to the Chen and Yu [9], this optimal strategy can be understood as a *two-stage procrastination strategy*: Buyer One holds his first two bids up to step 7 and the last two bids up to step 15. The intuition behind this two-stage procrastination is that, as shown in Figure 1, there is a sharp fall in the market demand curve accompanied by the sharp rise in the supply curve. This large-change topology suggests dividing the sequence of actions into two, one before the change and one after the change.

Using this optimal solution as a baseline, other solutions either found by software agents or human agents can then be compared. Chen and Yu [9] applied GP to solve the same problem, and the most commonly found solution is shown in the left-panel of Figure 2. Both the market timing and the bids are in sharp contrast to the benchmark. Basically, instead of using the two-stage procrastination strategy, the rule found by GP is a kind of one-stage procrastination strategy, which can be imagined as a local optimum trapped in a rugged landscape. However, with the presence of the discontinuity topology of Market IV, a single procrastination fails to lead to the maximum profit. In the same way, we will examine the solutions found by human subjects, possibly by their heuristics or gut feeling during the experiments, and evaluate their performance in relation to GP.



GP simulation		
Step	Bid value	deal?
1	-1	
2	618	
3	619	
4	622	
5	622	
6	1010	
7	1013	
8	1014	
9	1016	
10	4100	
11	4100	
12	4101	
13	4102	
14	**4545**	Yes
15	4545	
16	**4547**	Yes
17	4547	
18	**4548**	Yes
19	4548	
20	**4550**	Yes
21	-1	
22	-1	
23	-1	
24	-1	
25	-1	

Optimization		
Step	Bid value	deal?
1	-1	
2	-1	
3	-1	
4	-1	
5	-1	
6	-1	
7	**6988**	Yes
8	**6988**	Yes
9	-1	
10	-1	
11	-1	
12	-1	
13	-1	
14	-1	
15	**4548**	Yes
16	**4550**	Yes
17	-1	
18	-1	
19	-1	
20	-1	
21	-1	
22	-1	
23	-1	
24	-1	
25	-1	

Fig. 2. Trading Strategy: GP [9] (Left Panel) and the Benchmark (Right Panel)

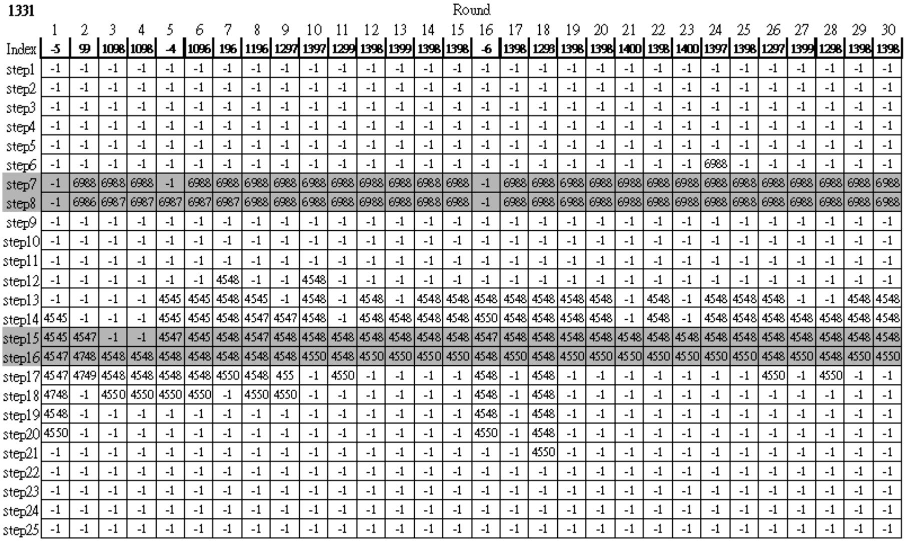
3.2 Learning Index

Observing the learning behavior of the human subjects can sometimes be a very perplexing task. Humans are emotional beings with physical limits and are not fault free. Their behavior in the laboratory may be difficult to replicate by any software agent, which generally does not share the human nature. This fundamental difference may call for a tailor-made evaluation design for humans, if we do not want to blindly treat humans as mechanical beings. In this subsection, we are going to develop a learning index, which can help us to better answer the questions with regard to the learning behavior of human subjects, in particular, whether they learned and when. We attempt to have a scoring system which can separate or self-classify very different types of learning behavior. As a concrete example, what is proposed in this section is a learning index built upon three criteria, namely, *the maximum earning*, *precise moment*, and *minimum effort*. These three provide some related but not redundant information about the learning behavior of human subjects. The first two criteria inform us of the global and local behavior of subjects, whereas the last one tells us about the trial-and-error effort made by the subject.

Maximum Earning. Regardless of what the subject bid and when he bid, as long as he can earn the maximum trading profit, a substantial 1,000 points are granted to distinguish the subjects who may have found, or have the potential to find, the best strategy from those who simply have no clue yet.

Precise Moment. In our experiments, many subjects failed to earn the maximum profit, but they may still learn part of the structure. To characterize the degree of partial learning, 100 points will be given for any of their single bids which match well with one of the bids in the optimal strategy, by time and by value.

Minimum Effort. Minimum effort means that subjects will trade by minimizing the number of the bidding frequencies. Take Figure 2 as an example. As shown by the right panel, only four times are required to bid. Hence, additional frequencies of bids will be considered unnecessary. While these may do no harm to the accumulated profits of the subject, the unnecessary effort made to trade may signal the possibility that the subject is still in a trial-and-error process, even though he almost already has the best solution. Therefore, minus one point shall be given for each of these unnecessary trials to downwardly adjust his learning performance.



1331	Round																													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Index	**-5**	**99**	**1098**	**1098**	**-4**	**1096**	**196**	**1196**	**1297**	**1397**	**1299**	**1398**	**1399**	**1398**	**1398**	**-6**	**1398**	**1298**	**1398**	**1398**	**1400**	**1398**	**1400**	**1397**	**1398**	**1297**	**1399**	**1298**	**1398**	**1398**
step1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step3	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step4	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step5	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step6	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	6988	-1	-1	-1	-1	-1	-1
step7	-1	6988	6988	6988	-1	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	-1	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988
step8	-1	6986	6987	6987	6987	6987	6987	6988	6988	6988	6988	6988	6988	6988	6988	-1	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988	6988
step9	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step10	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step11	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step12	-1	-1	-1	-1	-1	-1	4548	-1	-1	4548	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step13	-1	-1	-1	-1	4545	4545	4548	4545	-1	4548	-1	4548	-1	4548	4548	4548	4548	4548	4548	4548	-1	4548	-1	4548	4548	4548	-1	-1	4548	4548
step14	4545	-1	-1	-1	4545	4545	4548	4547	4547	4548	-1	4548	4548	4548	4548	4550	4548	4548	4548	4548	-1	4548	-1	4548	4548	4548	4548	4548	4548	4548
step15	4545	4547	-1	-1	4547	4545	4548	4547	4548	4548	4548	4548	4548	4548	4548	4547	4548	4548	4548	4548	4548	4548	4548	4548	4548	4548	4548	4548	4548	4548
step16	4547	4748	4548	4548	4548	4548	4548	4548	4548	4550	4548	4550	4550	4550	4550	4548	4550	4548	4550	4550	4550	4550	4550	4550	4550	4548	4550	4548	4550	4550
step17	4547	4749	4548	4548	4548	4548	4550	4548	455	-1	4550	-1	-1	-1	-1	4548	-1	4548	-1	-1	-1	-1	-1	-1	-1	4550	-1	4550	-1	-1
step18	4748	-1	4550	4550	4550	4550	-1	4550	4550	-1	-1	-1	-1	-1	-1	4548	-1	4548	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step19	4548	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4548	-1	4548	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step20	4550	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4550	-1	4548	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step21	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	4550	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step22	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step23	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step24	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
step25	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Fig. 3. Learning Index Applied to Subject #1331

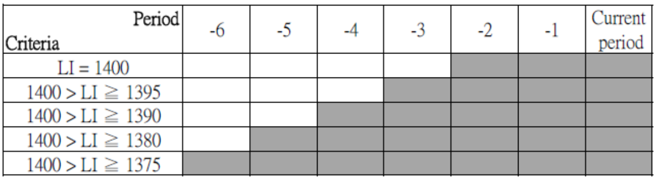
As an illustration, the proposed index applied to Subject #1331 is shown in Figure 3. The second row (the "Index" row) gives the sum of three scores for each of the three criteria. For example, the last column (period 30) shows that the subject did earn the maximum trading profits. In addition, the bids and bidding times are the same as the optimal solution. Hence, based on the first two criteria, a sum of 1,400 points (1000+4*100) is given to him. However, he also made two unnecessary early bids for the last two tokens; by the third criterion, he lost two points (-2). Therefore, his LI over the three criteria at period 30 is 1398 points. This, compared to the initial levels of "-5" (period 1) and "98" (period 2), shows a significant improvement. One, however, has to notice that LI, very typically, is not monotonically increasing in time. The LI of the subject falls sharply from a peak of "1,398" in period 15 to "-6" in period 16. This subtlety raises an issue with regard to the learning stability of humans and compels us to think harder as to whether and when the subject has learned the optimal strategy, an issue to which we now turn.

3.3 Learned or Not? An Accident-Tolerance Criterion

The next issue is to use the learning index (LI) developed above to decide whether the subject has actually learned the best strategy. This decision is more subtle than what one might think. Using the example above, can we consider a subject with a score of 1,400 to be a subject who has learned? The answer is *yes*, if he could repeatedly gain this score, but what happens if he does not. The idea to be discussed below is to allow for a kind of deviation which we shall call an *accident* and to develop an *accident-tolerance criterion* for determining whether the agent has learned.

To begin, let us consider the landscape of LI. The proposed LI maps the performance of subjects into several different plateaus. Both the first and the second criteria of the LI above contribute to the drifts. These drifts, therefore, help us to distinguish several different types of trading performances, and the one which concerns us most is the highest plateau, namely, the LI from 1,375 to 1,400.[1] Subjects who are able to obtain a score in this range are the ones who have already made the maximum profits, but did not minimize the number of bids. These additional bids (efforts) may be interpreted as a kind of *trembling* around the optimal value. This trembling may signify that the subjects have already learned the best strategy as long as they repeatedly behave in this way.

However, this is not a one-shot game. We, therefore, have to consider the case where the subject has learned the best strategy, but his LI may occasionally fall down to the other plateau (see Figure 3 for an illustration). These falls may occur for the following reasons. First, the subject was tired and made operational mistakes. Second, the subject did not know that he had already found the optimal strategy and attempted to explore further before realizing that nothing was there. Falls of these kinds can then be tolerated as long as they do not occur frequently. *Hence, a subject is considered to have learned the best strategy if he can stay on the highest plateau long enough to make any fall look like an accident.*



Period Criteria	-6	-5	-4	-3	-2	-1	Current period
LI = 1400					▨	▨	▨
1400 > LI ≧ 1395				▨	▨	▨	▨
1400 > LI ≧ 1390			▨	▨	▨	▨	▨
1400 > LI ≧ 1380		▨	▨	▨	▨	▨	▨
1400 > LI ≧ 1375	▨	▨	▨	▨	▨	▨	▨

Fig. 4. Accident-tolerance Criteria for Deciding Whether the Subject Has Learned

The discussion above motivates the development of the accident-tolerance criteria. One example is the one given in Figure 4. Depending on the score that the subject has, we can accept different frequencies of fall. As suggested in the figure, it is sufficient to consider that the subject has learned the best strategy if he had the highest score (1,400) twice over the last three periods. In other words, if he has been

[1] Our design of the LI by following the three criteria does not allow any possible score to lie between 1,300 and 1,375. In other words, the plateau next to this highest one starts from 1,300 and below.

really good on two occasions, then missing once is accepted as an accident. In a similar vein, we also consider a subject to have learned if his scores are between 1,395 and 1,400 three times over the last four periods, or between 1,390 and 1,400 four times over the last five periods, and so on and so forth, or if his scores are between 1,375 and 1400 six times over the last seven periods. In sum, the higher the degree of the trembling, the longer that the stay in the plateau is required for the subject to be considered to have learned.

4 Learning Performance and Working Memory

The learning index (Section 3.2) and the accident-tolerance criteria (Section 3.3) are now applied to the 165 subjects. The results are shown in Figure 5. To avoid redundancies, we only show those subjects who have learned, at least once, in the sense of the accident-tolerance criteria (Figure 4). In other words, one of the five possibilities must apply for the subject at least once during the 30-period experiment; if that never happens, the subject simply did not learn the best strategy and is not shown here. In this way, the learning dynamics of 29 subjects are presented in Figure 5. The grayed cell means that one of the accident-tolerance criteria applies for the respective agent in the respective period. Close to 20% of the 165 subjects had been able to visit the best strategy. Some were able to do so in the very beginning, like subject #1165; some needed a longer time to do so, like subject #1384.

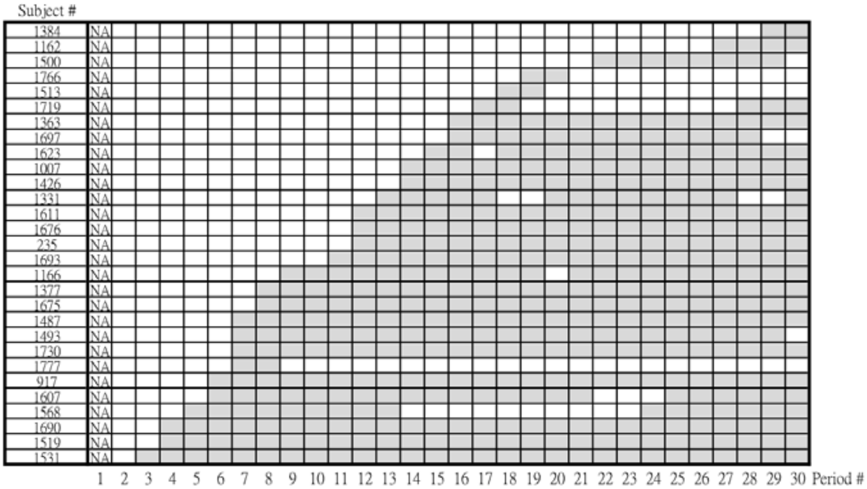


Fig. 5. Learning Performance

We further study the cognitive capacity of the subjects who learned and those who did not. The cognitive capacity of the subject is measured by a version of the working memory (WM) test [12]. Subjects are clustered into the groups: the performing group who learned (29 subjects) and the non-performing group who did not learn (136 subjects). It is found that the mean WM test score is 0.28036 for the performing

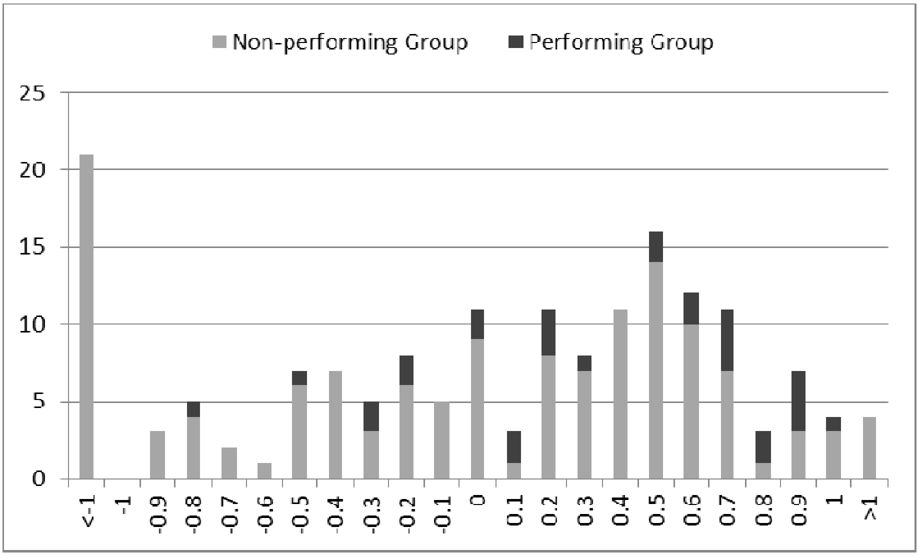


Fig. 6. Score Distribution of the Working Memory Test

group, but only -0.12648 for the non-performing group; the population average is -0.05004. This finding with regard to the significance of cognitive capacity is consistent with the one of Chen et al. [8]. Figure 6 gives the distribution of the WM Test Scores for the 165 subjects.

5 Conclusions

While the human-competitiveness of GP has been shown in many recent studies [10], in the context of double auction market experiments, we found that a small group of human subjects can perform better than GP. Specifically, this group of human subjects with a working memory capacity higher than average can successfully learn to use the optimal procrastination strategy to trade, while GP rarely can find this solution. Our analysis of the learning behavior of human subjects, including the proposed LI and the accident-tolerance criteria, is tailor-made for humans. We hope that the approach proposed in this paper can advance a more delicate analysis of the learning perplexity of human behavior observed in the human-subject laboratory.

One thing which deserves a more careful treatment is a further look at the patterns within the performing group and those within the non-performing group, both from a dynamic viewpoint, since our proposed LI and accident-tolerance criteria may reveal many interesting hidden learning dynamics which are not easy to catch by eye-browsing. In addition, to see the usefulness of our proposed tailor-made approach for human subjects, it is necessary to apply it to more human subject experiments. This will be the focus of our next study.

Acknowledgments. The authors are grateful to two anonymous referees for their painstaking reviews. The paper has been rewritten substantially in light of their very constructive suggestions. NSC research grants no. 98-2410-H-004-045-MY3 and no. 99-2811-H-004-014 are gratefully acknowledged.

References

1. Smith, V.L.: An Experimental Study of Competitive Market Behavior. Journal of Political Economy 70(2), 111–137 (1962)
2. Furuhata, M., Perrussel, L., Thévenin, J.-M., Zhang, D.: Experimental Market Mechanism Design for Double Auction. In: Nicholson, A., Li, X. (eds.) AI 2009. LNCS, vol. 5866, pp. 1–10. Springer, Heidelberg (2009)
3. Gjerstad, S.: The Strategic Impact of Pace in Double Auction Bargaining. In: IEDAS, `http://129.3.20.41/eps/mic/papers/0304/0304001.pdf`
4. Deshmukh, K., Goldberg, A.V., Hartline, J.D., Karlin, A.R.: Truthful and Competitive Double Auctions. In: Möhring, R.H., Raman, R. (eds.) ESA 2002. LNCS, vol. European Symposium on Algorithms, p. 361. Springer, Heidelberg (2002)
5. Rust, J., Miller, J., Palmer, R.: Behavior of Trading Automata in a Computerized Double Auction Market. In: Friedman, D., Rust, J. (eds.) Double Auction Markets: Theory, Institutions, and Laboratory Evidence. Addison Wesley, Redwood City (1993)
6. Rust, J., Miller, J., Palmer, R.: Characterizing Effective Trading Strategies: Insights from a Computerized Double Auction Tournament. Journal of Economic Dynamics and Control 18, 61–96 (1994)
7. Ariely, D., Norton, M.I.: Psychology and Experimental Economics: A Gap in Abstraction. Current Directions in Psychological Science 16(6), 336–339 (2007)
8. Chen, S.H., Tai, C.C., Yang, L.X., Shih, K.C.: The Significance of Working Memory Capacity in Double Auction Markets: Modeling, Simulation and Experiments. In: The 2011 Allied Social Sciences Association Meetings, January 6-9. Denver, Colorado (2011)
9. Chen, S.H., Yu, T.: Agents Learned, but Do We? Knowledge Discovery Using the Agent-based Double Auction Markets. Front. Electr. Electron. Eng. 6(1), 159–170 (2011)
10. Koza, J.R., Keane, A.M., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G.: Genetic Programming IV: Routine Human-Competitive Machine Intelligence. Kluwer Academic Publishers, Dordrecht (2003)
11. Xia, M., Stallaert, J., Whinston, A.B.: Solving the Combinatorial Double Auction Problem. European Journal of Operational Research 164, 239–251 (2005)
12. Lewandowsky, S., Oberauer, K., Yang, L.-X., Ecker, U.K.H.: A Working Memory Test Battery for MatLab. Behavior Research Methods 42(2), 571–585 (2011)

Portfolio Optimization Using SPEA2 with Resampling

Sandra García, David Quintana, Inés M. Galván, and Pedro Isasi

Computer Science Department, Carlos III University of Madrid
Avda. Universidad 30, 28911 Leganes, Spain
http://www.evannai.inf.uc3m.es

Abstract. The subject of financial portfolio optimization under real-world constraints is a difficult problem that can be tackled using multiobjective evolutionary algorithms. One of the most problematic issues is the dependence of the results on the estimates for a set of parameters, that is, the robustness of solutions. These estimates are often inaccurate and this may result on solutions that, in theory, offered an appropriate risk/return balance and, in practice, resulted being very poor. In this paper we suggest that using a resampling mechanism may filter out the most unstable. We test this idea on real data using SPEA2 as optimization algorithm and the results show that the use of resampling increases significantly the reliability of the resulting portfolios.

1 Introduction

The problem of choosing the right combination of financial assets has been the subject of research for a long time and it is one of the most active research lines in finance. This is often framed as a multiobjective optimization problem where the investor tries to find the right set of portfolios with the best risk/return profiles.

A large portion of the academic literature on this subject builds on the seminal work by Markowitz [5,6]. The approach suggested by this author works under some assumptions that allow the problem to be tackled with quadratic programming. Unfortunately, these assumptions do not hold in the real-world, which calls for alternatives. That is the reason why the framework of evolutionary multiobjective optimization is getting traction on this area [1,2,13]. In all these references, the authors use evolutionary multiobjective algorithms to evolve sets of solutions that minimize risk and maximize return. The first one, introduces a customized hybrid version NSGA-II and the last two compare the performance of different multiobjective algorithms.

One of the most important factors that asset managers face when they have to asses the results provided by any of the above-mentioned methods is stability. Very often, the expected efficient frontier lies far from the actual one as the forecasted risk/return profile of the portfolios is not accurate. This problem is one the major reasons why some practitioners mistrust the mentioned approach and the search for solutions has cleared the way for the field of robust portfolio optimization. The main contribution of this paper is the introduction of a new resampling mechanism that reduces the risk mentioned.

When we forecast the risk and return of a specific portfolio, we rely on estimates for the expected returns of individual assets and the variance-covariance matrix. The

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 127–134, 2011.

forecasts, which are likely to be inaccurate, are usually based on past data which may not be representative to predict the future due to, for instance, the presence of outliers.

In this scenario, there are several potential ways to approach the problem. The main two are either putting an emphasis on having robust estimates for the above mentioned parameters [7] or implementing a system that deals with uncertainty in the estimation process [8,9]. The approach suggested in this paper is a new technique that falls in the latter category. We will use an evolutionary multiobjective algorithm enhanced with a resampling mechanism that changes the parameters of the fitness function during the evolution process with the aim to obtain robust solutions. We consider that using a multiobjective genetic algorithm that exposes candidate solutions to different scenarios will improve the reliability of the resulting Pareto front. The use of resampling in the context of portfolio optimization is not new [11,12]. The most comparable approach is described by Idzorek [4] who suggests using combining traditional quadratic programming (QP) with Monte Carlo simulation to derive a set of fronts that are subsequently merged into a single solution. We understand that resampling within the context of multiobjective evolutionary algorithms is a better strategy as it would allow real-world constraints intractable by QP while, at the same time, approximating the efficient frontier in single run.

The rationale for the approach is that optimizing for a single scenario bears the risk of getting solutions that are hyper-specialized and might be extremely sensitive to deviations in the parameters used in the fitness evaluation. Given that it is almost certain that we will not be able to predict accurately the behaviour of all the assets, we could consider the alternative of targeting portfolios that offer appropriate risk/return tradeoffs under different scenarios. Even if the solutions do not seem to be as good as the specialized ones under the expected scenario, they might be more reliable in practice. Once we replace the expected parameters of the models with the observed ones, the actual risk and returns might be more likely to be closer to the expected values, hence providing more value to the decision maker.

The choice of performance metrics is a very important issue in multiobjective optimization [15]. It is generally admitted that there is no single metric that can be used to evaluate different objectives simultaneously. In addition to that, we also face the lack of standard robustness metrics that we could use in this context. For this reason, we introduce a new one that accounts for the average difference on the objective space using the estimates for the parameters vs. using their real values, across all the solutions.

The rest of the paper is organized as follows. First, we make a formal introduction to the portfolio optimization problem. Then, we describe in detail our approach and the proposed metric to evaluate the robustness of the solutions. That will be followed by the experimental results and a section devoted to summary and conclusions.

2 Portfolio Optimization's Problem Definition

The Modern Portfolio Theory was originated in the article published by Harry M. Markowitz, in 1952 [5]. In general, the portfolio optimization problem is the choice of an optimum set of assets to include in the portfolio and the distribution of investor's wealth among them. Markowitz [6,10] assumed that solving the problem requires the simultaneous satisfaction of maximizing the expected portfolio return $E(R_p)$ and minimizing the portfolio risk (variance) σ_p^2. This can be formally defined as:

- Minimize the **risk** (variance) of the portfolio:

$$\sigma_p^2 = \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} w_i w_j \sigma_{ij} \tag{1}$$

- Maximize the **return** of the portfolio:

$$E(R_p) = \Sigma_{i=1}^{n} w_i \mu_i \tag{2}$$

- Subject to:

$$\Sigma_{i=1}^{n} w_i = 1 \text{ and } 0 \leq w_i \leq 1; i = 1...n \tag{3}$$

where n is the number of available assets, μ_i the expected return of the asset i, σ_{ij} the covariance between asset i and j, and w_i are the weight giving the composition of the portfolio. The constraints referenced in equations 3 require the full investment of funds and prevents the investor from shorting any asset.

3 Parameter Resampling Strategy

In this work, the problem of obtaining stable and robust portfolios is tackled using an evolutionary multiobjective algorithm, specifically, SPEA2 [14]. This choice is justified by the results reported in [1] and [1,13]. Both compare several algorithms in the domain and conclude that SPEA2 offers the best results.

The encoding chosen will represent each portfolio as a vector of real numbers, this means that SPEA2 will work with real elements instead of binary ones. Each of these numbers represents the percentage of investment per asset (also called weight: w_i where $i = 1...n$ and n is the number of investable assets). Here, each portfolio will be represented by a single element of the population. Every individual must meet the constraints from eq. 3 and therefore, will be be repaired after applying the genetic operators.

As we mentioned, the main contribution of this paper is the way that we handle the fitness function. The starting point for the evaluation of portfolios is the framework introduced in section 2, where the fitness of each individual is determined by evaluating the two objective functions: return $E(R_p)$ and risk σ_p.

As it is apparent from equations 1 and 2, these functions are very dependent on the values of the expected asset returns and the variance-covariance matrix. One of the most important challenges in portfolio context is the dependence of the solutions on the estimates for these parameters. Given the difficulty inherent to financial forecasting, it is unlikely that these parameters are accurate. This lack of accuracy is likely to result in a set of portfolios that end up behaving in an unexpected way. Fig. 1 shows a real example of one Pareto front where the solutions are evaluated using the forecasted parameters (in red) and the real parameters (in green). For this reason, we suggest altering the fitness function in a way that it discards those portfolios that, under normal circumstances, could potentially show bad performance.

The basic idea behind the solution that we suggest is to keep changing the parameters of the fitness functions during the evolution process. If these parameters take values that

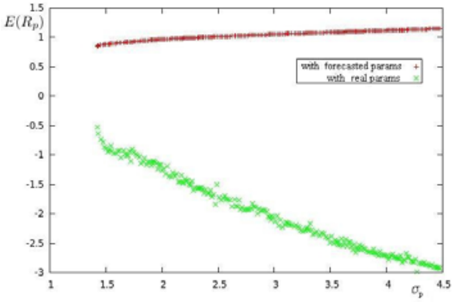


Fig. 1. Solution evaluated with forecasted and real parameters

are likely to be close to the real ones, evolution would tend to favour those individuals that tend to have a good performance in all circumstances (they will not be necessarily the best for the real value of the parameters, that will only be known a posteriori). Those solutions that tend to be good most of the time but are particularly bad in some scenarios may have to face them and therefore, would be weeded out of the population with high probability due to their low fitness. A key factor for the success of this strategy is the mechanism used to generate the scenarios.

The approach that we use to generate these scenarios is non-parametric bootstrap. Our algorithm used a time window to resample data, instead of using all the data to derive a single estimate for the parameters. The resampling process selects a random set of time periods that has the same size as the original window (each period might be selected more that once). Then, we average the returns for those time periods and compute the variance-covariance matrix. Whenever we do this, we generate new estimates for the parameters that are based on past data. These estimates can subsequently be used to calculate the risk and return of the portfolios. The process generates a new sample set S' with size N_s' from the original sample S with size N_s. While $N_s' \neq N_s$ one instance X_i is selected randomly from S and added to the new set $S' = S' + X_i$. At the beginning, $S' = \varnothing$ and $N_s' = 0$.

Hence, when we use resampling we create a set of ”likely scenarios” that prevents the solution form hyper-specializing in just one, making the solution more robust.

4 Evaluation Metric

In the context studied in this work, the metrics most commonly used, as Hypervolume (HV) and Spread, can not be used to measured the quality of solutions from a robustness point of view. HV is an indicator of the size of the dominated space and measured the volume enclosed by the union of the points in the front; the bigger value it gives, the better HV it has. The Spread is a diversity indicator which measures the extent of spread achieved among the obtained solutions so, the lower value, the more evenly distributed is the front.

In order to measure the robustness of the solutions, we have defined a new metric, named **Estimation Error (EE)**. The aim is to evaluate the average difference between the expected risk an return for every portfolio in the efficient frontier and the actual risk and return a posteriori once the real value of the parameters is observed. That is, the difference between the estimates for t_n based on data from t_1 to t_{n-1}, and the actual values at t_n. This metric is calculated measuring the average Mahalanobis distance (d_M) between the forecasted pair $(E(R_p), \sigma_p^2)$ for each portfolio, named $\overline{x_i}$, and the $(E(R_p), \sigma_p^2)$ for the same portfolio computed with the real parameters, named $\overline{x_i}'$ (the observed in t_n). The lower the value is, the higher is the reliability of the original front. Formally, it can be expressed as follows:

$$EE = \Sigma_{i=1}^{p} \frac{[d_M(\overline{x_i}, \overline{x_i}')]^2}{p} \tag{4}$$

where p is the number of portfolios in the Pareto front.

5 Experimental Results

In this section we test the aforementioned approach on a specific asset allocation problem. The sample consists of 320 monthly returns for eight broad financial indexes representing that many asset classes. The series of monthly returns cover from May, 1980 to December, 2006 and the source is *DataStream*. The list of indexes used in this work is the following: Frank Russell 2000 Growth (FRUS2GR), Frank Russell 1000 Value (FRUS1VA), Frank Russell 1000 Growth (FRUS1GR), S&P GSCI Commodity Total Return (GSCITOT), MSCI EAFE (MSEAFEL), BOFA ML CORP MSTR ($)(MLCORPM) and, BOFA ML US TRSY /AGCY MSTRAAA($)(MLUSALM).

As we have mentioned, SPEA2 has been used for experiments. The code was developed in Java under jMetal [3], that includes SPEA2. Simulated Binary Crossover and Polynomial Mutation are used as crossover and mutation operators. Probabilities are respectively fixed to 0.125 and 0.9.

The SPEA2+resampling strategy is compared to SPEA2. For both strategies, resampling and no-resampling, experiments varying the population size (50, 100 and, 200) and the number of evaluations (5000, 10000 and 20000 respectively) have been carried out. Experiments with and without resampling are, respectively, called as R-50, R-100, R-200, and NR-50, NR-100, NR-200 where the numbers 50, 100 and 200 denote the population size.

A sliding window must be used when fitness function is evaluated using the resampling technique. Each window has a size n of 200 monthly returns. The sets t_1 to t_{n-1} are used in the algorithm to get the final solutions and the period t_n is used to evaluate them. In our experimentation the window is moved 120 times (one month at a time) taking the interval from 30/05/1980 to 29/12/2006. The algorithm is run 30 times per window getting, for each window, a 30 solution set.

We report in table 1 the average results for the metrics referenced in 4. It shows the average, median, variance, maximum and minimum values of the metrics to compare experiments with and without resampling. To calculate HV and Spread, the extremes of optimal front are approximated with the limit points from the whole set of the experiments.

Table 1. Metrics values

EE	R-50	NR-50	R-100	NR-100	R-200	NR-200
Average	3.58389	3.62739	3.58586	3.75097	3.61118	3.76668
Median	3.75999	3.84983	3.77780	3.92876	3.80308	3.94310
Variance	0.18572	0.22016	0.19541	0.16160	0.19780	0.15766
Maximum value	3.94303	3.95819	3.96495	3.97959	3.97590	3.98965
Minimum Value	1.83713	1.84397	1.80752	2.03810	1.82528	2.07942
HV	R-50	NR-50	R-100	NR-100	R-200	NR-200
Average	0.51551	0.50941	0.47368	0.50176	0.47223	0.49715
Median	0.53966	0.45816	0.42839	0.45258	0.42656	0.44895
Variance	0.01767	0.01728	0.01700	0.01645	0.01679	0.01616
Maximum value	0.77764	0.78560	0.74998	0.76184	0.74420	0.75424
Minimum Value	0.22626	0.30859	0.24046	0.30668	0.23544	0.30320
SPREAD	R-50	NR-50	R-100	NR-100	R-200	NR-200
Average	0.89554	0.80186	0.88501	0.47855	0.88513	0.46028
Median	0.89228	0.79946	0.88210	0.49075	0.87906	0.47166
Variance	0.00715	0.01020	0.00997	0.00909	0.01034	0.00951
Maximum value	1.22227	1.11792	1.33979	0.64023	1.43464	0.61230
Minimum Value	0.57814	0.49306	0.59058	0.25040	0.52798	0.22788

We notice that all the experiments with resampling present smaller EE averages and variances. This means that resampled solutions behave as expected to a greater degree than non-resampled ones. In terms of HV, the averages for ”NR-100” and ”NR-200” are a bit larger that the ones observed for ”R-100” y ”R-200” but the HV of ”R-50” and ”NR-50” is similar. The average Spread for the SPEA2+resampling is larger than the equivalent metric for non-resampled solutions, which means that the latter tends to have a better distribution. We also see that EE grows with the population size. Regardless of this size, the resampling method always provides lower EE and more Spread that the basic one and these differences are significant at 1% using Wilcoxon test. This is due to the emphasis of resampling technique on the selection of stable individuals, which might be found in specific sections of the front, while the basic approach is not limited by this factor. The same test shows that the HV of ”NR-100” and ”NR-200” are higher, with a significance difference of 1%, than the ones of ”R-100” and ”R-200”, however this difference does not exist between ”R-50” and ”NR-50”. So, even if the resampled solutions do not seem to be as good as the others in terms of HV and spread, the EE metric show that they might be more robust and reliable in practice providing more value to the decision maker.

In order to illustrate the shape of the fronts, we show two fronts example in fig. 2, one from ”R-50” (in green) and the other one from ”NR-50” (in red).

The results presented above have shown that resampling always improves significantly the reliability of the solutions as measured by average error committed predicting the performance of the final set of portfolios. However, this robustness comes at a cost as non-resampled portfolios tend to be distributed more evenly along the front and for some cases the hypervolume tends to be higher.

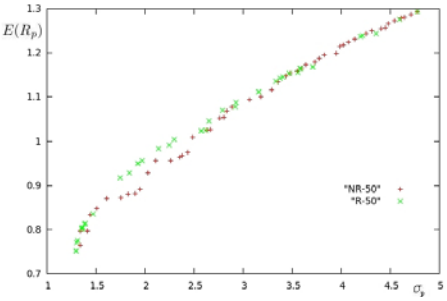


Fig. 2. Comparison between resampled and non-resampled fronts

6 Conclusions

Portfolio optimization is one of the most active research lines in finance and the choice of the right combination of financial assets can be framed as a multiobjective optimization problem.

Portfolio managers often face the problem that the expected efficient frontier derived from their forecasts for future returns is subject to uncertainty. This means that if the real parameters differ from the forecasted ones, the risk and returns the portfolios included in the estimated efficient frontier might deviate substantially from the predictions. This could result extreme underperformance of some portfolios. This uncertainty is one of the major reasons why some practitioners mistrust quantitative methods based on modern portfolio theory, and the search for solutions has cleared the way for the development of robust portfolio optimization.

We presented a new resampling mechanism based on non-parametric bootstrap that, used in combination with MOEAs such as SPEA2 should lower the likelihood of getting unstable or not robust solutions. The mechanism resamples past data to generate different scenarios that are used during the evolutionary process to evaluate the portfolios in the population. Those portfolios that perform very poorly in any of these scenarios get discarded by the algorithm. This results in a final front consists of individuals that tend to offer good performance in any circumstance.

The approach was tested on real data on a sample of monthly returns for eight indexes representing different broad investment categories including stock, bonds, etc. We compared the performance of SPEA2 vs SPEA2+Resampling. The results show that resampling enhances significantly the reliability of the solutions as measured by average error committed predicting the performance of the final set of portfolios. This is very important as it indicates decision makers could rely more in the robust front to pick the right portfolio according to their preferences.

Even though these results are encouraging, there are several issues left open that could lead to future extensions of this work. Among them, the performance of the resampling mechanism using other MOEAs, the scalability of the results with the size of the number of investment alternatives, or the performance of the approach once the decision maker faces cardinality restrictions to the amount he can invest in a specific asset class.

Acknowledgements. The authors acknowledge financial support granted by the Spanish Ministry of Science under contract TIN2008-06491-C04-03 (MSTAR) and Comunidad de Madrid (CCG10-UC3M/TIC-5029).

References

1. Anagnostopoulos, K.P., Mamanis, G.: The mean-variance cardinality constrained portfolio optimization problem: An experimental evaluation of five multiobjective evolutionary algorithms. Expert Systems with Applications (2011) (in press, corrected proof)
2. Deb, K., Steuer, R.E., Tewari, R., Tewari, R.: Bi-objective portfolio optimization using a customized hybrid NSGA-II procedure. In: Takahashi, R.H.C., Deb, K., Wanner, E.F., Greco, S. (eds.) EMO 2011. LNCS, vol. 6576, pp. 358–373. Springer, Heidelberg (2011)
3. Durillo, J.J., Nebro, A.J., Alba, E.: The jmetal framework for multi-objective optimization: Design and architecture. In: CEC 2010, Barcelona, Spain, pp. 4138–4325 (July 2010)
4. Idzorek, T.M.: Developing robust asset allocations (2006)
5. Markowitz, H.M.: Portfolio selection. The Journal of Finance 7(1), 77–91 (1952)
6. Markowitz, H.M.: Portfolio Selection: efficient diversification of investments. John Wiley & Son, Chichester (1959)
7. Perret-Gentil, C., Victoria-Feser, M.-P.: Robust mean-variance portfolio selection. Fame research paper series. International Center for Financial Asset Management and Engineering (April 2005)
8. Pflug, G., Wozabal, D.: Ambiguity in portfolio selection. Quantitative Finance 7(4), 435–442 (2007)
9. Tütüncü Reha, H., Koenig, M.: Robust asset allocation. Annals OR 132(1-4), 157–187 (2004)
10. Reilly, F.K., Brown, C.K.: Investment Analysis and Portfolio Management. The Dryden Press (1997)
11. Ruppert, D.: Statistics and finance: An introduction. Journal of the American Statistical Association 101, 849–850 (2006)
12. Shiraishi, H.: Resampling procedure to construct value at risk efficient portfolios for stationary returns of assets (2008)
13. Skolpadungket, P., Dahal, K., Harnpornchai, N.: Portfolio optimization using multi-objective genetic algorithms. In: Proceeding of 2007 IEEE Congress on Evolutionary Computation, pp. 516–523 (2007)
14. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength pareto evolutionary algorithm. Technical report, Computer Engineering and Networks Laboratory (TIK). Swiss Federal Institute of Technology (ETH), Zurich, Switzerland (2001)
15. Zitzler, E., Thiele, L.: An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report 43, Gloriastrasse 35, CH-8092 Zurich, Switzerland (1998)

Decoding Phase-Based Information from Steady-State Visual Evoked Potentials with Use of Complex-Valued Neural Network

Nikolay V. Manyakov, Nikolay Chumerin, Adrien Combaz, Arne Robben, Marijn van Vliet, and Marc M. Van Hulle

Laboratory for Neuro- and Psychofysiology, K.U.Leuven,
Herestraat 49, bus 1021, 3000 Leuven, Belgium
{NikolayV.Manyakov,Nikolay.Chumerin,Adrien.Combaz,Arne.Robben,
Marijn.vanVliet,Marc.VanHulle}@med.kuleuven.be

Abstract. In this paper, we report on the decoding of phase-based information from steady-state visual evoked potential (SSVEP) recordings by means of a multilayer feedforward neural network based on multi-valued neurons. Networks of this kind have inputs and outputs which are well fitted for the considered task. The dependency of the decoding accuracy w.r.t. the number of targets and the decoding window size is discussed. Comparing existing phase-based SSVEP decoding methods with the proposed approach, we show that the latter performs better for the larger amount of target classes and the sufficient size of decoding window. The necessity of the proper frequency selection for each subject is discussed.

1 Introduction

Research on brain-computer interfaces[1] (BCIs) has witnessed a tremendous development in recent years (see, for example, the editorial in IEEE Signal Processing Magazine [1]), and is now widely considered as a successful application of the neurosciences. BCIs can significantly improve the quality of life of patients suffering from amyotrophic lateral sclerosis, stroke, brain/spinal cord injury, cerebral palsy, muscular dystrophy, etc. Brain-computer interfaces are either invasive [2,3,4] or noninvasive [5,6]. The invasive ones use recordings made intracortically (local field potentials and action potentials) or from the surface of the brain (electrocorticogram), whereas the noninvasive ones mostly employ electroencephalograms (EEGs) recorded from the subject's scalp.

Several noninvasive methods have been proposed in the literature. The one we consider in this paper, is based on the steady-state visual evoked potential (SSVEP). This type of BCI relies on the psychophysiological properties of EEG

[1] A BCI is a device, which records and interprets brain activity automatically, allowing the subject to interact with the world through computers, robot actuators and so on, bypassing the need for muscular activity.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 135–143, 2011.

brain responses recorded from the occipital area during the periodic presentation of visual stimuli (flickering stimuli). When the periodic presentation is at a sufficiently high rate (> 6 Hz), the individual transient visual responses (which are time and phase locked to the stimulus onsets) overlap and become a steady state signal: the signal resonates at the stimulus rate and its multipliers [7].

Conventional SSVEP-based BCI systems [8,9,10,11] use an increase in amplitude of frequencies f, $2f$, $3f, \ldots$ in the power spectral density of the EEG data to detect that the subject is looking at target flickering at rate f. Since the relevant EEG activity is always embedded into other on-going brain activity and contaminated by (recording) noise, the detection task is not straight-forward. Considering a small recording interval it is quite likely to detect an (irrelevant) increase in the amplitude at frequency f. To overcome this problem and improve the decoding performance, several methods are used: averaging over several time intervals [8], recording over longer period of time [9], preliminary training [10], etc. Finally, to enhance the BCI functionality, several stimulation frequencies $f_1, \ldots, f_N$ are used simultaneously at the same time, instead of only one f. In this case each frequency corresponds to a particular command with which the BCI can communicate. The detection problem, therefore, becomes more complex since now, the only one of several possible stimulation frequencies f_i needs to be detected from the EEG recordings.

While these methods achieve good information transfer rate [11], their application using a computer screen (monitor) for visual stimulation has some limitations: the computer screen based stimulation is restricted by the refresh rate of the screen [12] (85 Hz in our case); only stimulations from some particular (and subject dependent) frequency interval produce good responses [10]; the harmonics of some stimulation frequencies could coincide with the other frequencies (and their harmonics) deteriorating the decoding performance [12]. These restrictions limit the number of target commands in SSVEP-based BCI. To be able to use more targets the differentiation between phases in SSVEP stimulation has been proposed in [13,14]: even a single frequency could be used to encode commands employing the phase lag. For example, one can perform visual stimulation using N targets simultaneously flickering with the same frequency f, but with different time delays $\Delta t_m = (m-1)/(fN)$ seconds for the command m ($m = 1, \ldots, N$). Thus, extracting phase information from a Fourier transform of the EEG signal and comparing it to the phase of some reference signal(s) (for example, to the phase of EEG response for stimulus with zero phase lag [14] or to the phases of EEG responses for all possible delayed stimulations [13]), one can detect the target the subject is looking at. Such an approach allows to increase number of possible commands by combining different frequencies with different phase shifts [13] or simply using one (well detectable) frequency with different phase shifts [14]. Here we account the problem of accurately detecting phase shifts using only short recording intervals, and properly assigning class labels to the estimated phase information. In this paper, we investigate the possibility of applying a multilayer neural network based on multi-valued neurons (MLMVN) [15] for decoding up to 16 phase shifted targets (vs. 4–6 in [13] and 8 in [14]), which uses information from any amount of harmonics (vs. one in [14]) and channels (vs. Oz in [14] and optimal or bipolar (Oz-POz) in [13]), and allow for a good approximation of circular data as we have in our case – the phase shifts and resulting classes are circular.

2 Methods

2.1 EEG Experiment

The EEG recordings have been performed with a prototype of a miniature wireless EEG system developed by *imec*[2] and built around their ultra-low power 8-channel EEG amplifier chip [16]. Each of the eight channels is recorded with sampling rate $f_s^{\mathrm{EEG}} = 1000$ Hz at resolution of 12 bits/sample. We have used an electrode cap with large filling holes and sockets for active Ag/AgCl electrodes (ActiCap, Brain Products). The eight electrodes were placed primarily on the occipital pole, namely at positions Oz, O1, O2, POz, PO7, PO3, PO4, PO8, according to the international 10–20 system. The reference and ground electrodes were placed on the left and right mastoids, respectively.

Four healthy subjects (all male, aged 24–34 with average age 29.5, three righthanded, one lefthanded) participated in the experiments. The experiment consisted of observing a flashing stimulus in the center of the screen (CRT monitor, refresh rate $f_s^{\mathrm{scr}} = 85$ Hz). To produce stronger SSVEP responses [12], we have used the (monitor) frame-based visual stimulation of period of six frames: three frames of intense (white) stimulus presentation followed by three frames without the stimulus (black). The stimulation frequency is, thus, $f = 85/6 \approx 14.16$ Hz close to 15 Hz, which is reported to elicit the largest SSVEP amplitude [7]. The EEG data have been collected during sessions of two minutes long visual stimulation.

2.2 Multilayer Feedforward Neural Network Based on Multi-valued Neurons

Networks based on complex-valued neurons were reported to learn faster and generalize better than traditional neural networks in different benchmarks and real world problems [15,17,18]. As was mentioned in [18], the use of complex-valued inputs/outputs, weights and activation functions make it possible to increase the functionality of a single neuron and of a neural network, to improve their performance and to reduce the training time.

We have used a multilayer feedforward neural network based on multi-valued neurons (MLMVN) [15,17]. Such a network incorporates derivative-free backpropagation training algorithms, resulting in fast convergence to the minimum of an error function [15]. In MLMVN, each neuron from every hidden or output layers has connections to all neurons from the previous layer and has a complex activation function $P(z) = z/|z|$, where $z = w_0 + w_1x_1 + ... + w_nx_n$, $x_i \in \mathbb{C}$ is

[2] http://www.imec.be

the output of the i-th neuron from previous layer (with n neurons in this layer) and $w_i \in \mathbb{C}$ is the corresponding weight.

For the simulations we have used a MLMVN with 16 inputs, one hidden layer consisting of 24 neurons, and one neuron in the output layer. The network inputs are computed using phase information of the stimulation frequency and its first harmonic, estimated from all eight electrodes. Considering the EEG data window $\mathbf{EEG}(t) = (\mathrm{EEG}_1(t), \ldots, \mathrm{EEG}_8(t))$ of length T seconds ($t \in [0, T]$), the phase $\phi_{k,f}$ of the frequency f from k-th channel EEG_k is estimated as:

$$\phi_{k,f} = \arg\left(\int_0^T \mathrm{EEG}_k(t)\cos(2\pi f t)dt + i\int_0^T \mathrm{EEG}_k(t)\sin(2\pi f t)dt\right). \tag{1}$$

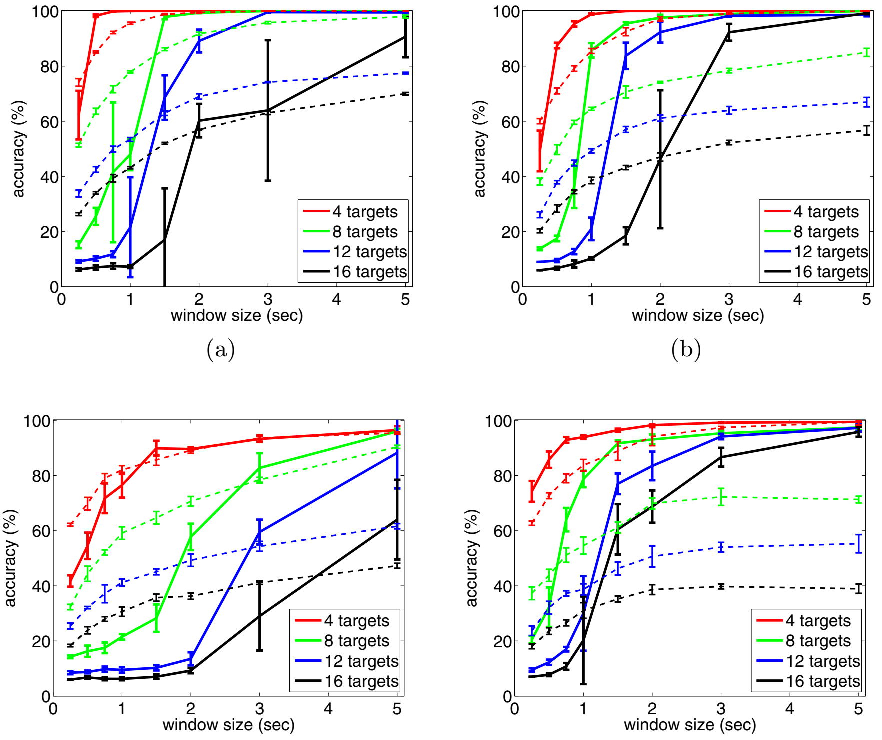
Since the phase information $\phi_{f,k} \in [0, 2\pi)$ (where 0 and 2π refer to the same phase), it can be represented by a complex number $e^{i\phi_{k,f}}$ on a unit circle on the complex plain. Estimated in such a way 16 (8×2) complex numbers constitute the input $\varPhi = (e^{i\phi_{1,f}}, \ldots, e^{i\phi_{8,f}}, e^{i\phi_{1,2f}}, \ldots, e^{i\phi_{8,2f}})$ to the MLMVN.

The output of the network $\mathcal{N}(\varPhi)$ is such index m ($m = 1, \ldots, N$, where N is the number of target classes) of the unit circle sector, that $2\pi(m-1)/N \leq \arg P(z) < 2\pi m/N$. Thus, after the network is trained, classification is obtained by simply looking for the target class correspondent to the obtained unit circle sector.

2.3 Training and Testing Data

To validate the applicability and the accuracy of the proposed classifier, it is sufficient to limit ourselves to off-line decoding. For this assessment, we opted for a 4-fold cross-validation results. Namely, the recorded data have been split into four parts. One of those parts has been used for training and the rest for testing. All possible selections of data for training were done, leading to four decoding results. Since we recorded data from one flickering stimuli having no phase shifts, we introduced phase-shifted targets (stimuli) artificially in the next way. Considering that under the normal conditions individual SSVEP latency is stable [13,19], the delay in the stimulation (using the same frequency) would introduce the same delay in the recorded EEG data. In this way, using proper shifts in the EEG data, the EEG responses for different phase shifted stimulations (with fixed frequency f) are constructed. From those constructed EEGs, we cut a number of intervals with length T second in such a way, that starting points of those intervals are aligned to the stimulus onsets. As a consequence, the interval starting points are distanced between each other on integer number of periods (with the length $1/f$ s). We have used this strategy to generate the training and test data. While in the training set, we allowed the constructed data to be highly overlapped (to have bigger amount of training samples), we tried to reduce those overlaps in the test set[3].

[3] As we found later on, such introduced sparsity does not much influence the decoding accuracy.



(a)

(b)

(c)

(d)

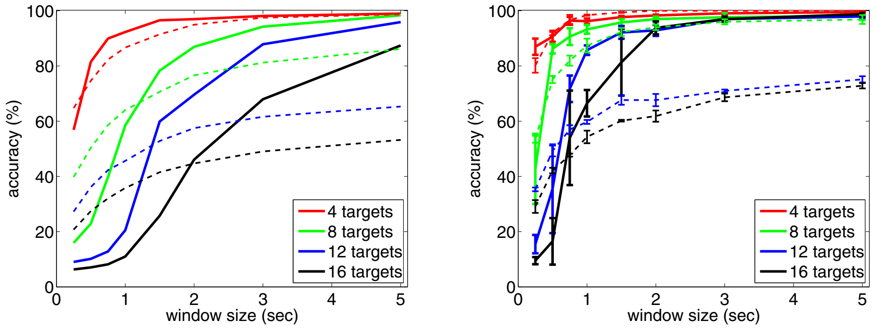
Fig. 1. Decoding results for 4 subjects based on the MLMVN (solid lines) and an extended version of the method from [13] (dashed lines). The results are shown for 4, 8, 12, and 16 targets (phase shifts). Stimulation frequency is $f = 85/6$ Hz. Error bars show standard deviations.

3 Results and Discussion

The results for four subject with stimulation $f = 85/6$ Hz and different number N of target classes are shown in Fig. 1, together with the averaged among all subject performance in Fig. 2a.

As it can be seen from those figures, MLMVN gives good classification performance for all subjects for a proper window length. But this accuracy strongly depends on the number of targets: good performance is achieved starting from $T = 1 - 1.5$ second window size for four targets, 2–3 second window for eight targets, above 3 second window for 12 targets and so on.

As it can be seen from Fig. 1c, one subject performance is quite low compared to the other subjects. This could be due to the fact, that this subject does not generate a proper SSVEP with the chosen frequency (85/6 Hz). To validate this, we made a second test with this subject, but with the lower frequency of



(a) (b)

Fig. 2. (a) Decoding results averaged among all subjects. Stimulation frequency is $f = 85/6$ Hz. (b) Decoding results for the same subjects as in Fig. 1c but for stimulation frequency $f = 85/10 = 8.5$ Hz. Results are given for 4, 8, 12, and 16 targets (phase shifts) and based on the MLMVN (solid lines) and extended for all electrodes method from [13] (dashed lines). Error bars show standard deviations.

8.5 Hz. As we can see from the result in Fig. 2b, this choice of stimulus frequency improves the resulting accuracy. This suggests that the stimulation frequency is subject dependent and should be selected during a calibration stage.

Based on the results as presented above, we suggest the use of MLMVN in an on-line BCI, since in addition to the good accuracy, the training time is also reasonable.

In order to justify the use of MLMVN for phase-based SSVEP classification, we made a comparison with the classifiers used in the literature for such types of BCIs. The comparison with algorithms used in [14] is not straightforward, since they use only one EEG channel and consider only the principal oscillation harmonic. The generalization into several EEG channels requires some additional logic in the algorithm. But here we can stress, that since our approach allows (and involves) the use of a large amount of relevant information for decoding, we highly probably can achieve better performance. In addition, with the use of MLMVN, we do not need to apply additional logic for incorporating circularity of the data, since this is per se included into the classifier.

For comparison with [13], we have generalized the algorithm from this paper by taking into account all channels (instead of restricting ourselves to only the best one). This generalization has been done by taking the summation of the obtained correspondent projected values from all channels (compare to [13]). We believe, that such summation will not give worse results than using only one channel, since SSVEP responses are presented in all channels we have. To make a valid comparison, we used the same training set (with 4-fold cross-validation) and the same number of harmonics for decoding. Fig. 1 and 2 show the corresponding results in dashed lines.

We notice that algorithms from [13] gives better results for small window sizes, but the obtained accuracy is not sufficient for reasonable use in BCI systems. For bigger window sizes MLMVN outperforms in the terms of accuracy. Inferior performance of MLMVN for small window sizes could be explained by the fact that here we do not have a clearly separable training set, which is tried to be fitted by such nonlinear classifier as MLMVN. Thus, while fitting badly separable training set, MLMVN is not able to make generalization (and we have a performance close to the chance level). While for more separable data (the case of bigger window size T), MLMVN is able to generalize data through much better nonlinear separation, then the classifier used in [13]. In addition, the comparison results suggest that for a small amount of target classes both classifiers produce almost equal results. And only starting from eight target classes MLMVN outperforms. The gain in accuracy for MLMVN becomes more prominent with the increase in number of classes.

For future steps in validation of MLMVN for use in a phase-based SSVEP BCI, different MLMVN topologies can be compared in order to increase the accuracy. Additionally to the phase information from each channel separately, we can also include features showing inter-channel relation, such as synchronization [20] and characteristics of propagating waves [21]. We should also consider the construction of an on-line BCI system and its evaluation on larger subject group. For these purposes, a search for a proper subset of electrodes is also highly advisable (without significant loss in accuracy), as it increases the user-friendliness of the system.

4 Conclusion

We have performed decoding of stimuli-target information, encoded through phase shifts in SSVEP EEG data. The use of multilayer feedforward neural network based on multi-valued neurons (MLMVN) has been validated for this task, showing its superior performance for the bigger amount of target classes and the sufficient window size used in the decoding procedure. All these recommend the use of our approach for an on-line brain-computer interface.

Acknowledgments. NVM is supported by the Flemish Regional Ministry of Education (Belgium) (GOA 10/019). NC is supported by the European Commission (IST-2007-217077). AC and AR are supported by a specialization grant from the Agentschap voor Innovatie door Wetenschap en Technologie (IWT, Flemish Agency for Innovation through Science and Technology). MvV is supported by G.0588.09. MMVH is supported by research grants received from the Program Financing program (PFV/10/008) and the CREA Financing program (CREA/07/027) of the K.U.Leuven, the Belgian Fund for Scientific Research - Flanders (G.0588.09), the Interuniversity Attraction Poles Programme – Belgian Science Policy (IUAP P6/054), the Flemish Regional Ministry of Education (Belgium) (GOA 10/019), and the European Commission (IST-2007-217077), and by the SWIFT prize of the King Baudouin Foundation of Belgium.

The authors are also grateful to Refet Firat Yazicioglu, Tom Torfs and Cris Van Hoof from the imec in Leuven for providing us the wireless EEG system.

References

1. Sajda, P., Müller, K.-R., Shenoy, K.V.: Brain-Computer Interfaces. IEEE Signal Proc. Magazine 25(1), 16–17 (2008)
2. Lebedev, M.A., Nicolelis, M.A.L.: Brain-Machine Interface: Past, Present and Future. Trends in Neurosc. 29(9), 536–546 (2005)
3. Manyakov, N.V., Van Hulle, M.M.: Decoding Grating Orientation from Microelectrode Array Recordings in Monkey Cortical Area V4. International Journal of Neural Systems 20(2), 95–108 (2010)
4. Velliste, M., Perel, S., Spalding, M.C., Whitford, A.S., Schwartz, A.B.: Cortical Control of a Prosthetic Arm for Self-Feeding. Nature 453, 1098–1101 (2008)
5. Birbaumer, N., Kübler, A., Ghanayim, N., Hinterberger, T., Perelmouter, J., Kaiser, J., Iversen, I., Kotchoubey, B., Neumann, N., Flor, H.: The Thought Translation Device (TTD) for Completely Paralyzed Patients. IEEE Transactions on Rehabilitation Egineering 8(2), 190–193 (2000)
6. Blankertz, B., Dornhege, G., Krauledat, M., Müller, K.-R., Curio, G.: The Non-Invasive Berlin Brain-Computer Interface: Fast Acquisition of Effective Performance in Untrained Subjects. Neuroimage 37(2), 539–550 (2007)
7. Herrmann, C.S.: Human EEG Responses to 1-100 Hz Flicker: Resonance Phenomena in Visual Cortex and Their Potential Correlation to Cognitive Phenomena. Exp. Brain Res. 137, 346–353 (2001)
8. Cheng, M., Gao, X., Gao, S., Xu, D.: Design and Implementation of a Brain-Computer Interface with High Transfer Rates. IEEE Transactions on Biomedical Engineering 49(10), 1181–1186 (2002)
9. Gao, Y., Wang, R., Gao, X., Hong, B., Gao, S.: A Practical VEP-based Brain-Computer Interface. IEEE Transactions on Neural Systems and Rehabilitation Engineering 14(2), 234–240 (2006)
10. Manyakov, N.V., Chumerin, N., Combaz, A., Robben, A., Van Hulle, M.M.: Decoding SSVEP Responses Using Time Domain Classification. In: Proc. of the International Conference on Fuzzy Computation and 2nd International Conference on Neural Computation, pp. 376–380 (2010)
11. Allison, B., Luth, T., Valbuena, D., Teymourian, A., Volosyak, I., Graser, A.: BCI Demographics: How Many (and What Kinds of) People Can Use an SSVEP BCI? IEEE Transactions on Neural Systems and Rehabilitation Engineering 18(2), 107–116 (2010)
12. Volosyak, I., Cecotti, H., Gräser, A.: Impact of Frequency Selection on LCD Screens for SSVEP Based Brain-Computer Interfaces. In: Cabestany, J., Sandoval, F., Prieto, A., Corchado, J.M. (eds.) IWANN 2009, Part I. LNCS, vol. 5517, pp. 706–713. Springer, Heidelberg (2009)
13. Jia, C., Gao, X., Hong, B., Gao, S.: Frequency and Phase Mixed Coding in SSVEP-based Brain-Computer Interface. IEEE Transaction on Biomedical Engineering 58(1), 200–206 (2011)
14. Lee, P.-L., Sie, J.-J., Liu, Y.-J., Wu, C.-H., Lee, M.-H., Shu, C.-H., Li, P.-H., Sun, C.-W., Shyu, K.-K.: An SSVEP-Actuated Brain Computer Interface Using Phase-Tagged Flickering Sequences: A Cursor System. Annals of Biomedical Engineering 38(7), 2383–2397 (2010)

15. Aizenberg, I., Moraga, C.: Multilayer Feedforward Neural Network Based on Multi-valued Neurons (MLMVN) and a Backpropagation Learning Algorithm. Soft Comput. 11, 169–183 (2007)
16. Yazicioglu, R.F., Torfs, T., Merken, P., Penders, J., Leonov, V., Puers, R., Gyselinckx, B., Van Hoof, C.: Ultra-low-power biopotential interfaces and their applications in wearable and implantable systems. Microel. J. 40(9), 1313–1321 (2009)
17. Aizenberg, I., Paliy, D., Zurada, J.M., Astola, J.: Blur Identification by Multilayer Neural Network based on Multi-Valued Neurons. IEEE Transactions on Neural Networks 19(5), 883–898 (2008)
18. Aizenberg, I.: Complex-Valued Neurons with Phase-Dependent Activation Functions. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2010, Part. II. LNCS (LNAI), vol. 6114, pp. 3–10. Springer, Heidelberg (2010)
19. Strasburger, H.: The Analysis of Steady State Evoked Potentials Revised. Clin. Vis. Sci. 1(3), 245–256 (1987)
20. Manyakov, N.V., Van Hulle, M.M.: Synchronization in Monkey Visual Cortex Analyzed with an Information-Theoretic Measure. Chaos 18(3), 037130 (2008)
21. Manyakov, N.V., Vogels, R., Van Hulle, M.M.: Decoding Stimulus-Reward Pairing from Local Field Potentials Recorded from Monkey Visual Cortex. IEEE Transactions on Neural Networks 21(12), 1892–1902 (2010)

Spectral Non-gaussianity for Blind Image Deblurring

Aftab Khan and Hujun Yin

The University of Manchester, UK
Manchester, M13 9PL, UK
aftab.khan@postgrad.manchester.ac.uk, h.yin@manchester.ac.uk

Abstract. A blind image deblurring method based on a new non-gaussianity measure and independent component analysis is presented. The scheme assumes independency among source signals (image and filter function) in the frequency domain. According to the Central Limit Theorem the blurred image becomes more Gaussian. The original image is assumed to be non-gaussian and using a spectral non-gaussianity measure (kurtosis or negentropy) one can estimate an inverse filter function that maximizes the non-gaussianity of the deblurred image. A genetic algorithm (GA) optimizing the kurtosis in the frequency domain is used for the deblurring process. Experimental results are presented and compared with some existing methods. The results show that the deblurring from the spectral domain offers several advantages over that from the spatial domain.

1 Introduction

The field of image restoration can be dated back to the early 1960s problem of non linear filtering of convolved signals. It refers to the estimation of the original image from a noisy, convolved version by using some a priori information about the degradation phenomenon. Main goal of the restoration is to estimate the degradation and apply the inverse process in order to recover the original signals. Image restoration is an ill posed inverse problem because in many cases a priori information is either not available or very limited. Major work in the field is related to the restoration of astronomical images and still interests many researchers. Nonetheless it has now found its application in medical imaging, digital films, antique picture restoration, etc, and particularly interests the law enforcement agencies.

Image restoration algorithms are usually based on some form of degradation model that establishes the relationship between the original and blurred images of an imaging system. The blurred image is assumed to be the result of the convolution between the original image and the transfer function of the imaging system. Any imperfection of the imaging system or environment can induce degradation to the captured image. The following subsections provide a brief description of the blurring process and basic schemes used to estimate the original image.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 144–151, 2011.

1.1 Blind Deconvolution

If the image formation process can be modeled as a linear system, a recorded image can be represented as the output of the convolution of the spatial impulse response or point spread function (PSF) of the linear blurring system with the original image (scene). Let $f(i, j)$ present the original image without any form of degradation, $h(i, j)$ be the PSF and the output of the system be given by $g(i, j)$. Mathematically, for a stationary impulse response of the system across the image (i.e. a spatially invariant stationary PSF), the discrete form of the convolution sum is given as,

$$g(i, j) = h(i, j) * f(i, j) + n(i, j) \tag{1}$$

where * represents the 2-D convolution operator and $n(i, j)$ represents additive noise.

The frequency domain model, with spectral coordinates $m$and n, obtained using the Fourier Transform is

$$G(m,n) = H(m,n)F(m,n) + N(m,n) \tag{2}$$

The goal of deblurring is to produce a good approximation of the original image $f(i, j)$. This process is generally known as convolutional filtering or deconvolution [1] and deblurring in the case of restoration of blurred images. In the noise free case, having prior knowledge of the PSF, $H(m,n)$, Eq. (2) can be used to find $F'(m,n)$, an approximation of $F(m,n)$,

$$F'(m,n) = H^{-1}G(m,n) \tag{3}$$

Such that

$$F'(m,n) \approx F(m,n) \tag{4}$$

This is known as inverse filtering. In practice, often little or no information is available about the PSF or the original image. The problem of deconvolution of the two signals when both are unknown is termed as blind deconvolution [2].

Many researchers have presented solutions to this problem. From Richardson-Lucy [3], total variation (TV) [4], minimum entropy deconvolution (MED) [5] to independent component analysis (ICA) [6], the list of methods for deblurring is exhaustive. The following section describes the scheme with the non-gaussianity principles which forms as a base for the current research work.

1.2 Blind Signal Separation (BSS) and Non-gaussianity

Blind signal separation (BSS) is a technique that aims to recover original signals from a set of mixtures without any prior knowledge of the original signals [7]. Assuming statistical independence of the source signals, it implies that according to the Central Limit Theorem the output signals of a linear system are more gaussian. So the scheme tries to find independents signals that are maximally non-gaussian. This also forms the basis of independent component analysis (ICA). Thus a measure of non-gaussianity is required to achieve this solution.

Higher-order cumulants can measure the difference between a random vector and a gaussian random vector with an identical mean vector and covariance matrix [6]. As a consequence, they can be used for extracting the non-gaussian part of a signal. Kurtosis, a fourth order cumulant, is generally employed to measure non-gaussianity of a signal. Mathematically, kurtosis is defined as

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \tag{5}$$

For y with unit variance, it reduces to

$$kurt(y) = E\{y^4\} - 3 \tag{6}$$

This is equal to the normalized form of the fourth standardized moment $E\{y^4\}$. Negentropy is also used for measuring non gaussianity [6].

In [8] a blind deconvolution and deblurring algorithm based on non-gaussianity measures and the genetic algorithm has been proposed. It uses kurtosis based fitness function for estimating the blur parameters. It has been demonstrated that this technique is effective in restoring severely blurred images without any knowledge of either the image or PSF characteristics. The method is computationally intensive, as every iteration involves transformation between spatial and frequency domains.

Extending the idea of BSS and ICA, we can fairly assume that the source signals are independent in the frequency domain as well as in the time domain. Under this assumption, a blind deconvolution system using the genetic algorithm optimizing the non-gaussianity measure (kurtosis) in the frequency domain is proposed. To distinguish between the time and frequency domain kurtosis, we term 'spectral kurtosis' for frequency domain and 'spatial kurtosis' for the time domain measures.

The following section presents a comparison among the spatial and spectral kurtosis measures; while Section 3 describes the proposed method in detail with the experimental setup and results on various degradation functions. Discussion and conclusions are given in Section 4.

2 Spectral Kurtosis for Iterative Blind Deconvolution (IBD)

2.1 Spectral Kurtosis as Non-gaussianity Measure

The image blurring process makes the blurred images more gaussian than the original one by producing correlation among the adjacent pixels of the image. Fig. 2 verifies this fact in the frequency domain of the image by using the spectral kurtosis measure. Images in Fig. 1 were blurred with gaussian PSF of increasing variance.

The spectral kurtosis measure was calculated by

$$kurt(Y) = \log_{10}(E\{Y^4\} - 3) \tag{7}$$

where Y is the absolute value of the Fourier transform of y. The logarithmic value is taken only for scaling purpose.

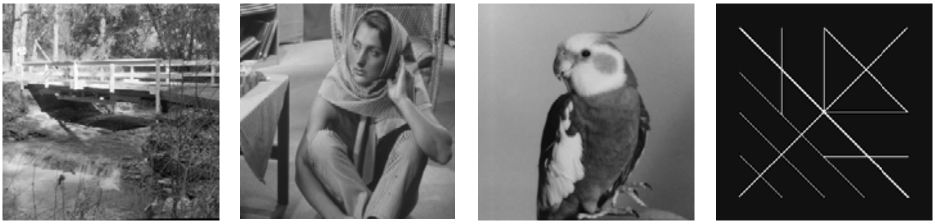


Fig. 1. Test images for gaussianity analysis (Bridge, Barbara, Bird and Crosses)

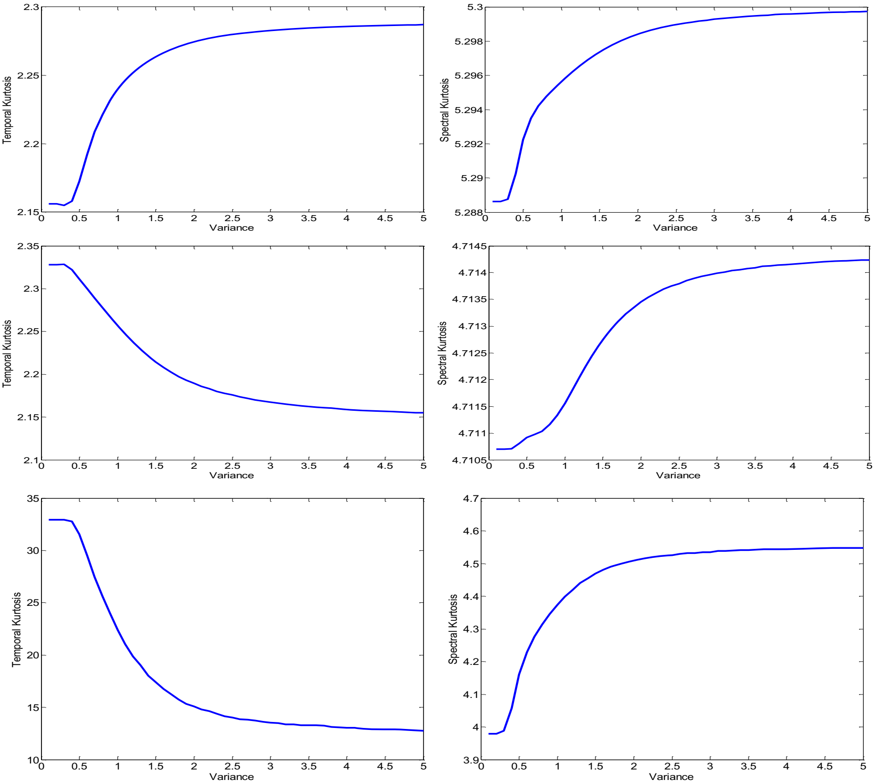


Fig. 2. Gaussianity analysis: left and right columns are measured spatial kurtosis and spectral kurtosis for gaussian blurred images of Bridge, Barbara, Bird, and Crosses respectively. The gaussian blurring variance varies from 0.1 to 5.0.

Fig. 3 demonstrates the nongaussianity measures of the two images after deblurring on various values of theta for motion deblurring. The two images have been blurred with motion blur PSF of angles 23 and 55 respectively. The blurred image was inverse filtered with various values of theta and its spectral kurtosis was measured as shown in the figure. The value of theta on which the system optimized the spectral kurtosis is the actual value of motion blurring applied to the images.

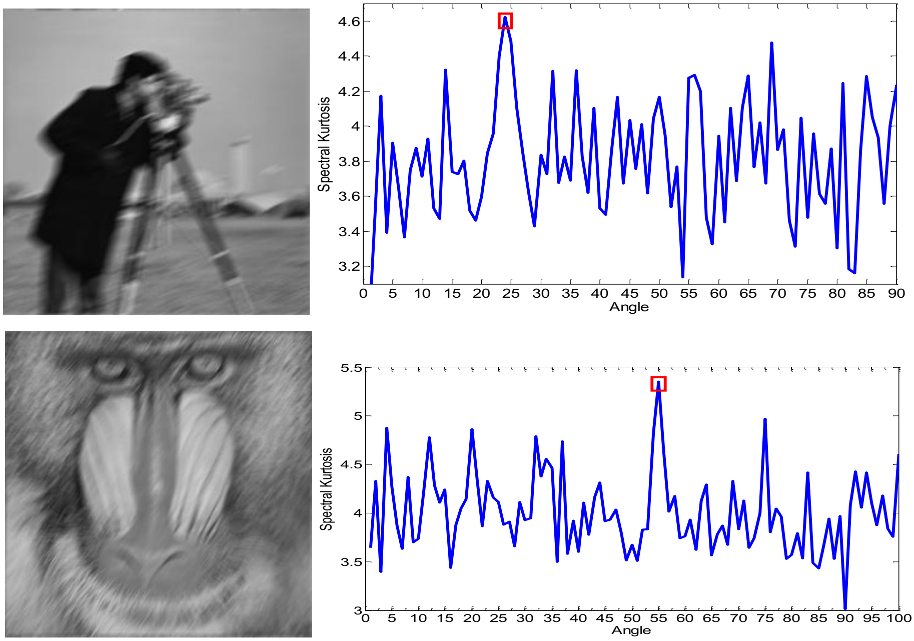


Fig. 3. Spectral non gaussianity (kurtosis) analysis for the cameraman image (top) and mandrill image (bottom) under motion blur. The kurtosis maximizes at the angle (theta) value on which the image was blurred (depicted by the square).

2.2 Proposed Scheme

A blind image deconvolution approach using ICA and genetic algorithm in the frequency domain is proposed. The estimated (deblurred) image is obtained from the version of inverse filter that optimizes the spectral domain non gaussianity measure (kurtosis). The kurtosis based fitness function is used in the genetic algorithm to find the blur parameters in an iterative manner. Since the operation is calculated in the frequency domain, it is more computationally efficient, compared to the spatial kurtosis scheme. The efficacy of the scheme is improved by the fact that the image from the frequency domain is not transformed back into the time domain for computing the non gaussianity measure (since spectral kurtosis is measured in the frequency domain). The scheme is summarized as follows.

- Initialize the genetic algorithm parameters i.e. population, size, crossover rate, mutation rate etc.
- Perform first iteration and for different values of the optimizing parameter (e.g. sigma in case of gaussian blur, theta in case of motion blur etc); find the restored image through inverse/wiener filtering in the spectral domain and calculate its spectral kurtosis (i.e. the fitness function) for different population samples.

- Generate the child population for the next iteration by evolving from the parents on the basis of the fittest function in subsequent iterations (generations).

The proposed method is easy to implement. Often only one or few parameters of the blurring filter are optimized like, e.g. sigma in case of gaussian and theta in case of motion blurs.

Table 1. Parameters and PSNR performance of test images with gaussian blur

Image	Original σ	PSNR (dB)	Estimated σ by spatial kurtosis	PSNR (dB)	Estimated σ by spectral kurtosis	PSNR (dB)
Barbara	2.9	11.2	2.1	14.58	2.5	18.8
Boat	0.9	13.5	1.3	19.85	1.1	26.1

The proposed scheme was tested on various images degraded with different degrading functions (without the presence of noise). The genetic algorithm was able to estimate the blurring parameters of the PSF. Fig. 4 shows the restoration of gaussian blurred images with different values of sigma. The proposed scheme was able to provide better estimates of the variance (sigma) of the gaussian blur than the spatial domain. The optimized parameters and the PSNR performances of the blurred and restored images are given in Table 1.

Fig. 5 presents the motion blurred images and their deblurred results. Exact parameters were identified using the proposed scheme. In this case both the length of the blur and angle were estimated correctly. Table 2 shows these parameters and the PSNR values of the blurred and deblurred images.

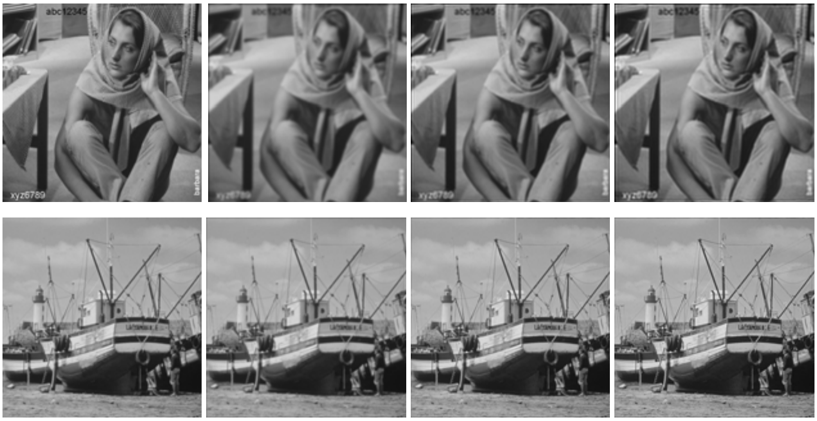


Fig. 4. Column 1: original images, column 2: gaussian blurred, column 3: restored using spatial kurtosis, and column 4: restored using spectral kurtosis.

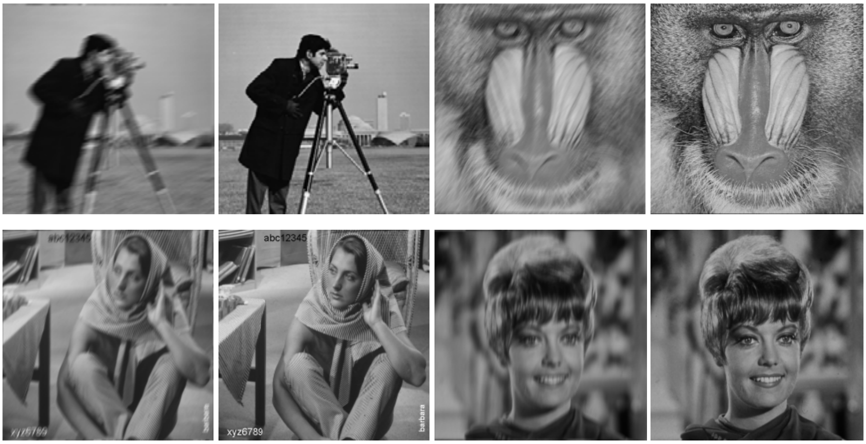


Fig. 5. Motion blurred images (left) and their spectral kurtosis based estimates (right)

Table 2. PSNR performance of test images with motion blur

Image	Original blur parameter		Estimated value	
	Theta	PSNR (dB)	Theta	PSNR (dB)
Cameraman	23	17.3	23	67.14
Barbtext	63	15.6	63	66.16
Mandrill	55	19.2	55	65.18
Zelda	83	19.9	83	66.14

Fig. 6 presents the out of focus blurred images and restored ones using spectral kurtosis scheme. Cameraman and Lena image were blurred with blur radius of length 7 and 13 respectively. Spectral kurtosis estimated the exact parameter at the global maxima. Fig. 7 shows the spatial and spectral kurtosis curves for deblurring processes of Cameraman image.

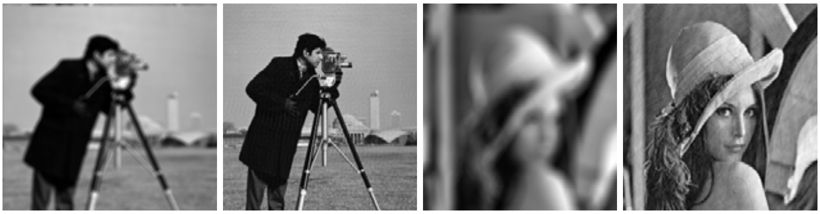


Fig. 6. Out of focus blurred images (left) and their spectral kurtosis based estimates (right)

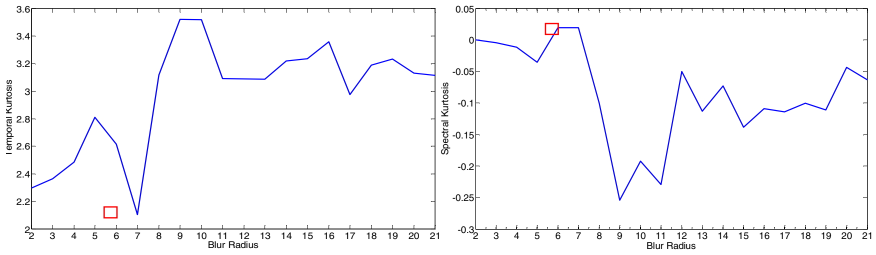


Fig. 7. Estimation of out of focus blur radius using spatial kurtosis (left) and spectral kurtosis (right). Blur radius of 7 was estimated at the point of global minima for spatial kurtosis and at global maxima in case of spectral kurtosis.

3 Conclusions

An improved method based on nongaussianity measure in frequency domain and the genetic optimization algorithm has been proposed for blind image deblurring. The proposed method is simple and easy to implement. The method was able to estimate blurring parameters such as the width of the gaussian blur and both length and angle of the motion blur. As the processing is entirely operated in the frequency domain, it is computationally efficient. As the kurtosis can be prone to outliers, negentropy can be a suitable alternative measure for the scheme. The experimental results show the capability and advantages of the proposed method. A comparison with the spatial kurtosis based scheme indicates improved performance of the proposed. Future endeavor is to tackle noise and outliners during function optimization.

References

1. Oppenheim, A.V., Schafer, R.W., Stockham, T.G.: Nonlinear Filtering Of Multiplied and Convolved Signals. Proceedings of the Institute of Electrical and Electronics Engineers 56, 1264 (1968)
2. Stockham, T.G., Cannon, T.M., Ingebretsen, R.B.: Blind Deconvolution through Digital Signal-Processing. Proc. IEEE 63, 678–692 (1975)
3. Richardson, W.H.: Bayesian-Based Iterative Method of Image Restoration. Journal of the Optical Society of America 62, 55 (1972)
4. Chan, T.F., Wong, C.K.: Total Variation Blind Deconvolution. IEEE Trans. Image Process. 7, 370–375 (1998)
5. Wiggins, R.A.: Minimum Entropy Deconvolution. Geoexploration 16, 21–35 (1978)
6. Hyvarinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley-Interscience Publication, Hoboken (2001)
7. Comon, P., Jutten, C.: Handbook of Blind Source Separation (Independent Component Analysis and Applications). Elsevier, Amsterdam (2010)
8. Yin, H.J., Hussain, I.: Independent Component Analysis and Non-Gaussianity for Blind Image Deconvolution and Deblurring. Integr. Comput.-Aided Eng. 15, 219–228 (2008)

A Graph Partitioning Approach to SOM Clustering

Leandro A. Silva[1] and José Alfredo F. Costa[2]

[1] School of Computing and Informatics, Mackenzie Presbyterian University, São Paulo, Brazil
prof.leandro.augusto@mackenzie.br
[2] Departament of Electrical Engineering, Federal University, UFRN, Brazil
alfredo@ufrnet.br

Abstract. Determining the number of clusters has been one of the most difficult problems in data clustering. The Self-Organizing Map (SOM) has been widely used for data visualization and clustering. The SOM can reduce the complexity in terms of computation and noise of input patterns. However, post processing steps are needed to extract the real data structure learnt by the map. One approach is to use other algorithm, such as K-means, to cluster neurons. Finding the best value of K can be aided by using an cluster validity index. On the other hand, graph–based clustering has been used for cluster analysis. This paper addresses an alternative methodology using graph theory for SOM clustering. The Davies–Bouldin index is used as a cluster validity to analyze inconsistent neighboring relations between neurons. The result is a segmented map, which indicates the number of clusters as well as the labeled neurons. This approach is compared with the traditional approach using K-means. The experimental results using the approach addressed here with three different databases presented consistent results of the expected number of clusters.

Keywords: Self-Organizing Map, Data clustering, K-Means; graph theory.

1 Introduction

Data clustering, also known as cluster analysis, is an important methodology in exploratory data analysis. The objective of data clustering is to discover the natural groupings, K, of M input patterns, data points or objects, $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_M\}$ [1], [2].

In pattern recognition, data clustering is part of the unsupervised learning problems (clustering), which do not use category labels that tag objects with prior identifiers, i.e., class labels [3].

There are many scientific fields and applications that utilize clustering techniques such as: image segmentation in computer vision [2], customer grouping in marketing [4], document multimedia data clustering in information retrieval [5] and others.

Clustering algorithms can be divided into two groups: hierarchical and partitional. The most well-known hierarchical algorithms are single-link and complete-link; the most popular and the simplest partitional algorithm is K-means. Both clustering algorithms, hierarchical and partitional are not capable of automatically determining the number of clusters, which has been one of the most difficult problems in data clustering [1]. Usually, clustering algorithms are run with different tree cuts, in

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 152–159, 2011.

hierarchical algorithm, which is implemented with a dendrogram or with different values of K, in partitional algorithm; the best value is then chosen based on a predefined criterion [1], [2], [6].

Determining automatically the number of clusters, involves making several clustering trials with different values of K, which increases computational complexity. Thus, the Self-Organizing Map (SOM) [3], [7] emerges as a pre-processing stage in a process of automatically determining the number of clusters. SOM projects input patterns on prototypes of a low-dimensional regular grid. A set of prototypes is used as an intermediate step and the total complexity is reduced, as well as there is a reduction of noise in the input patterns [6].

Many graph theoretic clustering have been proposed for cluster analysis. These consider that the input patterns to be clustered are represented by an undirected adjacency graph with edge capacities assigned to reflect the similarity between the linked vertices [1], [2]. The problem with these approaches remains its computational complexity, because each input pattern is a vertex in the graph. The use of SOM can be again an interesting approach. The regular grid (2-D) output of a SOM map can be viewed as a display of neurons, each of them specialized in sets of signals (data patterns). Considering neighboring relations among neurons, a form of a graph which defines the relationship used during training to update neighbor units, some heuristics can be used to cut off inconsistent edges and enabling automatic segmentations of the map. Information that can be used to aid this process include [8]: large distance between the weight vectors of two adjacent neurons greater than the mean distance of all other adjacent neurons to them; importance of neuron in terms of assigned signals (hits) regarding some data statistics (e.g., less than 0.5 of mean expected for all neurons); disparity between signal means and neuron weight values, and so on. Usually, some of the involved parameters are experimentally defined, which makes their use in others databases difficult [8].

This paper addresses an alternative method using graph partitioning for SOM clustering. The trained SOM map is considered a graph; the Davies–Bouldin index (DBI) cluster validity [9], [10] is used to measure the overlap between neighboring neurons; and a threshold is defined to cut out inconsistent neighboring edges between neurons. The result is a segmented map that represents the clusters. This approach has as main advantages: (1) has a simple heuristic and (2) involves the choice of only one parameter value.

Comparisons are performed with results of traditional approach for automatically determining the number of clusters using K-means for clustering SOM neurons. Three different databases were utilized in the experiments.

The remainder of the paper is organized as follows: Section 2 presents a brief explanation of Self-Organizing Map. The proposed method to determine the number of clusters is described in section 3. Experimental results and discussion are given in Section 4. Conclusions and final remarks are shown in section 5.

2 Clustering of the Self-Organizing Map

A Self-Organizing Map (SOM) consists of neurons located on a regular low-dimensional grid, usually two-dimensional (2-D). The lattice of the 2-D grid is either hexagonal or rectangular. Assume that each input pattern from the set of patterns (X)

$\mathbf{x}_i$ is defined as a real vector $\mathbf{x}_i = [x_{i1}, x_{i2},\ldots, x_{id}]^T \in \Re^d$. Each neuron has a d-dimensional weight vector $\mathbf{w}_u = [w_{u1},w_{u2}, \ldots,w_{ud}]^T \in \Re^d$ [7] called a prototype. Fig. 1 illustrates the SOM architecture.

The SOM training algorithm is iterative. Initially, in $t = 0$, the weight vectors are randomly initialized, preferably from the input vectors domain [7]. At each training step t, an input pattern $\mathbf{x}_i(t)$ is randomly chosen from a training set (X). General distances between $\mathbf{x}_i(t)$ and all weight vectors are computed. The winning neuron is the prototype closer to $\mathbf{x}_i(t)$ or the Best Match Unit (BMU). The BMU weight vector is updated, as well as the vector of weights of neighboring neurons, but with minor intensity (see [7] for the complete SOM training algorithm).

The SOM is especially suitable for data survey because it has prominent visualization properties. It creates a set of prototype vectors representing the set of input patterns and carries out a topology-preserving projection of the prototypes from the n-dimensional input space onto a low-dimensional grid. This ordered grid can be used as a convenient visualization surface for showing different features of the SOM (and thus of the input patterns); for example, the cluster structure [7].

K-means can be used for the clustering of the SOM, and consequently for automatically determining the number of clusters. The two-stage procedure - first using SOM to produce the prototypes that are then clustered in the second stage - performs well when compared with direct clustering of the data, in reducing the computational time and the noise [6]. However, the methodology is applied in prototypes, i.e. the K-means is performed in prototypes instead of input patterns such as the dendrogram.

Graph theoretic clustering represents the data points as nodes in a weighted graph. The edges connecting the nodes are given weights by their pair-wise similarity. The central idea is to partition the nodes into two subsets A and B, for example, such that the cut size, i.e. the sum of the weights assigned to the edges connecting nodes in A and B, is minimized [2]. In the next subsection, the graph theoretic approach to SOM clustering is discussed.

3 Graph Partitioning Approach to SOM Clustering

The SOM map to be clustered is an undirected adjacency graph G; each vertex of G corresponds to a neuron in the Map, and an edge links two vertices in G if the corresponding neurons are neighbors according to the Map topology, i.e. G(V,A) is a graph with vertex set V={v_1, v_2,…,v_n,…,v_N} with N as the number of neurons (for example, in Fig. 1, N=9) and edge set A={[$a_{i,j}$, $a_{i,j+1}$];[$a_{i,j}$, $a_{i+1,j}$]} denoting the connection between adjacent neurons defined by regular topology (see Fig. 1).

In this work an undirected graph is used (if neighboring vertices are adjacent) and the adjacency matrix (A) is given by the Map topology. This is represented in Table 1. If there is an edge from an adjacent vertex, then the element in A is 1, otherwise it is 0. In terms of computing, this matrix makes it easy to find subgraphs or, in this case, the clusters.

The proposal advanced here is a threshold graph, which is defined as an undirected adjacency graph, where there is an edge between two vertices if and only if the DBI measured from the weight vector adjacent is greater or equal to *v*, where *v is* a given threshold value. The steps of the algorithm is described as follows:

1) Given a trained SOM map, compute the DBI between adjacent neurons as suggested in Table 1

2) For each adjacent neuron the edge is considered adjacent when DBI >= v

3) For each edge inconsistency (DBI < v), a null connection is considered in position i,j of the Map (represented in Fig. 1 by dotted lines); otherwise set 1 in position i,j (represented in Fig. 1 by continuous lines)

4) A different code is assigned to each connected neuron set (represented in gray levels in Fig. 1).

The result is a partitioned Map, which indicates the number of clusters as well as labeled neurons, as showed in Fig. 1. After this process the input patterns can be projected into a Map and the label of the neuron can be assigned to each input. The experimental results of the proposed approach are contrasted with the K-mean approach. In the next section, experimental results from this proposal are discussed.

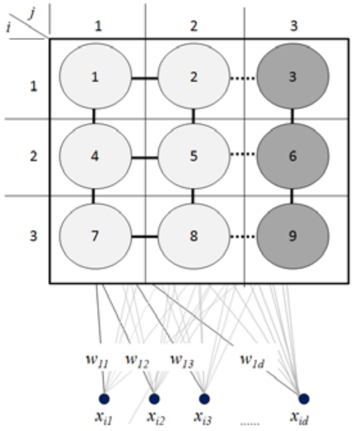


Fig. 1. SOM map architecture. Each neuron is labeled with a number and the grid is also indicated by a matrix position. The continuous line '—' represents arc consistency and the dotted lines '….' represents arcs inconsistency. The color of neurons indicate a group, in this case, 2 clusters.

Table 1. Adjacency Matrix: the values used here are from Fig.1

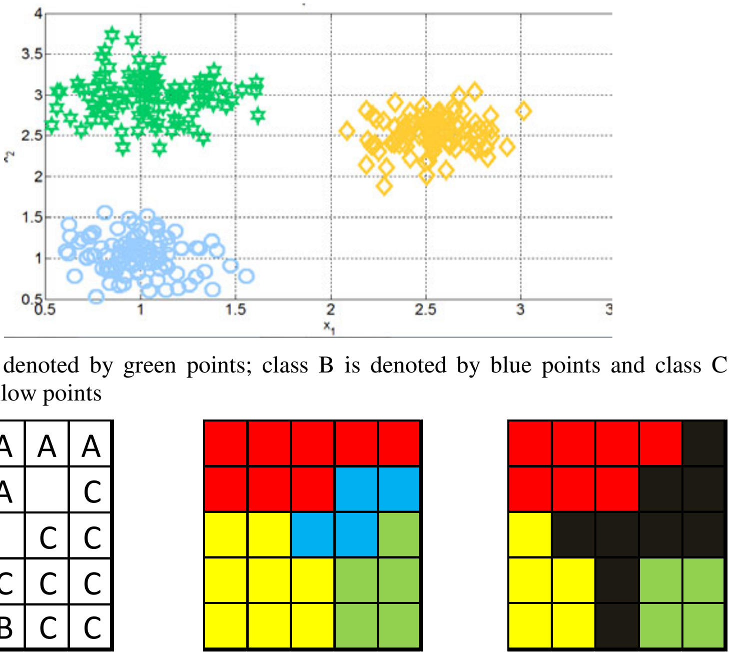
n	*i,j+1*	*i+1,j*
1	DBI($\mathbf{w}_1$,$\mathbf{w}_2$)	DBI($\mathbf{w}_1$,$\mathbf{w}_4$)
2	DBI($\mathbf{w}_2$,$\mathbf{w}_3$)	DBI($\mathbf{w}_2$,$\mathbf{w}_5$)
i,j	DBI($\mathbf{w}_{i,j}$,$\mathbf{w}_{i,j+1}$)	DBI($\mathbf{w}_{i,j}$,$\mathbf{w}_{i+1,j}$)
N	0	0

4 Experimental Results

The system was implemented on an Aton 1.67 GHz 2GB RAM computer using MATLAB software. The SOM Toolbox [11] was used in these experiments. Some specific new functions were implemented to conduct the SOM clustering using graph theory. The approach using K-means for SOM clustering was implemented in the SOM Toolbox.

Three databases were used in our experiments:

- Synthetic database: a dataset with 3 classes generated with a Gaussian distribution consisting of 300 input patterns, 100 points for each class. The three classes are linearly separable.
- Spiral-database: A two-dimensional two-spiral dataset consisting of 2000 input patterns, 1000 points for each class. The two classes are not linearly separable.
- Iris-database: The data set contains 3 classes of 50 input patterns each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other.



a) Class A is denoted by green points; class B is denoted by blue points and class C is denoted by yellow points

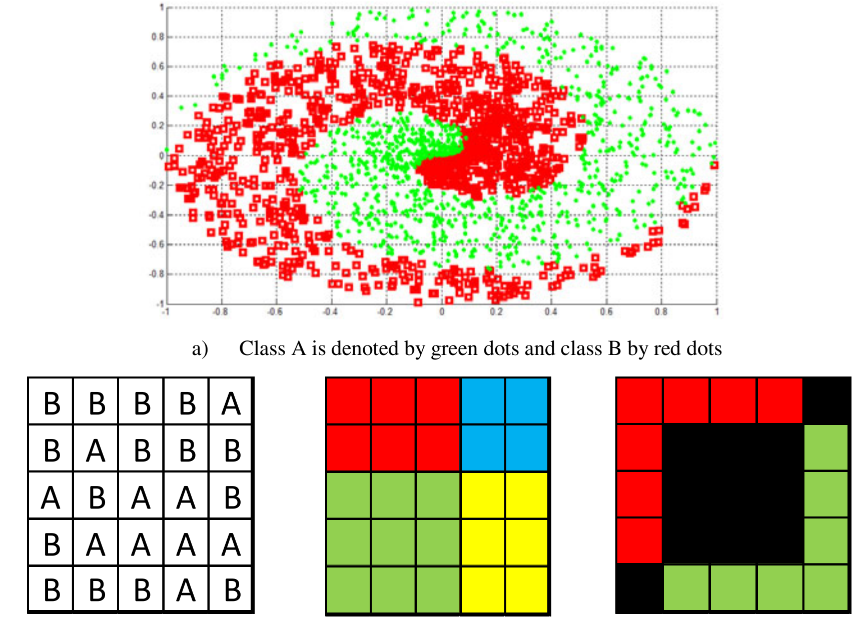
A	A	A	A	A
A	A	A		C
B			C	C
B	B	C	C	C
B	B	B	C	C

b) The SOM map with each neuron labeled with the class with the highest frequency

c) Example of a figure caption. (figure caption)

d) The SOM map clustered by graph. The results indicate 3 clusters, the black color representing the inconsistent neurons

Fig. 2. Experimental results using the Synthetic database



a) Class A is denoted by green dots and class B by red dots

B	B	B	B	A
B	A	B	B	B
A	B	A	A	B
B	A	A	A	A
B	B	B	A	B

b) The SOM map with each neuron labeled

c) The SOM map clustered by K-means. The results indicate 4 clusters

d) The SOM map clustered by graph. The results indicate 2 clusters

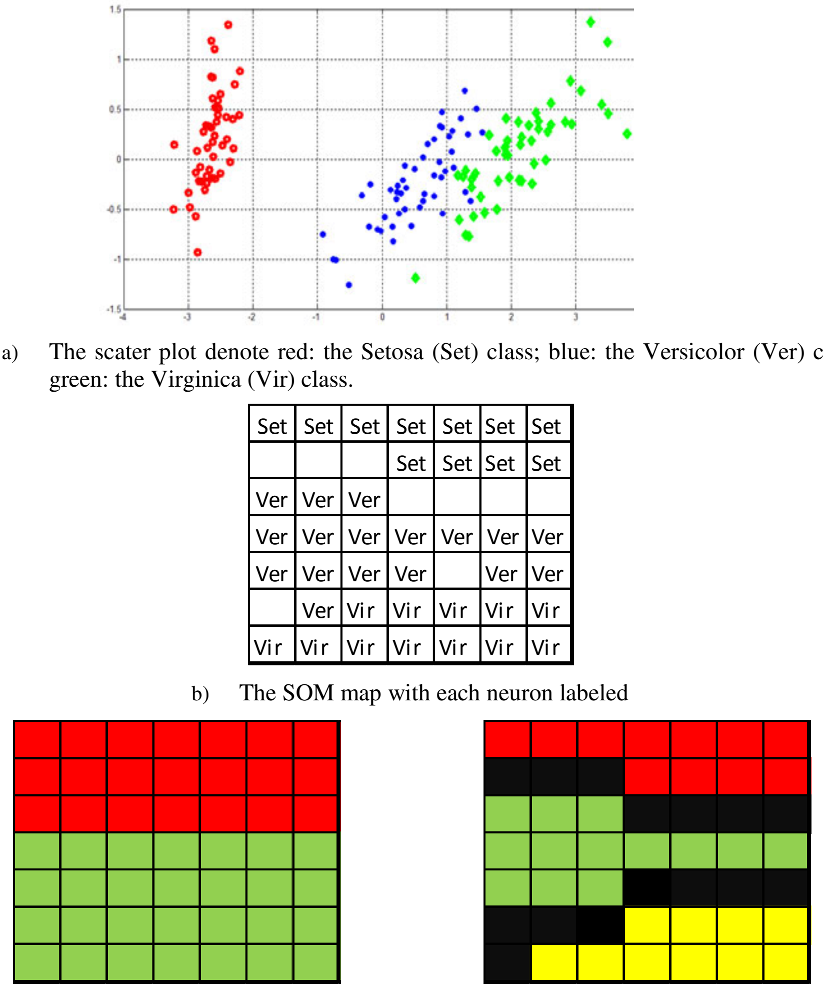
Fig. 3. Experimental results using two spirals in XY-plane database

For each experiment, a different database is used and the approach addressed here is compared with K-means SOM clustering [6]. For each experiment four graphs are showed: a) plot of the database; b) The SOM map, where each neuron is labeled with the class with the highest frequency; c) the SOM clustering using K-means and d) the Map clustering using the proposal addressed here.

For the two approaches a SOM map was trained with rectangular topology, learning rate 0.5, Gaussian update function; a linear weight vector was used to initialize. The heuristic to define the threshold parameter in the experiments was the mean value of DBI from the adjacency matrix.

Fig. 2 represents the results using the synthetic database. For this experiment, a SOM map with 25 neurons (5 x 5) was used. The proposal addressed here found the correct number of clusters to be 3, with the black neurons representing those that are inconsistent or that do not represent any input patterns. Contrary to this, the number of clusters showed by K-means was 4. This is a bad result obtained after several experiments, where in some cases the correct number of clusters was obtained. This can happen due to the initialization processes of K, in K-means.

Fig. 3 shows the results for the spiral database using a SOM map with 25 neurons (5 x 5). Here, in all experiments using K-means the results were incorrect. This can happen because of the assumptions on the form of clusters. For example, K-means tries to find spherical clusters. Yet, the proposal addressed here found 2 clusters to be correct and the segmented Map is shown in a form that represents the data distribution.



a) The scater plot denote red: the Setosa (Set) class; blue: the Versicolor (Ver) class; green: the Virginica (Vir) class.

Set	Set	Set	Set	Set	Set	Set
			Set	Set	Set	Set
Ver	Ver	Ver				
Ver	Ver	Ver	Ver	Ver	Ver	Ver
Ver	Ver	Ver	Ver		Ver	Ver
	Ver	Vir	Vir	Vir	Vir	Vir
Vir	Vir	Vir	Vir	Vir	Vir	Vir

b) The SOM map with each neuron labeled

🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟥	🟥	🟥	🟥	🟥	🟥	🟥
🟩	🟩	🟩	🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩	🟩	🟩	🟩
🟩	🟩	🟩	🟩	🟩	🟩	🟩

c) The SOM map clustered by K-means. The results indicate 2 clusters.

🟥	🟥	🟥	🟥	🟥	🟥	🟥
⬛	⬛	⬛	🟥	🟥	🟥	🟥
🟩	🟩	🟩	⬛	⬛	⬛	⬛
🟩	🟩	🟩	🟩	🟩	🟩	🟩
🟩	🟩	🟩	⬛	⬛	⬛	⬛
⬛	⬛	⬛	🟨	🟨	🟨	🟨
⬛	🟨	🟨	🟨	🟨	🟨	🟨

d) The SOM map clustered by graph. The results indicate 3 clusters.

Fig. 4. Experimental results using the Iris-database

Fig. 4 shows the results for the Iris-database using a SOM map with 49 neurons (7 x 7). To illustrate the class separability, the scatter plot (2D) was used. Here, the number of clusters found using the approach is 3. Meanwhile, the number of clusters shown by K-means was 2, i.e. the non-linearly separable classes were grouped in one cluster. This is a bad result obtained after several experiments, where in some experiments the correct result was obtained.

5 Conclusion

Automatically determining the number of clusters has been one of the most difficult problems in data clustering. In this context, this work proposes an approach that applies graph theory to SOM clustering.

The results supported the use of the graph theoretic approach as in all experiments the number of clusters discovered was correct. On the other hand, the SOM clustered by K-means is sensitive to K initialization and input data distribution; for example, in experiments using the Synthetic database and Iris-database in some cases the number of clusters was correct, but in most cases the results were wrong (as shown in Fig. 2c and Fig. 4c); the same happened in experiments using the Spiral-database (Fig. 3c).

These interesting results for this approach are still preliminary and extensive studies, such as: verifying the sensibility with other weight vector initializations, using different size Maps, using another Map topology and others, need to be carried out.

References

1. Jain, A.L.: Data clustering: 50 years beyond K-Means. Pattern Recognition Letter 31(8), 651–666 (2010)
2. Wu, Z., Leahy, R.: An optimal graph thoretic approach to data clustering: theory ans its application to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(11) (November 1993)
3. Duda, R., Hart, P., Stork, D.: Pattern Classification and Scene Analysis. John Wiley Professio. Wiley (2000)
4. Sassi, R., Silva, L.A., Del-Moral-Hernandez, E.: A Methodology Using Neural Network to Cluster Validity Discovered from a Marketing Database. In: Brazilian Symposium on Neural Networks, SBRN 2008, vol. 08, pp. 3–8 (2008)
5. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-Based Multimedia Information Retrieval: State of the Art and Challenges. ACM Transactions on Multimedia Computing, Communications and Applications 2(1), 1–19 (2006)
6. Vesanto, J., Alhoniemi, E.: Clustering of the self-organizing map. IEEE Transaction on Neural Network 11, 586–600 (2000)
7. Kohonen, T.: Self-Organizing Maps. Third extended edn. Springer, Heidelberg (2001)
8. Costa, J.A.F., Netto, M.L.A.: Segmentação do SOM baseada en particionamento de grafos. In: Proceedings of the VI Brazilian Conference on Neural Networks - CBRN, pp. 451–456 (2003)
9. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans Patt. Anal. Machine Intell., PAMI-1, 224–227 (1979)
10. Halkidi, M., Batistakis, Y., Michalis, V.: On Clustering Validation Techniques. Journal of Intelligent Information Systems 17(2), 107–145 (2002)
11. SOMToolbox. SOM toolbox, a function package for matlab 5 implementing the self-organizing maps, SOM (2011), `http://www.cis.hut.fi/projects/somtoolbox/`

Using the Clustering Coefficient to Guide a Genetic-Based Communities Finding Algorithm

Gema Bello, Héctor Menéndez, and David Camacho

Departamento de Ingeniería Informática, Escuela Politécnica Superior,
Universidad Autónoma de Madrid,
C/Francisco Tomás y Valiente 11, 28049 Madrid, Spain
{gema.bello,hector.menendez,david.camacho}@uam.es
http://aida.ii.uam.es

Abstract. Finding communities in networks is a hot topic in several research areas like social network, graph theory or sociology among others. This work considers the community finding problem as a clustering problem where an evolutionary approach can provide a new method to find overlapping and stable communities in a graph. We apply some clustering concepts to search for new solutions that use new simple fitness functions which combine network properties with the clustering coefficient of the graph. Finally, our approach has been applied to the Eurovision contest dataset, a well-known social-based data network, to show how communities can be found using our method.

Keywords: clustering coefficient, social networks, community finding, genetic algorithms.

1 Introduction

The clustering problem is based on blind search on a dataset. Some classical solutions such as K-means (for a fixed number of clusters) [7] or Expectation-Maximization [3] (for a variable number of clusters), amongst others, are based on distances or metrics that are used to determine how the cluster should be defined. The clustering problem is harder when is applied to find communities in networks. Some algorithms such as Edge Betweenness [5] or CPM [4] have been designed to solve this problem following a deterministic process. In our study of the previous problem, we adopt an evolutionary approach based on the K-means algorithm, a popular and well-known algorithm. It is a straightforward clustering guided method (usually by a heuristic or directly by a human) which tries to classify data in a fixed number of clusters (each element is associated to one class). The number of clusters can be predefined or can be estimated using heuristics or other kinds of algorithms, such as genetic algorithms [6].

In the process of community finding problems, K-means cannot be directly applied because it does not allow overlapping. In contrast, it is common for communities to share members. An alternative solution could be fuzzy k-means [8] which allows every one element to belong to several clusters giving a probability

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 160–169, 2011.

of membership, so same kind of overlapping for an element can be considered. Communities in networks have been studied using CPM (Clique percolation method) and Edge Betweenness algorithms which have been applied in our previous work [2] for community classification. CPM (Clique percolation method) [4] finds communities using k-cliques (where k is fixed at the beginning and the network is represented as a graph). It defines a community as the highest union of k-cliques. CPM has two variants: directed graphs and weighted graphs. [9] Edge Betweenness [5] is based on finding the edges of the network which connect communities and removing them to determine a good definition of these communities.

Our new approach develops an evolutionary k-means inspired by the concept of fuzzy k-means and with the same objective as CPM and Edge Betweenness algorithms: finding communities or overlapping clusters in the network.

In this work we propose a new way to combine both community finding and clustering algorithms. In our approach, a genetic algorithm is used to find communities in a dataset that represents humans voting on a social network. To guide the genetic algorithm, the fitness takes the clustering coefficient defined in graph theory to improve the results that could be obtained through a simple K-means.

The rest of the paper is structured as follows. Section 2 shows a description about the web dataset used to test our algorithm. Section 3 presents the genetic algorithm used to detect communities in the web dataset. Section 4 presents a discussion about the experimental results obtained. Finally, the conclusions and some future research lines of work are presented.

2 Genetic-Based Community Finding Algorithm (GCF)

The Genetic-based community finding algorithm uses a genetic algorithm to find the best k communities in a dataset that could be represented as a graph and where any particular neighbour could belong to different clusters. To describe GCF, we will explain the following: the codification, the genetic algorithm and the fitness function definition.

2.1 GCF Codification

An important problem in any Genetic Algorithm (GA) is related to the codification of the chromosomes. In our case the genotypes are represented as a set of binary values. Each allele represents the membership of a node of the graph and each chromosome is used to represent a community. In this binary representation 1 means the node belongs to the community and 0 the opposite, see Figure 1 which exemplifies nodes as countries because of the data set to be used for experimentation.

This simple codification allows us to represent nodes belonging to several communities (as we have in fuzzy k-means and CPM algorithm), and also provides a simple method to define reproduction, crossover and mutation using a standard Genetic Algorithm strategy[10].

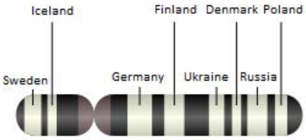


Fig. 1. A Chromosome representing graph nodes. In this case, each node represents a country and its belonging or not to the current community.

2.2 GCF Evolutionary Approach

The GCF strategy works as follows:

1. A random population of communities is generated.
2. The population evolves using a standard GA.
3. The chromosomes that are the k-best solution of the algorithms are selected. The selection process subsumes the communities which have better fitness and belong to a bigger community. The process has the following steps:
   (a) A list of k communities is created.
   (b) The chromosomes are sorted by their fitness value.
   (c) If there is an empty position in the list or one of the members is contained in the chromosome that we are going to check, we add the checked chromosome in the mentioned position (or in an empty position) subsuming the other.
   (d) If the list is full and the chromosome that we are going to check does not satisfy the last condition, the algorithm stops. It also stops if the fitness of the new chromosome is bigger than a fixed value (in this case, the value is fixed as half of the maximum fitness).

Defining and selecting an appropriated fitness function, which we will now discuss, is the most critical issue in the GCF algorithm as it will be used to optimize the quality of communities.

2.3 GCF Fitness Functions

For this problem we have implemented three kind of fitness functions, each of which has a different goal. The first one tries to find nodes with a similar rating behaviour (minimal distance fitness), the second one tries to find clusters using the clustering coefficient (maximum clustering coefficient fitness) and, finally, the last fitness function combines both strategies trying to find communities with similar rating behaviours whose members are connected between them (hybrid fitness).

Minimal Distance Fitness (MDF). The objective of this fitness function is to find communities of nodes that are similar. The evaluation of this fitness function are done using the following criteria:

1. Each node belonging to a community is represented as a vector of attributes. The definition of these attributes depends on the problem being solved.

2. The average euclidean distance between vectors of attributes within a community is calculated. The fitness calculates distances to be taken into account from peer to peer, between all vectors.
3. The fitness value for the community is the average distance of the values calculated in previous step (we are trying to minimize the fitness). It is a measure of similarity for those rows, hence it checks if they follow the same ballot pattern. We call this average distance d_{in} (see Figure3).
4. Fitness penalizes those cases where the community has a single node, giving it a value of zero.

Maximum Clustering Coefficient Fitness (MC^{2}F). The goal of this fitness is to discover communities whose members are connected between them. It is measured through the clustering coefficient, defined as follows:

Definition 1. *Let $G = (V, E)$ be a graph where E is the set of edges and V the set of vertices. Let $v_i \in V$ be a vertex and $e_{ij} \in E$ an edge from v_i to v_j. Let Σ_{v_i} be the neighbourhood of the vertex v_i defined as $\Sigma_{v_i} = \{v_j \mid e_{ij}, e_{ji} \in E\}$. If k is considered as the number of neighbours of a vertex, we can define the clustering coefficient of a vertex as follows:*

$$C_i = \frac{|\{e_{jk}\}|}{k(k-1)}$$

Where $|\{e_{jk}\}|$ satisfies that $v_j, v_k \in \Sigma_{v_i}$.

Definition 2. *The clustering coefficient of a graph is defined as:*

$$C = \frac{1}{|V|} \sum_{i=0}^{|V|} C_i$$

Where $|V|$ is the number of vertices.

The fitness takes the sub-graph defined by the community and calculates its clustering coefficient. It returns the inverse value, because the genetic algorithm tries to minimize the fitness function.

Hybrid Fitness (HF). This last fitness function combines both Clustering Coefficient and Distance fitness ideas: it tries to find a set of communities satisfying both conditions already defined. With this method we try to find strong and similar communities (members are highly connected between them and they have similar behaviour). The function defined is a simple weighted function: suppose that $F(x, y)$ is the fitness function, CC the clustering coefficient and d_{in} the value of HF fitness is:

$$F_i(CC, d_{in}) = w_1 * \frac{CC_i}{Max(\{CC_i\}_{i=1}^{K})} + w_2 * \frac{d_{in_i}}{Max(\{d_{in_i}\}_{i=1}^{K})}$$

Where w_i are the weights given to each fitness: $w_i \in (0, 1)$. The values were set experimentally to $w_1 = 0.1$ and $w_2 = 0.9$.

3 The Dataset Description

The Eurovision Song Contest has been studied using different clustering methods since the nineties. The main interest was to study and analyse alliances between countries, which has already been reflected by clustering and communities. The data used in this work has been extracted from Eurovision's official website.

3.1 The Dataset Representation: The Eurovision Voting System

Since 1975, the scoring system in the Eurovision Contest consists of the following rules:

- Each country distributes among others participants the following set of points: 1, 2, 3, 4, 5, 6, 7, 8, 10, 12.
- These countries give the highest punctuation to the best song and the lower to the less popular on preferred.
- When all countries cast their votes, the final ranking is obtained and the country with the highest punctuation wins the contest.

This data can be easily represented using a graph for each year of the contest. In this graph, the vertices will be countries and the points emitted can be used to weight the edges. The graph could be *directed* (the edges represent votes), or *undirected* (the edges only connect countries which have exchanged points in any direction). If we consider the latter, it is similar to setting edge weights uniformly to 1. According to this problem, the dataset will be represented as the latter case, we named this representation Eurovision graph, or Eurovision network.

3.2 Study and Comparison of the Eurovision Network in a Random Context

The first approximation that shows patterns can be obtained using a simple comparison between the Eurovision graph and a randomly generated graph with the same rules applied in the contest. Namely, each participant country assigns its ten set of points (generating an edge for every point cast) randomly among the remaining participant countries. We call this representation Random network.

The random network model assumes that a given country does not favours or penalize other countries and all songs have equal musical quality. So a country X will give points randomly to another ten countries. If, for example, there are N countries then the probability that country X votes for country Y is given by P = 10/(N-1). Usually, in social networks, two vertices with corresponding edges to a third vertex have a higher probability of being connected to each other. Hence, it may be possible to observe the same effect in the Eurovision network. Therefore, to study this effect it is reasonable to analyse the clustering coefficient defined in section 2.

When we compare two different graphs, Eurovision and Random graphs, a greater CC in the Eurovision graphs means there is an “intention of vote” between countries. So the graph distribution of edges is not random and we could conclude that communities, or alliances between countries, exist.

Figure 2 shows the clustering coefficients calculated for years ranging between 1992 to 2010. It can be seen how Eurovision clustering coefficients are always greater than random network values. Hence, the results provide an evidence that the voting system is not random and there are some partnerships between countries.

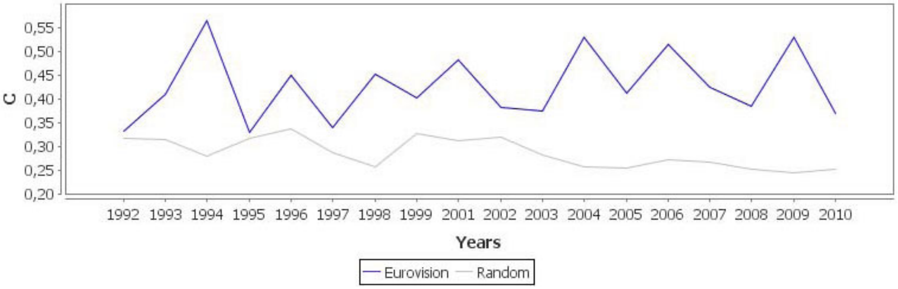


Fig. 2. Clustering Coefficient comparison between the Eurovision network and a random graphs

4 Experimental Results

The preliminary data analysis, showed in Figure 2, confirms the existence of alliances between participant countries. Specifically, 2009 has the greatest difference in clustering coefficient. This means it contains a large set of different communities. Hence, we have selected this year to perform the experimental analysis of our algorithm.

We have calculated the distance between the community centres to compare the results obtained; we call this measure d_{out} as shown in Figure 3. A large distance between countries is preferable as it means a bigger gap between classes or communities, and thus better results.

The genetic parameters of GCF have been set as:

- crossover probability: 0.1
- mutation probability: 0.2
- generations: 2500
- population size: 3000
- selection criteria: $\mu + \lambda$ where μ is the original population (we choose 200 best chromosomes for reproduction process and they also survive), λ is the population generated in the reproduction process
- number of communities (K): 6

K is a parameter of the genetic algorithm that sets the number of communities. Table 1 presents the communities obtained using K equal to 6 for every fitness. This value was experimentally obtained simulating different executions of our algorithm for values of K ranging between 2 and 10. The optimal number of communities with minimal overlapping was found to be 6. In the following subsections we explain the results obtained attending to each fitness.

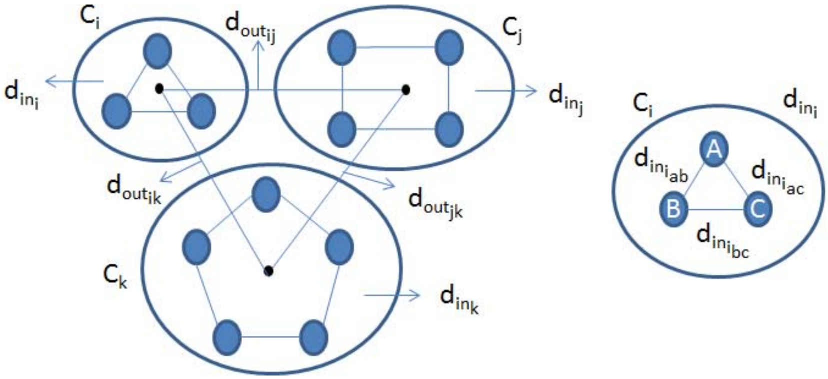


Fig. 3. Sample network graph illustrating three communities and the distances that are calculated in the experimental phase. The distance d_{in} represents the average distance calculated between the countries which belong to a community. And the distance d_{out} represents the distance between community centres.

4.1 Distance Model

The first fitness, MDF, takes the minimum distances (d_{in}) between the points that represent the countries trying to find communities that vote in a similar way. This algorithm was described in section 3.2. From this experiment it can be noticed that the number of countries contained in these communities is dramatically small, as can be seen in Table 1. The d_{in} distance values obtained are lower, meaning that the communities found cast their points very similar, but all of these groups only have two countries.

4.2 Clustering Coefficient Model

This model is based in the clustering coefficients of a network, and it tries to find groups of countries that they are giving votes between them. The resulting communities are shown in Table 1 identified by the fitness called MC^{2}F.

Analyzing the found communities, we see that many of them present high overlapping among countries. This effect was also noticed in the distance between centres (d_{out}), it has decreased dramatically from *14.65* (obtained by the previous model) to *5.40*. Therefore, the communities found are very close to each other, and present a higher overlapping.

Considering the intra-community distance, d_{in}, increases of up to twice the previous values are observed. We can conclude that we have achieved the goal of finding larger groups, but now these groups present too much overlapping to be considered as stable communities. So the final goal of the algorithm has not been really achieved.

4.3 Hybrid Model

Finally, these fitness functions have been combined in a new hybrid fitness (see previous section). The first fitness finds communities which are too small, formed by only 2 countries. The second has a good clustering coefficient and the communities are larger, but the distance between communities is not as good as in the first case, therefore overlapping is too high.

In this last model, combining the two GCF cost functions enables discovering groups of countries which cast votes in a similar way, and also exchange points between them. The communities found are shown in Table 1.

It is interesting to compare these results to the equivalent values for the previous models. The distance between centres, d_{out}, has been greatly improved

Table 1. Communities found with K = 6 using Clustering Coefficient. The distances between centres (d_{out}) obtained by fitness are: (a) MDF = 14.65 (b) MC^{2}F = 5.40 (c) HF = 11.26.

Fitness	**Communities**	d_{in}	**CC**
MDF	**Lithuania Latvia**	**10,91**	**0**
MDF	**Sweden Denmark**	**11,04**	**0**
MDF	**Sweden Hungary**	**11,31**	**0**
MDF	**Cyprus Moldova**	**11,40**	**0**
MDF	**Israel Netherlands**	**11,66**	**0**
MDF	**Albania Germany**	**11,83**	**0**
MC^{2}F	**Sweden Bosnia and Herzegovina Moldova Russia Finland Ukraine Iceland Turkey Germany**	**20,57**	**1**
MC^{2}F	**France Sweden Moldova Russia Finland Iceland Germany Azerbaijan UnitedKingdom**	**21,20**	**1**
MC^{2}F	**France Sweden Moldova Finland Romania Iceland Germany Azerbaijan UnitedKingdom**	**21,78**	**1**
MC^{2}F	**France Estonia Sweden Finland Iceland Germany UnitedKingdom**	**20,93**	**1**
MC^{2}F	**Sweden Moldova Russia Finland Ukraine Iceland Azerbaijan**	**20,55**	**1**
MC^{2}F	**Estonia Sweden Bosnia and Herzegovina Finland Iceland Turkey Germany**	**21,89**	**1**
HF	**Estonia Sweden Finland Iceland**	**18,03**	**1.0**
HF	**Sweden Moldova Russia Finland Ukraine Iceland**	**19,52**	**1.0**
HF	**Norway Sweden Denmark Iceland**	**18,77**	**0.92**
HF	**Moldova Russia Ukraine Poland**	**16,40**	**0.75**
HF	**Armenia Russia Lithuania Ukraine**	**16,56**	**0.75**
HF	**France Germany United-Kingdom**	**19,93**	**1.0**

and now is closer to the value obtained by the first fitness function (*11.26*). The intra-cluster distance, d_{in}, and the clustering coefficient take values lying between the first and second models' values. In addition, we found that the given communities have an appropriate size with a reduced overlapping.

This model allows us to answer two different questions about what standing closer or belonging to the same community means for a group of countries. On the one hand, we can use the similarities in the voting process to establish relationships and, on the other hand, we can consider the points that any country assigns to the other members in its community. Therefore, we can consider that the partnerships found with this model will be stronger and more useful to measure the quality of the community found. They have similar votes and also many of these votes are exchanged between them, globally, these communities have a high number of points.

5 Conclusions and Future Work

To find communities in a web dataset that can be represented by a social network, we have designed and implemented a genetic algorithm based on the graph clustering coefficient. We have centred our research around how to guide the fitness to improve the results that could be obtained through a classical K-means.

We have implemented three different fitness functions: the first one based on a euclidean distance, the second one based on the clustering coefficient, and the last one as a combination of the previous two (hybrid model).

Our experimental findings show that, using the clustering coefficient defined in graph theory to guide the hybrid fitness,is able to reach the best result. This model find communities that have an appropriate size, reduced overlapping and closer distances between centres.

Finally some improvements could be made in the fitness function. In our hybrid model, the fitness could be adapted to accept a weighted clustering coefficient[1] to obtain a better distance. This new fitness could be used in the future to measure the strength of a community. Also, for the Eurovision dataset, other features such as geographical distances or historical behaviours could be included in future fitness functions to study the analysis of the GCF algorithm.

References

1. Barrat, A., Barthélemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. Proceedings of the National Academy of Sciences of the United States of America 101(11), 3747–3752 (2004)
2. Bello, G., Cajias, R., Camacho, D.: Study on the impact of crowd-based voting schemes in the eurovision european contest. In: 1st International Conference on Web Intelligence, Mining and Semantics (WIMS 2011). ACM press, New York (2011)

3. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. Journal of the Royal Statistical Society. Series B (Methodological) 39(1), 1–38 (1977)
4. Derényi, I., Palla, G., Vicsek, T.: Clique Percolation in Random Networks. Physical Review Letters 94(16), 160202-1–160202-4, (April 2005)
5. 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 99(12), 7821–7826 (2002)
6. Gonzalez Pardo, A., Granados, A., Camacho, D., Rodruez, F.D.: Influence of music representation on compression based clustering. In: IEEE World Congress on Computational Intelligence, pp. 2988–2995. IEEE, Los Alamitos (2010)
7. Macqueen, J.B.: Some methods of classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
8. Oussalah, M., Nefti, S.: On the use of divergence distance in fuzzy clustering. Fuzzy Optimization and Decision Making 7, 147–167 (2008)
9. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043), 814–818 (2005)
10. Vose, M.D.: The Simple Genetic Algorithm: Foundations and Theory. MIT Press, Cambridge (1998)

A Framework to Mine High-Level Emerging Patterns by Attribute-Oriented Induction

Maybin K. Muyeba[1], Muhammad S. Khan[2], Spits Warnars[1], and John Keane[2]

[1] Sch. of Computing, Maths and Digital Techn., Manchester Metropolitan University, UK
{m.muyeba,s.warnars}@mmu.ac.uk
[2] Department of Computer Science, School of Electrical Engineering and Computer Science,
University of Liverpool, UK
mskhan@liverpool.ac.uk
[3] School of Computer Science, The University of Manchester, UK
john.keane@cs.manchester.ac.uk

Abstract. This paper presents a framework to mine summary emerging patterns in contrast to the familiar low-level patterns. Generally, growth rate based on low-level data and simple supports are used to measure emerging patterns (EP) from one dataset to another. This consequently leads to numerous EPs because of the large numbers of items. We propose an approach that uses high-level data: high-level data captures the data semantics of a collection of attributes values by using taxonomies, and always has larger support than low-level data. We apply a well known algorithm, attribute-oriented induction (AOI), that generalises attributes using taxonomies and investigate properties of the rule sets obtained by generalisation algorithms.

Keywords: attribute-oriented, algorithm, rulesets, high-level, emerging pattern.

1 Introduction

Data mining aims to find patterns in data. Recently, emerging patterns [1] have become popular for classification problems [7][11]. Emerging patterns (EP) [9] represent contrasting characteristics between two data sets usually expressed as conjunctions of attribute values in a given class of records. The most familiar approaches use classification [6][10][11]. A pattern is emerging (EP) if its support from one dataset to another increases. A pattern is jumping emerging (JEP) if its support from the previous dataset changes from zero to non-zero.

EPs have been successfully used in classification algorithms with mainly low-level (primitive) data. Low-level data has a tendency to be distinct yet represent semantically similar information e.g. for an attribute "Course", there may be two different university degree subjects "Chemistry" and "Physics" that are both in category "Science", a level higher than both subjects. These are two distinct items of data yet they semantically belong to one item "Science". The problem with low-level EP algorithms is the generation of many EPs because of the combinatorial problem in the number of items and also the use of small supports. As a consequence, most EP

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 170–177, 2011.

classifiers use level-wise border searches to control pattern explosion [8][10]. In contrast using high-level summarised data or attribute taxonomies (is-a hierarchies) that capture significant data features often tends to prune common and irrelevant features usually found from low-level data, leaving only high-level supported items [13]. Attribute taxonomies reveal attribute details at various higher levels in the hierarchy. It is well known that larger supports of an attribute's values occur at higher levels than at lower levels of a given taxonomy [12]. Using this fact, it is imperative that EPs can be used to exploit various taxonomic levels of attributes to express varying support levels of item combinations, and hence more significant EPs.

A well established algorithm for mining *is-a* hierarchies from large data to produce conjunctions of attribute-value pairs is attribute-oriented induction (AOI) [14]. Attribute taxonomies, also known as background knowledge or concept hierarchies, are provided by a domain expert or generated automatically. AOI can generate various types of rule patterns, including discriminant, characteristic and classification rules. In the latter case, there is no need to train the data as AOI searches through the input space using both low-level data and their corresponding taxonomies. The reader is referred to [3] for details on the basic AOI algorithm.

Our motivation is three-fold: firstly, AOI is a versatile algorithm for solving the EP problem using various techniques; secondly, larger supports of attribute values mostly occur higher up the taxonomy than would be for low-level values; thirdly, as there is usually a combinatorial explosion of patterns at a low-level, it makes it more difficult for a user to interpret so many patterns compared with a general pattern i.e. patterns expressed at high taxonomic levels. Thus such general patterns have more expressive potential, and represent global data semantics better, than single primitive patterns.

In this paper we give formal definitions of the problem of mining HEPs and introduce and begin to evaluate an algorithm, AOI-HEP (High-level Emerging Patterns) which mines high-level emerging patterns using an enhanced AOI approach.

The paper is organised as follows: Section 2 presents background; Section 3 gives the problem definition; Section 4 introduces the new AOI-HEP algorithm; experimentation is described and analysed in Section 5; and Section 6 presents conclusions.

2 Background

In this section, we introduce formal HEP definitions used in the paper.

Let $D = \{A_1, A_2, .., A_m\}$ be a dataset with m attributes each with a domain $Dom(A_i)$ and N tuples. For each attribute there are a set of values or instances $\{a_i^k\}$, $1 \le k \le m$ and $1 \le i \le N$. We can also label classes to which these instances belong as $C = \{C_1, C_2, .., C_v\}$. An item is an (attribute, instance) pair (A_k, a_i^k). A set of items is an itemset when there is a combination of items, $\bigcup_{i=1}^{n} a_i^k, n < m * N$, for some values of k. The number $m * N$ is the largest possible number of combinations of items for m attributes and N tuples. Accordingly, in AOI, for each attribute there is a taxonomy $H_i, 1 \le i \le m$ linked to it. To define support, we first represent a characteristic AOI rule pattern as a conjunction of items with instance values of any object x, which can be referred to as a complex pattern [7] of the form:

$$A_1(x) = v_1 \wedge .. \wedge A_k(x) = v_k, \quad [s\%] \tag{1}$$

where $s\%$ is the usual support (% of tuples in the dataset) and $v_i \in \{H_i \cup Dom(A_i)\}$. We see that support is for a more complex expression of item-value pairs than single items. Note that $v_i s$ are not necessarily low-level data or domain values but extracted from attribute hierarchies H_i. In AOI, this pattern forms part of a characteristic rule [5]. The definition of itemset uses the term ruleset as a complex pattern represented by (1). A characteristic rule according to [4] is defined as

$$h \rightarrow e \tag{2}$$

where e is evidence (shown as equation 1) and h is the hypothesis or class description we wish to characterise. Thus we rewrite (1) as a characteristic rule:

$$C_i(x) \rightarrow A_1(x) = v_1 \wedge .. \wedge A_k(x) = v_k \; [s\%]$$

in relation to some hypothetical class C_i. Given two data sets D_1, D_2 with rulesets R_1, R_2, we want to find interesting emerging rule patterns like (3) using supports $s_1 = count_{R_1}(X) / | D_1 |, s_2 = count_{R_2}(X) / | D_2 |$ where X is a given complex pattern (equation 3), namely $X = \{(A_1, v_1),..,(A_k, v_k)\}$, for k attribute-value pairs. Note that we can use many relevant interestingness measures from the literature to compare interestingness of patterns, not only based on support, as highlighted in [4].

3 Problem Definition

Following the definitions of emerging patterns, we formulate the problem as follows: given rule sets R_i, R_j of datasets D_1, D_2, a ruleset is a series of attribute conjunctions. A subset $X \subseteq R_i$ is called a $k-rule\ set$ if $k =| X |$ is the number of attribute-value pairs in X. We need to define subsumption properties for the case where one ruleset subsumes another by one or more values from the taxonomy. A HEP is a ruleset whose support increases from one ruleset of a dataset to another. A ruleset X is a HEP from R_1 of D_1 to R_2 of D_2 if $\sigma = s_2 / s_1 > 0$. Note that $\sigma \in [0, \infty]$ and if $\sigma = \infty$, then the pattern is a jumping high-level emerging pattern (JHEP). If $\sigma = 0$, then there is no HEP, otherwise $0 < \sigma < \infty$ is merely an

$\rho - HEP$ where ρ is a threshold. A HEP pattern r_i can consist of low and high-level values from the taxonomy. There is a subsumption property such that if two HEPs exist and $r_i \subseteq r_j$, then r_i is covered by r_j i.e. r_j has one or more ancestor concept values of some or all values in r_i. There is therefore an order relation $\le$ defined on child-ancestor pairs (v_i, v'_i) as $v_i \le v'_i$ for each attribute taxonomy. Below we give subsumption properties that help to find HEPs and JHEPs. All patterns found by the AOI-HEP algorithm with properties as in section 3.1 are HEP patterns.

3.1 Subsumption Properties

P1. Total Subsumption Emerging Pattern (TSEP). We say that ruleset X is totally subsumed by ruleset Y if $x \le y, \forall x \in X, y \in Y$ and $|X| \le |Y|$. Note that this property is true for both partial orders i.e. cases where $x = y$ or $x < y$. For example, let $X = \{(a_1, v_1), (a_3, v_3), (a_4, v_4)\}$ and $Y = \{(a_1, v'_1), (a_3, v'_3), (a_4, v'_4)$ where $ancestor(v_k) = v'_k$. Then X is totally subsumed by Y. The TSEP rule condition "=" or "equality-based" may be rare to find as it is equivalent to finding exactly matching rulesets in the two datasets where as "<" or "ancestor-based" means X has some ancestors in Y . The equality condition means the same number of conjunctions and attribute-value pairs exist. We note that the ancestor subsumption property can be equivalent to finding large and frequent itemset in classical frequent itemset mining.

P2. Partial Subsumption Emerging Pattern (PSEP). We say that ruleset X is partially subsumed by ruleset Y if there exists some $y \in Y$ which is an ancestor of some $x \in X$ and $|X| < |Y|$. For example, let $X = \{(a_1, v_1), (a_3, v_3), (a_4, v_4)\}$ and $Y = \{(a_1, v_1), (a_3, v'_3), (a_4, v_4)$. There is one partial subsumption value such that $ancestor(v_3) = v'_3$ and all other values satisfy $x = y$. In addition, given $Z = \{(a_1, v_1), (a_3, v_3)\}$, then Z is partially subsumed by Y without (a_4, v_4).

P3. Overlapping Emerging Patterns (OEP). Overlapping emerging patterns occur when there are one or more common patterns between rulesets. . If we have $Y = \{(a_1, v_1), (a_3, v'_3), (a_4, v_4)$ and $Z = \{(a_1, v_1), (a_3, v_3), (a_5, v_5)\}$ then Y overlaps Z except for (a_5, v_5). The pattern (a_5, v_5) absent in Y is a jumping HEP (JHEP) from Y to Z. We call this a partially subsumed and overlapping jumping high-level emerging pattern $p_3 - JHEP$ under property P3. Conversely, the overlapping property can also be used to find diminishing patterns i.e. if suddenly the whole pattern disappears from one dataset to another.

Intuitively, both HEP and $JHEP$ can be obtained from patterns exhibiting properties P1, P2 and P3 by comparing supports using a growth function. The basic emerging pattern problem was highlighted in [1] with a growth rate given in terms of simple support i.e. $Growth-rate(X)=G(X)=\frac{\sup p_2(X)}{\sup p_1(X)}$ where X is an itemset of datasets D_1, D_2 for some threshold ρ. In our case, as we have subsumption properties between rulesets, we represent pattern X from D_1 and pattern Y from D_2 and supports s_1, s_2 respectively as defined earlier (Equation (1)). Given that emerging patterns are a function of supports s_1, s_2, the growth rate can be measured by any function $f(s_1, s_2)$. The subsumption properties hold as follows:

$$G(X,Y)=\begin{cases} 0 & if\ s_1=0, s_2=0, \approx P1, P2 \\ \infty & if\ s_1=0, s_2>0, \approx P1, P2, P3 \\ f(s_1,s_2) & otherwise \approx P1, P2 \end{cases} \tag{3}$$

Equation (3) shows that property P1 is synonymous with many classification approaches [1][10] where $f(s_1,s_2)=\frac{s_2}{s_1}$ for measuring emerging patterns when the itemsets match exactly. For rulesets, a coverage function $C(R_i, R_j)$ is used to measure how two rules compare (their similarity) or simply a measure of distance between rules as defined in [2]. This measures the number of attributes in both rules, overlapping and non-overlapping with special conditions. When coverage is determined, rulesets are paired to measure emerging ruleset patterns using the growth function f. This process is equivalent to finding large and frequent itemsets in classical frequent itemset mining and comparing them between datasets as used in emerging patterns. Note that properties P1 and P2 can lead to all three values $(0, \infty, s_2/s_1)$; P3 is used to help find jumping emerging patterns. Section 4 shows how these patterns are extracted by firstly determining the coverage of rulesets.

4 AOI-HEP Algorithm

The AOI-HEP algorithm uses **a** growth function f and a coverage measure $C(R_i, R_j)$ (similar to the distance metric in [2]) between any rulesets R_i, R_j, given N rulesets**.** The algorithm scans through characteristic rulesets mined from two datasets D_1, D_2 and puts them into relevant pairings according to their coverage using properties P1, P2 and P3 as discussed (Line 3). At Line 4, $C(R_i, R_j)$ checks rule similarity and collects different rulesets e.g. TSEP, OEP etc. After collecting k rulesets, the function f is applied to determine the degree of growth.(Line 8).

```
Input : rulesets $R_{1,} ..., R_N$ from D1, D2; threshold t, using AOI
Output: $EP = \{[ep_1, val_1],..,[ep_m, val_m]\}, Val_i \in [0,\infty], m \leq N$
 $EP$=emerging Pattern, $i = 1, j = 1$
1. $EP$ ← $\varnothing$, rulesets←$\varnothing$
2. Iterate through the rule sets by comparing $R_i, R_j$ rules
3.  WHILE NOT ($R_i == \varnothing$and $R_j == \varnothing$)and i, j <= t
4.   if $C(R_i, R_j)$ is satisfied // distance or similarity of rules
5.        $rulesets[k] \leftarrow rulesets[k] + Add(R_i, R_j)$
6. END WHILE
7. FOR  $k : 0 to \mid rulesets \mid DO$
8.       $EP = EP \cup f(ruleset_k)$//Apply growth function f
9. OUTPUT $\{EP\}$
```

The result obtained at Line 9 returns a mixture of patterns, OEP, TSEP, PSEP etc. We can apply a ranking function to order the significance of such patterns in terms of their growth-rate. Note that the growth-rate function can be more complex than use of simple support ratios as high-level rule-based emerging patterns have fewer but more complex patterns compared to single itemset patterns.

5 Experimentation

To demonstrate the effectiveness of the proposed AOI-HEP, we have processed a breast cancer Wisconsin dataset using 5 attributes that influence cancer diagnosis (699 patterns) [15], and constructed concept hierarchies for each: clump thickness, cellSize, cellShape, bareNuclei and normalNuclei. Dataset D1 has 349 tuples and D2 has 350 tuples. Some of the challenges in the experiments and the presented framework was setting an optimal threshold so that the rules and growth rates are meaningful, as with AOI. Evidently, bigger thresholds generate numerous patterns while smaller thresholds generate fewer and meaningless patterns. AOI-HEP was run with thresholds 3, 4 and 5.

Firstly, we assumed that all occurrences of the attribute value “ANY” in the output patterns were meaningless, and so we did not consider threshold 2 in that case. We did not set a growth-rate threshold but ordered all the growth-rates in descending order. Note also that values “-“ in Table 1 indicate no patterns are found. It is easy to infer from Table 1 that TSEP patterns obviously occur when the lowest or highest thresholds are used.

Table 1. Patterns from cancer datasets D1, D2 [15]

Threshold	OEP Growth%		PSEP Growth%		TSEP Growth%	
	High	Low	High	Low	High	Low
3	11.30	0.08	1.50	0.12	0.58	-
4	23.46	1.09	6.00	0.09	-	-
5	3.28	0.46	0.75	0.14	2.49	0.05

The former justifies the need to remove root node “ANY” but, in the latter, we can have numerous patterns in which we case we need to pick the strongest ones. We noted, as per property 1, that TSEP rules (i.e. exactly matching in some cases) were rarely found, but not OEP patterns. Using threshold 3 between D1 and D2, we found one large OEP pattern: Rule 3 (D1): “Clumpthickness=highriskClump AND cellSize=aboutAverage”, Rule 1 (D2): “Clumpthickness=lowriskClump AND cellSize=aboutAverage”,growth-rate (0.79/0.068)=11.30. In contrast, we found the smallest OEP pattern to be: Rule 1 (D1):“Clumpthickness=lowriskClump AND cellSize=aboutAverage AND cellShape=aboutAverage” overlapping with Rule 3 (D2): “Clumpthickness=highriskClump AND cellSize=aboutAverage”, growth-rate (0.04/0.51)=0.08. Technically, the former presents a redundancy in that clump thickness does not discriminate on the growth of the pattern. In the real world, practitioners may find this pattern of concern, noting the growing number of patients with thickening clumps despite average cancer cell sizes. In the latter, practitioners may use the least growing OEP pattern and note the role played by the differentiating attribute “cell shape = about average” between patients with high risk clump thickness and those with low risk clump thickness despite cell sizes being about average.

We observed threshold 5 for patterns. The highest OEP pattern (see Table 1) had a growth-rate of 3.28 i.e. rules R2 (D1) and R1 (D2): “clumpthickness=mediumClump AND cellSize=smallSize AND cellShape=smallShape AND bareNuclei= smallNuclei”. Higher thresholds may be useful drill-down strategies to check or further validate rules already found using lower thresholds. In this case, we may look for high impact patterns, for example the TSEP growth-rate 2.49 for threshold 5 and check whether this pattern is supported sufficiently at higher levels accordingly using some growth-rate threshold.

Further experiments have been done on the UCI repository adult dataset and similar and interesting patterns were discovered. Generally, the sequence of patterns OEP, PSEP and then TSEP in Table 1 denote their order of importance. That is OEPs are quite “frequent” as would be found in non-generalisation algorithms; TSEPs are expected in generalisation and merge approaches; and TSEPs can be rare but may reflect trends of “similar” subsumed patterns between datasets.

6 Conclusion

The paper has presented a novel framework for mining HEPs using AOI, the AOI-HEP algorithm. This framework has highlighted different aspects of mining emerging patterns by use of more complex rulesets, instead of itemsets, and their subsumption properties that translate into different types of emerging patterns. HEP patterns are particularly representative and informative in relation to large datasets and complex rulesets. Initial evaluation suggests that matching rulesets can use a general function that evaluates coverage or similarity of rules. We intend to apply the algorithm on further diverse real datasets, noting that optimal threshold choices (not too small or too big) may also influence ruleset generation and consequently growth rates. We will also extend the presented framework for mining total subsumption patterns at different hierarchical levels (including root node "ANY") by taking into account features of hierarchical data such as distances, similarity between concepts and appropriate level supports. We also note that the pattern properties presented here are well placed to handle uncertainty or fuzziness in the patterns.

References

1. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, California, United States, August 15-18 (1999)
2. Gago, P., Bentos, C.: A metric for selection of the most promising rules. In: Żytkow, J.M., Quafafou, M. (eds.) PKDD 1998. LNCS, vol. 1510, pp. 19–27. Springer, Heidelberg (1998)
3. Han, J., Cercone, N., Cai, Y.: Attribute-Oriented Induction in Relational Databases. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases, pp. 213–228 (1991)
4. Hilderman, R.J., Hammilton, H.J.: Knowledge Discovery and measures of interest. Kluwer academic, Dordrecht (2001)
5. Kamber, M., Shinghal, R.: Evaluating the interestingness of characteristic rules. In: Proceedings of the Second on Knowledge Discovery and Data Mining (KDD 1996), Portland, Oregon, USA, pp. 263–266 (1996)
6. Ramamohanarao, K., Fan, H.: Patterns Based Classifiers. In: World Wide Web, vol. 10(1), pp. 71–83 (2007)
7. García-Borroto, M., Martínez-Trinidad, J., Carrasco-Ochoa, J.A.: Fuzzy emerging patterns for classifying hard domains. Knowledge and Information Systems (2010)
8. Dong, G., Li, J.: Mining border descriptions of emerging patterns from dataset pairs. Knowledge and Information Systems 8, 178–202 (2004)
9. Fan, H., Ramamohanarao, K.: Fast Discovery and the Generalisation of Strong Jumping Emerging Patterns for building Compact and Accurate Classifiers. IEEE Transactions on Knowledge and Data Engineering 18(6), 721–737 (2006)
10. Dong, G., Li, J.: Mining border descriptions of emerging patterns from dataset pairs. Knowledge and Information Systems 8, 178–202 (2005)
11. Ceci, M., Appice, A., Malerba, D.: Emerging pattern based classification in relational data mining. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 283–296. Springer, Heidelberg (2008)
12. Agrawal, A., Srikant, R.: Mining Generalised association rules. In: VLDB (1995)
13. Qian, X., Bailey, J., Leckie, C.: Mining Generalised Emerging Patterns. In: Sattar, A., Kang, B.-h. (eds.) AI 2006. LNCS (LNAI), vol. 4304, pp. 295–304. Springer, Heidelberg (2006)
14. Chen, Y.L., Wu, Y.Y., Chang, R.I.: From data to global generalized knowledge. Journal of Knowledge and Information Systems (2010)
15. UCI Machine Learning Repository, `http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.names`

Finding First-Order Minimal Unsatisfiable Cores with a Heuristic Depth-First-Search Algorithm*

Jianmin Zhang, Weixia Xu, Jun Zhang, Shengyu Shen, Zhengbin Pang,
Tiejun Li, Jun Xia, and Sikun Li

School of Computer Science, National University of Defense Technology
410073 Changsha, China

Abstract. Explaining the causes of infeasibility of formulas has practical applications in various fields, such as artificial intelligence and formal verification. A minimal unsatisfiable core provides a succinct explanation of infeasibility and is valuable for applications. The problem of deriving minimal unsatisfiable cores from Boolean formulas has been addressed rather frequently in recent years. However little attention has been concentrated on extraction of the first-order unsatisfiable subformulas. In this paper, we present DFS-Finder, which finds minimal unsatisfiable cores in first-order logic, adopting a heuristic depth-first-search strategy. We demonstrate the effectiveness of this approach on a very extensive test of SMT-LIB benchmarks.

1 Background and Introduction

Formal verification and artificial intelligence has been based on efficient reasoning engines, such as Binary Decision Diagrams(BDD), and more recently propositional satisfiability(SAT) procedures, reasoning at Boolean level. Especially, during the last decade of impressive advances in the efficiency of SAT solvers, SAT-based method are now a fundamental technique in many industrial applications, including equivalence checking and property verification for VLSI chips, and AI planning. However the source of hardware verification problems has increasingly moving from Boolean level to higher levels: most designers work at register transfer level or even higher levels. The formalism of plain propositional logic is often not suitable or expressive enough for representing the verification of RTL or behavioral designs. The high-level structural information is identified as a suitable representation formalism for practical problems of many applications, and thus such problems more naturally expressible as satisfiability problems in first-order theories, namely Satisfiability Modulo Theories (SMT).

SMT is the problem of determining the satisfiability of a quantifier-free first-order logic formula with respect to one or more of decidable theories. The SMT solvers are able to determine whether a large formula is satisfiable or not. When

* This work is supported by the National Natural Science Foundation of China under grant No.61070132.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 178–185, 2011.

a formula is unsatisfiable, we are generally interested in a minimal explanation of infeasibility that excludes irrelevant information. Thus it is often required to find a minimal unsatisfiable core, that is, an unsatisfiable subset if it becomes satisfiable whenever any of its clauses is removed. Localizing a minimal unsatisfiable core is necessary to determine the underlying reasons for the failure, and is used in many practical applications, including AI planning[1], and model checking on predicate abstraction[2], vacancy detection[3],etc.

In the past decade, there have been considerable research works in finding Boolean unsatisfiable cores[4,5,6,7,8,9,10,11,12]. However the substantial advances in algorithms and implementations of SMT solver for years have inspired the quest of efficient solutions for the problem of unsatisfiable core extraction in SMT. Consequently, the development of effective methods for computing unsatisfiable subformulas in SMT is highlighted as an important goal for the research community. Although some SMT solvers support unsatisfiable cores generation, such as CVCLite[13], MathSAT[14] and Yices[15]. A simple and flexible algorithm [16] is the first published works on deriving unsatisfiable cores from formulas in SMT. However, as they said, one limitation of these approaches is that the resulting unsatisfiable subformula is not guaranteed to be minimal.

To the best of our knowledge, DFS-Finder is a tool firstly aiming at extracting minimal unsatisfiable cores from practical problem instances in SMT. This tool constructs the subformulas of original instance as a searching graph at first, and then recursively remove the clauses not included by the minimal unsatisfiable core from the original formula, adopting depth-first-search way. Simultaneously it uses some heuristic strategies, such as pruning methods and dynamically changing the order of variables in the subformula. Some pruning techniques are integrated into the algorithms to remove those unnecessary satisfiability checks as soon as possible, such as conflict clauses sharing and subformulas caching. To evaluate the efficiency of DFS-Finder, we implement another tool called BFS-Finder adopting the breadth-first-search algorithm, as compared with DFS-Finder. DFS-Finder and BFS-Finder are implemented in C++ [1]. An open-source SMT solver called ArgoLib[17] is integrated in the tools.

The paper is organized as follows. The next section gives the theoretical analysis and detailed description for the DFS-Finder algorithm. Section 3 shows and analyzes the experimental results on the benchmarks used by SMT solver competition of CAV 2010. Finally, Section 4 concludes the paper and outlines future research work.

2 DFS-Finder Description

DFS-Finder introduces the searching graph as an organizing framework. Firstly, the definition of searching graph of a formula in SMT is given as follows:

Definition 1. ***(Searching graph).*** *Given an unsatisfiable formula φ in SMT, if a directed acyclic graph $G(V, E, s)$ satisfies the following conditions: (a) it*

[1] The two tools are available for downloaded at http://www.ssypub.org/∼zjm/

contains only one sink node, which is on behalf of φ; (b) $\forall p \in V$, the node p represents the formula $\psi = \wedge_1^n C_i$; If v is the k-th child node of p, the node v denotes the formula $\phi_k = \wedge_1^n C_i \setminus C_k$, where $v \in V$,$1 \leq k \leq n$; e_{pv} is an edge from the parent node p to the child node v, where $e_{pv} \in E$. Then $G(V, E, s)$ is called a searching graph of φ.

Furthermore, we can classify all of the nodes of $G(V, E, s)$ into three categories: the live nodes, the dead nodes and the pending nodes. The following shows the definitions of the three types of nodes and the transition relation of the nodes.

Definition 2. *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. Suppose $v \in V$, and ϕ denoted by v, where $\phi \subseteq \varphi$. Then v is a **live node** iff ϕ is unsatisfiable; v is a **dead node** iff ϕ is satisfiable; If the search process has not reach the node v, v is called a **pending node**.*

Definition 3. *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. In $G(V, E, s)$, the **transitions** from pending nodes to dead nodes and live nodes are defined as: (a) pending nodes → dead nodes: When a subformula corresponding to a pending node is proved to be satisfiable, the pending node is changed to a dead node; (b) pending nodes → live nodes: When a subformula denoted by a pending node is unsatisfiable, the pending node is changed to a live node.*

Suppose φ is an unsatisfiable formula in SMT, and then DFS-Finder builds a searching graph $G(V, E, s)$, in which the original formula φ is represented by the sink node, and each internal node corresponds to one of the subformulas of φ. However, what is the relationship between the minimal unsatisfiable cores and the nodes of the searching graph? According to the above definitions, we may come to the following conclusions.

Conclusion 1 *Given an unsatisfiable formula φ in SMT, and $G(V, E, s)$ is the searching graph of φ. Then the subgraph, in which the sink corresponds to a dead node, cannot contain an unsatisfiable core. Further a subformula ϕ denoted by the live node v is a minimal unsatisfiable core, iff all children of v are the dead nodes, where $\phi \subseteq \varphi$ and $v \in V$.*

DFS-Finder employs the searching graph to construct the searching process. Given an unsatisfiable formula φ in SMT, if a directed acyclic graph $G(V, E, s)$ satisfies the following conditions: Firstly, it contains only one sink node, which is on behalf of φ; Secondly, $\forall p \in V \setminus \{s\}$, the node p represents the formula $\psi = \wedge_1^n C_i$; If v is the k-th child node of p, the node v denotes the formula $\phi_k = \wedge_1^n C_i \setminus C_k$, where $v \in V$,$1 \leq k \leq n$; e_{pv} is an edge from the parent node p to the child node v, where $e_{pv} \in E$. Then $G(V, E, s)$ is called a searching graph of φ.

Fig.1 provides the pseudo code of DFS-Finder. It employs an incremental way: Firstly, it computes an unsatisfiable core; Further, it derives the minimal unsatisfiable core. The function called *ComputeUS* returns an unsatisfiable core

```
**DFS_Finder**($formula$)
    $SmallUS$ = ComputeUS($formula$)
    if ($SmallUS$ == $formula$) then
        return $formula$
    else
        IsMinUS = VerifyMinimalUS($SmallUS$)
        if (IsMinUS) then
            return $SmallUS$
        else
            $MinimalUS$ = DFS_Finder($SmallUS$)
            return $MinimalUS$

**ComputeUS**($formula$)
    $ite$ = EliminateITE($formula$)
    $abs$ = AbstratExpression($ite$)
    for (arity = 0; arity < $formula$.size; arity++) do
        $interim$ = $abs$
        for (count = $formula$.size; count > 0; count−−) do
            $SmallUS$ = GraphPruning($interim$)
            $cnf$ = BooleanConversion($SmallUS$)
            IsSAT = SATSolve($cnf$)
            if (!IsSAT) then
                return $SmallUS$
        $abs$ = DynamicVarOrder($abs$)
    return $formula$
```

Fig. 1. DFS-Finder Procedure

of the input formula in SMT. After getting an unsatisfiable core, DFS-Finder judges and branches. If the returned unsatisfiable core is the input subformula named *formula*, *formula* is the derived minimal unsatisfiable core. Otherwise, according to the above conclusion, the function called *VerifyMinimalUS* is used to determine whether the unsatisfiable core *SmallUS* is minimal or not. If *SmallUS* is the derived minimal unsatisfiable core, DFS-Finder will stop; or else the approach regards *SmallUS* as the new input formula, and recursively computes the minimal unsatisfiable core in the depth-first-search way. When the order of branches changes, DFS-Finder may obtain different minimal unsatisfiable cores.

Fig.1 also shows the process of *ComputeUS*. This function firstly builds a searching graph for the input formula, and then finds a live node in the depth-first-search way. The function called *EliminateITE* is to remove the ITE terms from the formuals. Next *AbstratExpression* replaces the literals in the formula by the abstract variables. Then an unsatisfiable core is explored in the space of the searching graph. Some heuristics are integrated into the function named *GraphPruning*, which is used to prune the redundant subformulas and clauses from the subgraph, by the way of sharing the conflict clauses and caching dead nodes to avoid the unnecessary satisfiability checks. *BooleanConversion* converts the formula to a Boolean formula, and a SAT solver with DPLL procedure engages in determining satisfiability of the Boolean formula. If unsatisfiable, we get an unsatisfiable core denoted by that live node. Otherwise the current node is dead and should be abandon. The function called *DynamicVarOrder* is an effective heuristic technique. This function dynamically changes the order of variables in current subformula according to the frequency of false assignment. Finally, if the iteration is finished, the function will return the input formula itself as the derived unsatisfiable core. In Fig.1, *EliminateITE*, *AbstratExpression* and *BooleanConversion* are the functions belonging to a SMT decision procedure.

Another tool, called BFS-Finder explores a live node with all children being dead nodes in breadth-first way, but space precludes discussing its detailed process. The principles of BFS-Finder are similar to DFS-Finder. The main difference between them is the searching strategy. BFS-Finder moves horizontally on the branches of the graph. All of the same size of subformulas are firstly evaluated, and then the smaller ones are considered. DFS-Finder instead decides the satisfiability of all subformulas with the size decreasing in the same subgraph at first, and then moves to the neighborhood subgraph.

3 Experiments

To evaluate the effectiveness of DFS-Finder, we have selected the problem instances from the SMT solver competition benchmarks[18], and compared DFS-Finder and BFS-Finder on these benchmarks. The inputs are the formulas in SMT-LIB format. The experiments were conducted on a 2.5 GHz Athlon*2 machine having 2 GB memory and running the Linux operating system. The runtime is in seconds, and the value of timeout was set to 1800 seconds.

The empirical results of DFS-Finer and BFS-Finder on 15 typical formulas are listed in Table 1. Table 1 shows the number of variables (vars) and the number of clauses (clas) in each problem instance. Table 1 also gives the number of clauses (core size) in the minimal unsatisfiable core derived by the two tools. Table 1 also provides the runtime of DFS-Finer in seconds (DFS-Finer time). The last column presents the runtime of BFS-Finder in seconds (BFS-Finder time).

Fig.2 gives the results of DFS-Finder and BFS-Finder on problem instances of SMT solver competition benchmarks. This figure is a log-log scatter plot that charts the runtime of DFS-Finder along on the x-axis against the runtime of BFS-Finder on the y-axis. Each dot in this figure is denoted by the ratio of runtime of BFS-Finder and DFS-Finder.

From Table 1, we may observe the following. For all problem instances, the percentage of clauses in the minimal unsatisfiable cores is quite small, in most cases from 1% to 50%. Therefore, the minimal unsatisfiable cores can generally provide more succinct explanations of infeasibility, and is more valuable for a variety of practical applications.

In Fig.2, DFS-Finder generally outperforms the BFS-Finder, and the gap is enlarging with the size of formulas increasing, while original formulas contain more clauses, because more and more scatter dots lie above the diagonal with

Table 1. Performance results on 15 typical problem instances

Problem Instances	**vars**	**clas**	**core size**	**DFS-Finer time**	**BFS-Finder time**
bad_echos_ascend.base	58	259	11	5.18	**5.03**
sc_init_frame_gap.base	58	265	13	**5.11**	5.14
good_frame_update.induction	89	439	161	29.00	**28.74**
good_frame_update.base	89	467	311	72.81	**67.75**
windowreal-safe2-2	37	404	188	0.73	**0.68**
windowreal-safe-2	37	404	195	0.75	**0.68**
lpsat-goal-1	83	1345	17	**1.67**	1.69
lpsat-goal-2	142	2650	1283	**12.30**	17.16
lpsat-goal-3	201	3955	2548	**43.47**	68.55
windowreal-no_t_deadlock-15	219	2933	1351	**176.96**	179.53
windowreal-no_t_deadlock-16	233	3128	1441	**208.13**	236.58
windowreal-no_t_deadlock-17	247	3323	1531	**293.38**	303.32
windowreal-no_t_deadlock-18	261	3519	1622	**347.24**	391.02
windowreal-no_t_deadlock-19	275	3714	1712	**463.78**	497.97
windowreal-no_t_deadlock-20	289	3909	1802	**547.44**	633.77

the runtime increasing. Then we can arrive at a conclusion: the performance of DFS-Finder and BFS-Finder is comparable and at the same order of magnitude; While the clauses in the problem instances become more and more, DFS-Finder is much more efficient. The main causes are that BFS-Finder is more simple for implementation and performs more moves per seconds, especially for those formulas with less clauses. On the other hands, after finding an unsatisfiable core, DFS-Finder will continue to search the smaller ones along this subgraph, until

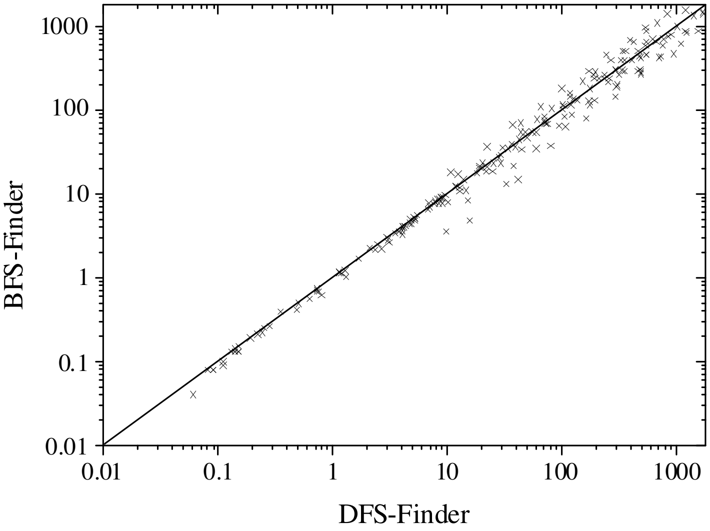


Fig. 2. Performance results on SMT solver competition benchmarks

reaching a minimal unsatisfiable core. This searching way of DFS-Finder determines that it is more and more efficient while the problem instances containing more and more clauses.

4 Conclusion

We present a tool called DFS-Finder to derive the minimal unsatisfiable cores from the formula in SMT. A very extensive test on SMT-LIB benchmarks is executed to evaluate this tool, as compared with another tool called BFS-Finder. The results show that DFS-Finder is generally outperforms the BFS-Finder, while the original formulas contain more clauses, and the gap is enlarging with the size of formulas increasing. The future works is to explore more aggressive techniques to prune the unnecessary satisfiability checks.

References

1. Jorg, H., Ri, B.: Conformant planning via heuristic forward search: A new approach. Artificial Intelligence 170, 507–541 (2006)
2. Jain, H., Kroening, D.: Word level predicate abstraction and refinement for verifying RTL Verilog. In: Proceedings of the 42nd Design Automation Conference (DAC 2005), pp. 445–450 (2005)
3. Simmonds, J., et al.: Exploiting resolution proofs to speed up LTL vacuity detection for BMC. In: Proceedings of the 7th International Conference on Formal Methods in Computer Aided Design (FMCAD 2007), pp. 3–12 (2007)
4. Oh, Y., Mneimneh, M.N., Andraus, Z.S., Sakallah, K.A., Markov, I.L.: AMUSE: a minimally-unsatisfiable subformula extractor. In: Proceedings of the 41st Design Automation Conference (DAC 2004), pp. 518–523 (2004)
5. Lynce, I., Marques-Silva, J.P.: On computing minimum unsatisfiable cores. In: Hoos, H.H., Mitchell, D.G. (eds.) SAT 2004. LNCS, vol. 3542, pp. 305–310. Springer, Heidelberg (2004)
6. Liffiton, M.H., Sakallah, K.A.: On finding all minimally unsatisfiable subformulas. In: Bacchus, F., Walsh, T. (eds.) SAT 2005. LNCS, vol. 3569, pp. 173–186. Springer, Heidelberg (2005)
7. Gershman, R., Koifman, M., Strichman, O.: Deriving small unsatisfiable cores with dominators. In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 109–122. Springer, Heidelberg (2006)
8. Gregoire, E., Mazuer, B., Piette, C.: Boosting a complete technique to find MSS and MUS thanks to a local search oracle. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007), pp. 2300–2305 (2007)
9. van Maaren, H., Wieringa, S.: Finding guaranteed mUSes fast. In: Büning, K.H., Zhao, X. (eds.) SAT 2008. LNCS, vol. 4996, pp. 291–304. Springer, Heidelberg (2008)
10. Liffiton, M.H., Mneimneh, M.N., Lynce, I., Andraus, Z.S., Marques-Silva, J.P., Sakallah, K.A.: A branch and bound algorithm for extracting smallest minimal unsatisfiable formulas. Constraints 14(4), 415–442 (2009)
11. Piette, C., Hamadi, Y., Saïs, L.: Efficient combination of decision procedures for MUS computation. In: Ghilardi, S., Sebastiani, R. (eds.) FroCoS 2009. LNCS, vol. 5749, pp. 335–349. Springer, Heidelberg (2009)

12. Nadel, A.: Boosting minimal unsatisfiable core extraction. In: Proceedings of the 10th International Conference on Formal Methods in Computer Aided Design (FM-CAD 2010), pp. 221–229 (2010)
13. Barrett, C.W., Berezin, S.: CVC Lite: A New Implementation of the Cooperating Validity Checker Category B. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, pp. 515–518. Springer, Heidelberg (2004)
14. Bozzano, M., Bruttomesso, R., Cimatti, A., Junttila, T., van Rossum, P., Schulz, S., Sebastiani, R.: An incremental and Layered Procedure for the Satisfiability of Linear Arithmetic Logic. In: Halbwachs, N., Zuck, L. (eds.) TACAS 2005. LNCS, vol. 3440, pp. 317–333. Springer, Heidelberg (2005)
15. Dutertre, B., de Moura, L.: A Fast Linear-Arithmetic Solver for DPLL(T). In: Ball, T., Jones, R.B. (eds.) CAV 2006. LNCS, vol. 4144, pp. 81–94. Springer, Heidelberg (2006)
16. Cimatti, A., Griggio, A., Sebastiani, R.: A simple and flexible way of computing small unsatisfiable cores in SAT modulo theories. In: Marques-Silva, J., Sakallah, K. (eds.) SAT 2007. LNCS, vol. 4501, pp. 334–339. Springer, Heidelberg (2007)
17. Marić, F., Janičić, P.: ARGO-LIB: A generic platform for decision procedures. In: Basin, D., Rusinowitch, M. (eds.) IJCAR 2004. LNCS (LNAI), vol. 3097, pp. 213–217. Springer, Heidelberg (2004)
18. Barret, C., Deters, M., Oliveras, A., et al.: http://www.smtcomp.org/2010/bench-marks.shtml (accessed in 2010)

3D Markerless Motion Tracking in Real-Time Using a Single Camera

Luis Quesada[1] and Alejandro León[2]

[1] Department of Computer Science and Artificial Intelligence, CITIC,
University of Granada, Granada 18071, Spain
[2] Department of Software Engineering,
University of Granada, Granada 18071, Spain
lquesada@decsai.ugr.es, aleon@ugr.es

Abstract. We present a novel motion tracking system that estimates the three-dimensional position of a moving object in real time by analyzing the image stream from a single lowest-end camera. Tracking is achieved without the need of any markers, calibration, or previous knowledge about the object. Our proposal can be applied given there is enough brightness contrast of the object with its surroundings. Such a technique allows for new Human-Computer Interaction solutions to be implemented in already running systems without a significative deployment expense.

Keywords: Motion tracking, 3D, video analysis, real-time, low budget, markerless.

1 Introduction

Motion tracking means continuously locating a moving object in a video sequence. 2D tracking aims at following the image projection of objects or parts of objects that move within a 3D space. On the other hand, 3D tracking aims at estimating all six degrees of freedom (DOFs) movements of an object relative to the camera: the three translation DOFs and the three rotation DOFs [8].

A limited 3D motion tracking technique that only estimates the three translation DOFs (namely moving up and down, moving left and right, and moving forward and backward) provides a cursor-like virtual input device that allows the interaction with a computer. This input device could be used either as a standard 2D mouse-like pointing device or, by complementing it with software that considers the depth estimation, as a novel input device that allows the development of new input paradigms (e.g. pushing or grasping in-screen objects). Support for these kinds of interactions provides a more flexible and surrounding environment for an overall more immersive user experience.

A zero development and deployment cost explotation of motion tracking systems is possible by restricting the resource usage. Particularly, imposing a requirement of only using a single lowest-end camera allows the implementation

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 186–193, 2011.

of the motion tracking system in a wide spectrum of devices, as such a hardware is found embedded in most laptops and phones.

However, using a low budget camera has several drawbacks: monocular vision; a low image resolution; high noise levels; JPEG artifacts caused by compression; a maximum framerate of 30 frames per second; and, in some cases, an automatically adjusting shutter speed that cause changes in the brightness level between consecutive frames and a maximum framerate of 10 frames per second when in low lighting conditions.

3D motion tracking techniques have direct applications in several huge niche market areas: the leisure industry, as all the popular gaming companies have presented either controller-based or controller-less motion tracking systems; the military industry, which benefits from user interfaces and object tracking; the medical industry, which requires devices for interactive simulation; the manufacturing industry, as it needs reliable robotic applications that interact with moving objects; and the software industry, as three-dimensional input devices offer new ways of interacting with software.

We present a 3D motion tracking technique that is able to determine the three-dimensional position and track the movements of shape-unknown non-rigid moving objects by analyzing the image stream from a single lowest-end camera. This system needs no calibration and the objects do not need to be marked. They just have to be opaque and evenly colored, and their brightness level should contrast enough with their surroundings. Our proposal is able to detect the most relevant object from the input image stream and ignore the background movement and partial occlusion, and provides a failback strategy that copes with tracking errors.

2 Background

Traditional 3D motion tracking approaches [1,5,7] consist in matching the geometrical features of the target object with its projection in the image. These techniques require for the object to be rigid and previously modeled, in order to know its relevant features. However, when the object surroundings are cluttered, geometric-based methods may produce wrong matches, as the best match may include background regions.

Some recent extensions to this approach [4] allow for the target object to be partially and self ocludded.

Active Appearance Model (AAM) methods [2] obtain a model of the target object and produce object reconstructions that match the object projection. These methods have to be trained with a set of labelled images before the actual system usage. As they cannot model the appearance of an object from different angles, self-occlusion or object rotation render them unusable.

Extensions to this approach [3] use view-based appearance representations. They model the appearance of the target object as a set of 2D templates corresponding to its various views and, in runtime, select the most suitable template for performing the appearance matching. The main drawbacks of these techniques are that they cannot track non-rigid objects, and that they need a very intensive training in order to model all the different views of an object.

The aforementioned techniques require a high processing time to perform the adjustment of the object reconstruction to the actual object projection, so they are not applicable in real-time tracking.

Another recent technique for object tracking is based on determining the rotation, the scale, and the translation transformations between consecutive images [9]. This technique cannot cope with several moving objects at the same time, and does not produce accurate results when there is simultaneous movement in the three axes, as different combinations of scale and translation transformations produce the same changes in the object projection.

Summarizing, existing techniques impose strong constraints as the need for the target object to be rigid or marked, having a model of the target object, or previously calibrating the system. Furthermore, most of them do not allow real-time processing. This limits the applications to controlled lab-like environments.

3 3D Markerless Motion Tracking

We introduce a motion tracking system that imposes fewer constraints on the environment. Namely, it can be applied given there is enough brightness contrast of the target object with its surroundings.

Our system takes as input a stream of frames from a camera and produces as output the position of the projection of the target object, an estimation of the projection shape and its area, and the estimated 3D space coordinates relative to the camera.

After a frame is captured, a preprocess as shown in Figure 1 is performed. First, the current frame and the previous frame are convoluted with a Poisson Disk Filter that reduces the effects of compression artifacts and noise. Then, the current frame is applied an edge detection filter based on the Sobel operator [10,6] in order to obtain a grayscale edges image. Lastly, a squared differences image of the current and the previous convoluted frames is obtained.

During the tracking procedure, the following measures of the object projection are computed: its center in the field of view, the estimation of its area, and an approximation of its shape.

Figure 2 summarizes the system behaviour in the form of a state machine.

When the system is in the ***INITIALIZING*** state, it means a one-time startup process is being performed. It forces the system to wait for two seconds while the camera shutter adjusts its speed, avoiding the flashes that may occur during this automatic setup in most cameras. After that timeout, a frame is read from the input stream and preprocessed, and the state is switched to *READY*.

When the system is in the ***READY*** state, it means it is waiting for an object to track. A new frame is read and preprocessed. If the global inter-frame movement, measured as the average of the values of all the pixels of the squared differences image, is higher than a threshold, a new object has to be tracked. The center of the object is estimated as the average position weighted by the

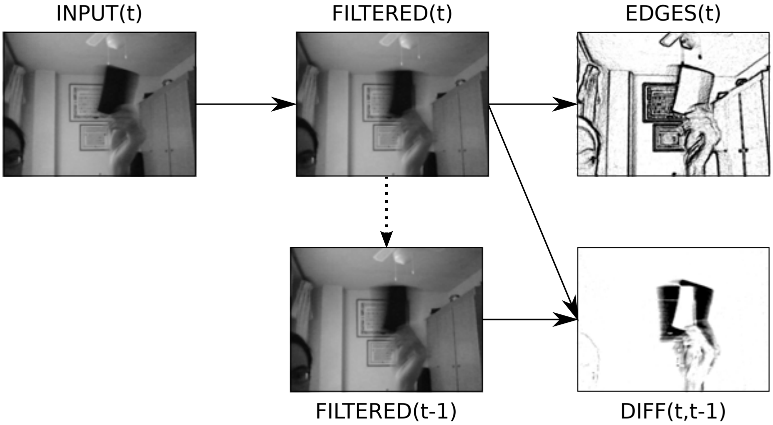


Fig. 1. Frame preprocessing

pixel magnitude of the squared differences image, and the state is switched to *TRACKING*.

When the system is in the ***TRACKING*** state, it means it is following a particular object. A new frame is read and preprocessed. The tracking is performed by repeatedly readjusting the center according to the former center position and the current edges image. 32 rays are iteratively casted from the current center position until they meet the edges that are higher than a threshold, and the new center position is calculated as the average of the ray hit locations for a maximum number of five iterations or until the last center adjustment was negligible. An example of the ray casting-based center readjustment procedure is shown in Figure 3. The system switches to the *VALIDATING* state if the tracking succeeded or to the *READY* state if it failed.

The tracking procedure fails in the following cases: if the global inter-frame movement has been below a threshold for several consecutive algorithm iterations, which means the target object has stopped moving; if the ratio of the movement near the object area against the global inter-frame movement has been below a threshold for several consecutive algorithm iterations, which means the target object might not currently be the most relevant object; or if the object area is too small or too big, which would make it difficult for the tracking algorithm to perform correctly and, indeed, may be caused by a tracking error.

It should be noted that casting 32 rays and averaging their hits locations reduces the effect of the outliers that might be caused by wrong edge detection and object occlusion.

When the system is in the ***VALIDATING*** state, it means it is checking if the new target object center found in the tracking step is consistent with the former object information. If the average color of the area around the previous center in the previous convoluted frame and the average color of the area around the new center neighborhood in the current convoluted frame differ more than a certain value or the new center is located outside of the former object area,

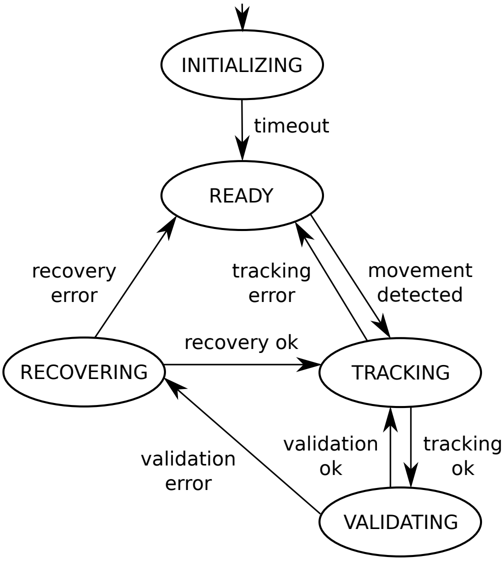


Fig. 2. System states and transitions

the state is switched to *RECOVERING*, as the center could have been wrongly repositioned. In any other case, the state is switched to *TRACKING* back again.

When the system is in the ***RECOVERING*** state, it means it is trying to relocate the center because it seems to have been located outside of the target object limits. This is done by searching the recovery point whose neighborhood average color is the most similar to the previous center neighborhood average color. The recovery points are generated by adding some horizontal, diagonal and vertical displacements to the current center position, as shown in Figure 4. If the most similarly colored area and the color average of the previous valid center neighborhood differ in no more than a predefined threshold, the recovery has succeeded and the center is repositioned in the recovery point. The center is now into the object and is readjusted by performing raycasting up to five iterations. The state is switched to *TRACKING*. If the difference surpasses the threshold, the recovery has failed and the state is switched to *READY*.

It should be noted that this failback strategy allows the tracking of fast moving objects. It also fixes tracking errors that may occur when moving the object through a similarly colored zone. If due to a wrong edge detection, the center is repositioned outside of the object and into the similarly colored zone, a recalibration will be forced, bringing it back inside the target object.

While the system is tracking an object, the output is obtained as follows: the x and y positions are obtained from the position of the center of the object projection in the image; the object shape approximation is calculated as the polygon whose vertices are the ray hit locations in the current edges image; a proportional area is computed by considering the object projection shape a rectangle, which is achieved by calculating the average width and height values

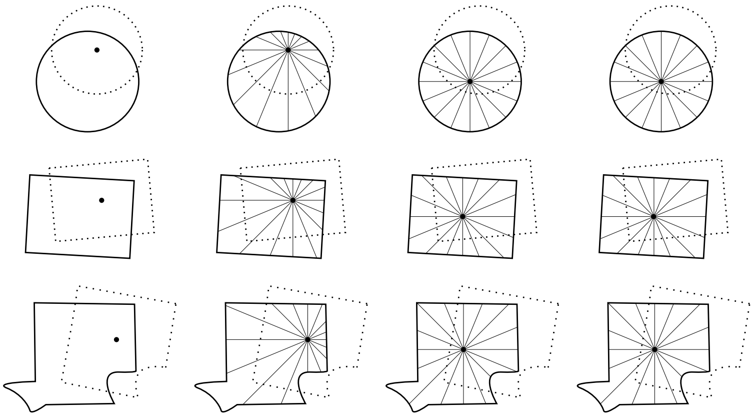


Fig. 3. Center adjustment process based on raycasting and edges detection. Example of 16 ray casting in round object projection, square object projection, and square object projection with partial oclussion and incorrect edge detection. The dotted shapes represents the former position of the object projection.

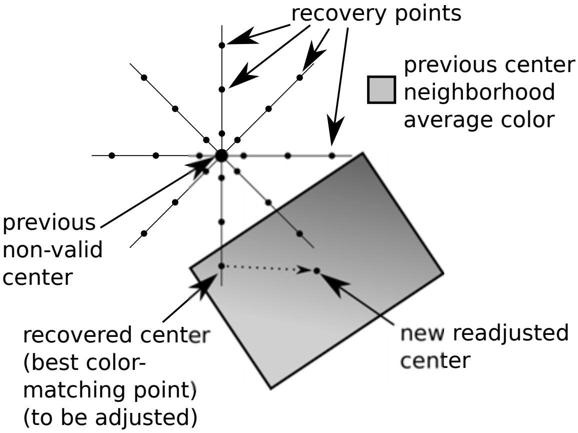

recovery points
previous center neighborhood average color
previous non-valid center
recovered center (best color-matching point) (to be adjusted)
new readjusted center

Fig. 4. Color matching-based center recovery

out of the 32 rays casted from the center of the object, as shown in Figure 5; the z or depth position is computed inversely proportional to the square root of the object area, as the bigger the area of the projection is, the closer the object is to the camera.

A smoothing of the 3D coordinates is performed to filter out possible tracking errors, raycasting outliers, and temporary losses of the object projection center. This is performed by applying a factor to the 3D coordinates so their value becomes 95% of their old value and 5% of the new value. These values provide a good balance between sensitivity and error recovery.

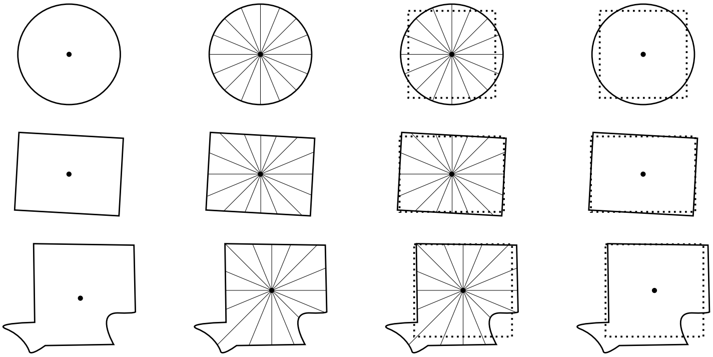


Fig. 5. Area approximation process based on rays average and edges. Example of 16 ray casting in round object projection, square object projection, and square object projection with partial oclussion and incorrect edge detection. The dotted shapes represents the former position of the object projection.

It should be noted that as no information on the real object depth is available, the depth value of 0 can be assumed to correspond to the area of the object projection when the system started tracking it.

4 Experimental Results

The proposed system was tested under several different conditions. The following results were obtained:

- The system was able to detect and track objects that were held in a hand and moved within or brought into the field of view.
- Tracking was not lost for objects that became self or partially ocludded, and their area approximation was reduced proportionally to the ocludded zone.
- If the tracking is lost, for example because the target object was moved fast enough for the center to be relocated outside of it, or because it was taken out of the field of view, the recovery strategy managed to respectively relocate the center or discard it in most cases, as it was expected.
- When the target objects were moving through low background contrast zones, the center location, the shape approximation, and the area approximation were a bit off. In good contrast conditions, they matched the object projection.
- As the output 3D coordinates are smoothed, losing the object tracking for a few frames barely affects the system behaviour.
- The system was able to obtain precise results for both 10fps or 30fps shutter speeds, although the faster the camera is, the more sharply the edges are detected, and consequently better results are obtained.
- The less ambient illumination is present, the more prone to error the system is, due to the shutter speed decrease in low illumination conditions and the increase in noise in the images.

5 Conclusions and Future Work

We have presented a 3D motion tracking technique that is able to determine the three-dimensional position and track the movements of shape-unknown non-rigid moving objects by analyzing the image stream from a single lowest-end camera.

This system needs no calibration and the objects do not need to be marked. The objects just have to be opaque and evenly colored, and their brightness level should contrast with their surroundings.

Our proposal is able to detect the most relevant object from the input image stream and ignore the background movement, and provides a failback strategy that copes with object occlusion and tracking errors.

We plan to extend the system by making it able to accurately determine changes in the orientation of unknown-shaped objects by analyzing changes in the projection shape and in the object features.

We also plan to develop new Human-Computer Interaction paradigms that use depth information.

Acknowledgements. Work partially supported by research project TIN2009-08296.

References

1. Aloimonos, J.Y., Tsakiris, D.P.: On the visual mathematics of tracking. Image and Vision Computing 9(4), 235–251 (1991)
2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active appearance models. IEEE Transactions on Pattern Analysis and Machine Intelligence 23(6), 681–685 (2001)
3. Cootes, T.F., Walker, K., Taylor, C.J.: View-based active appearance models. Image and Vision Computing 20, 657–664 (2002)
4. David, P., Dementhon, D., Duraiswami, R., Samet, H.: SoftPOSIT: simultaneous pose and correspondence determination. International Journal of Computer Vision 59(3), 259–284 (2004)
5. Gennery, D.B.: Visual tracking of known three-dimensional objects. International Journal of Computer Vision 7(3), 243–270 (1991)
6. Kanopoulus, N., Vasanthavada, N., Baker, R.L.: Design of an image edge detection filter using the sobel operator. IEEE Journal of Solid-State Circuits 23(2), 358–367 (1988)
7. Koller, D., Daniilidis, K., Nagel, H.H.: Model-based object tracking in monocular image sequences of road traffic scenes. International Journal of Computer Vision 10, 257–281 (1993)
8. Lepetit, V., Fua, P.: Monocular model-based 3D tracking of rigid objects: A survey. Foundations and Trends in Computer Graphics and Vision 1(1), 1–89 (2005)
9. Martín, J.A., Santos, M., de Lope, J.: Orthogonal variant moments features in image analysis. Information Sciences 180(6), 846–860 (2010)
10. Pratt, W.K.: Digital Image Processing, 3rd edn. (2001)

Extensions for Continuous Pattern Mining

Marcin Gorawski[1,2] and Pawel Jureczek[1]

[1] Silesian University of Technology,
Institute of Computer Science,
Akademicka 16, 44-100 Gliwice Poland
{Marcin.Gorawski,Pawel.Jureczek}@polsl.pl
[2] Wroclaw University of Technology,
Institute of Computer Science,
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
Marcin.Gorawski@pwr.wroc.pl

Abstract. In this paper we present extensions for continuous pattern mining. Our previous continuous pattern mining algorithm mines the set of all frequent sequences satisfying the minSup condition. However, those sequences contain an explosive number of frequent subsequences, which makes the analysis and understanding of patterns very difficult. In order to overcome these difficulties, we propose four new algorithms for mining maximal and closed continuous patterns. These algorithms return a superset of the result patterns and then a post-pruning algorithm is performed to eliminate redundant sequences. For each type of patterns (maximal or closed) two algorithms are presented (with and without some improvements). The key idea is to omit as many redundant sequences as possible during the exploration. The proposed algorithms allow one to reduce the size of the result set when input sequences have low uniqueness.

1 Introduction

The increasing advances in information technology and the availability of mobile navigation devices make spatio-temporal data analysis more important for many groups of users. Also the volumes of trajectory data increase. In order to effectively analyze this data it is important to develop new algorithms and ways of aggregation. To this end, we propose four new algorithms for mining closed and maximal continuous patterns of moving objects.

In the literature, we can find several approaches for mining sequential patterns. PrefixSpan [1] uses the pattern-growth paradigm and recursively projects sequence databases. SPADE [2] adopts the vertical data format and SPAM [3] utilizes the vertical bitmap representation. All of them use depth-first search. For mining maximal frequent itemsets there are GenMax [4] and FPMAX [5]. Algorithms for finding the closed itemsets include, for example, CLOSET [6] and CHARM [7], whereas for closed sequential patterns CloSpan [8] and BIDE [9]. Both CLOSET and CHARM use depth-first search. Besides, CHARM uses diffsets and CLOSET applies data structure called FP-Tree. CloSpan follows the candidate maintenance-and-test approach, i.e., it maintains the set of already found closed candidates and performs pattern closure checking. In turn, BIDE mines closed patterns without candidate maintenance. Another important

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 194–203, 2011.

structure is the WAP-Tree [10] which helps mining sequences with repeated elements. The authors of [11] extended the previous algorithm in order to explore patterns associated with mobile services.

In our recent research we have developed the CPGrowth algorithm for mining continuous sequences of regions of interest [12]. Those sequences represent trajectories. The CPGrowth algorithm is based on the aggregation tree presented in [13] and header table introduced by the authors of [14]. Our research showed that a large size of result set increases the time needed to obtain knowledge. Therefore, it was necessary to find a way to adjust results to user requirements. To solve our problems we have decided to use the maximal and closed patterns.

In this paper we present four new algorithms for mining closed and maximal continuous patterns. They use a prefix tree and employ depth-first search. The mining process consists of two main steps. In the first step, (maximal or closed) continuous candidates are retrieved and, in the second, pattern checking is performed to eliminate redundant continuous subsequences. The main idea is to omit as many redundant sequences as possible during the first step. Please notice, that proposed algorithms mines only frequent single-element sequences.

The rest of the paper is organized as follows. Section 2 gives definitions about continuous patterns. Section 3 describes new algorithms. Section 4 gives pseudocodes of the algorithms. In Section 5 experimental results are shown and in Section 6 we summarize the paper.

2 Definitions

In this section we present definitions used in the rest of paper.

Definition 1. *Definition of a continuous sequence:*

Given a set of elements $E = \{e_1, e_2, \ldots, e_n\}$, a continuous sequence is a sequence of elements $\langle a_1 a_2 \ldots a_m \rangle$, where $a_i \in E$ $(1 \le i \le m)$ and for any two elements a_i, a_j $(i \neq j)$ we have $a_i \neq a_j$.

The condition $a_i \neq a_j$ means that elements of a sequence must be unique. For instance, the continuous sequence $GBCB$ (shortened form of $\langle GBCB \rangle$) is not correct since the element B appears twice in the sequence.

Please note that an element of E represents a region of interest (RoI) and a continuous sequence represents a route traveled by a moving object.

Definition 2. *Definition of containing ($\subseteq$) one continuous sequence in other sequence:*

A continuous sequence $s_1 = \langle b_1 b_2 \ldots b_m \rangle$ is a continuous subsequence of a sequence $s_2 = \langle a_1 a_2 \ldots a_n \rangle$ $(n \ge m)$, denoted as $s_1 \subseteq s_2$, if for certain integer i $b_1 = a_i, b_2 = a_{i+1}, \ldots, b_m = a_{i+m-1}$. On the other hand, the sequence s_2 is a supersequence of s_1.

For instance, the sequence BA is a subsequence of $HJBAL$ but AB or $BALK$ is not.

Definition 3. *Definition of support:*

The continuous database, denoted as $SD = \{s_1, ..., s_m\}$, is a set of continuous sequences. $|SD|$ represents the number of all continuous sequences in SD.

The absolute support sup of a continuous sequence s is the number of continuous sequences in SD that contain s: $sup(s) = |\{s_i | s_i \in SD \wedge s \subseteq s_i\}|$.

Definition 4. *Definition of a sequence frequency:*

A continuous sequence s is called frequent if its absolute support is no less than a threshold $minSup$ given by the user: $sup(s) \geq minSup$.

A frequent continuous sequence is called a pattern p and the set of all patterns is called a pattern set $P = \{p_1, p_2, \ldots, p_n\}$.

Definition 5. *Definition of a maximal continuous pattern:*

A maximal continuous pattern m is a frequent continuous sequence that is not a continuous subsequence of any other continuous sequence.

The set of all maximal continuous patterns is called a maximal pattern set $M = \{m_1, m_2, \ldots, m_n\}$.

Definition 6. *Definition of a closed continuous pattern:*

A closed continuous pattern c is a frequent continuous sequence s and there exists no proper supersequence of s with the same support value.

The set of all closed continuous patterns is called a closed pattern set $C = \{c_1, c_2, \ldots, c_n\}$.

Based on the definitions of sets P, C and M, one can conclude that C may contain less pattern than P, and M less than C ($M \subseteq C \subseteq P$) — it depends on data.

3 Algorithms

3.1 Motivation

In earlier work we have developed the continuous pattern mining algorithm [12] named CPGrowth. Its characteristic feature is that it generate a large number of frequent sequences. However, you may notice that some patterns of the result set are contained in other patterns and not all of those patterns bring key information. Grounds are as follows. Given continuous sequential patterns A, B, C, AB, BC, ABC (some subset of a result set) and assuming that they have the same support, it can be said that some objects were moving through the same regions. This stems from the following observations: if a support of the pattern A equals a support of AB, it means that a certain number of objects passed through the region A (equal to the support) and all objects from the region A had to move to the region B, because otherwise we would not have the pattern AB. The situation is similar for the other patterns, and therefore the first 5 patterns are redundant and can be omitted. Only the pattern ABC is necessary and based on it we can restore all omitted patterns.

Now we will consider a more complex case. We have the same patterns, i.e., A, B, C, AB, BC, ABC, but patterns containing the region C have a smaller support than the others. Therefore there are two groups of patterns, one with the patters containing C and one with the rest. A support for each group of the patterns is the same. Based on the earlier explanation, the same objects had to reach the regions A and B. However, not all objects from the region B reached the region C. This could be due to the fact that some objects stopped or chosen a different direction (in such a case e.g., pattern ABD may occur in the result set). Therefore, two patterns AB and ABC should be inserted into the result set to maintain the support change. We can conclude that the use of closed patterns allow us to keep information about all changes of routes and reduce the size of a result set.

The maximal patterns in turn are determined if one wants to find all longest frequent routes over which objects are moving. Additionally, we are not interested in searching for changes in the number of objects on routes. It is only important if there is a required number of objects. Therefore, we have to bear in mind some important issues. The most important thing is that those compacted (maximal) patterns do not allow one to reconstruct the original routes along with their supports. Another issue is that the number of objects on a route may change abruptly, i.e., at the beginning of the route there may be a large number of objects, later there may be few objects (but not less than a required value), and at the end of the route even more than in the beginning - we are not able to determine that. The indisputable advantage is that it is possible to reduce a resulting set of patterns needed to be analyzed. It should be noted that when sequences differ significantly from each other the size of the result set may not be reduced as much as expected. However, the number of maximal patterns must be lower (except if the length of all maximal patterns is 1).

3.2 Building Prefix Tree

In this section we introduce the prefix tree, called EUCP-Tree (Extended UCP-Tree), that represents the sequence database SD in the compressed form. Each sequence is inserted starting from the root of the prefix tree. For instance, in Table 1 we show a sequence database consisting of 8 sequences. The database has totally 8 unique elements. The prefix tree after inserting 2, 4, 6 and 8 sequences is shown in figures 1a, 1b, 1c, 1d, respectively. The difference between the presented tree and the one in [12] is that every node has the pointer to its parent. For more details, please see [12].

3.3 Mining Maximal Patterns

In this section, we present a naive algorithm for mining maximal patterns.

CPMaxGrowth (generic version):

1. Obtain all maximal candidate from the EUCP-Tree; insert the candidates into a result set.
2. Remove from the result set all candidates that are included in other.

The obtaining of closed patterns is similar, except that in Point 2 the condition for support equality is present.

Table 1. Sequence database SD

No.	Input sequences	No.	Input sequences
1	AGFJC	5	BAGF
2	DGJ	6	BA
3	AGF	7	BAG
4	B	8	BH

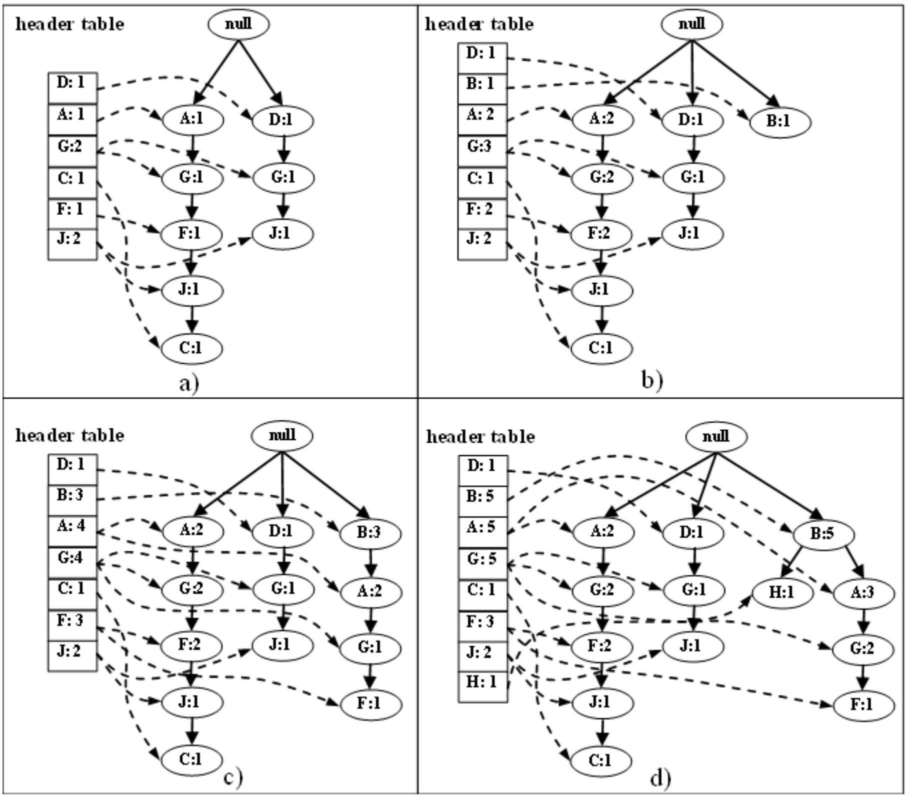


Fig. 1. Building EUCP-Tree

Figure 2 shows the initial header table along with the prefix tree (EUCP-Tree). The minimum support is set to 3 (taken as default in the examples). In the beginning we want to find all maximal patterns starting with the label A. In order to accomplish that, we find all nodes in the tree, which contain the label A — it is performed using the initial header table. Then, for those nodes their children are retrieved and a new header table is created. Next, the process is repeated recursively for each item of the newly created header table. The difference between the CPGrowth and CPMaxGrowth algorithms lies in the fact that in the latter intermediate sequences generated during the exploration of the prefix tree are not inserted into the result set. It is only done when the leaf is reached. Therefore, here, only one maximal candidate, AGF, is obtained. As a result of exploring all items in the initial header table we have the following candidates: F, GF, AGF and BA. One should notice that in the result set there are redundant sequences F and GF. In order to obtain the final result set, the candidates contained in others should be removed. The easy way is to sort all candidates in ascending order of length and remove all redundant occurrences.

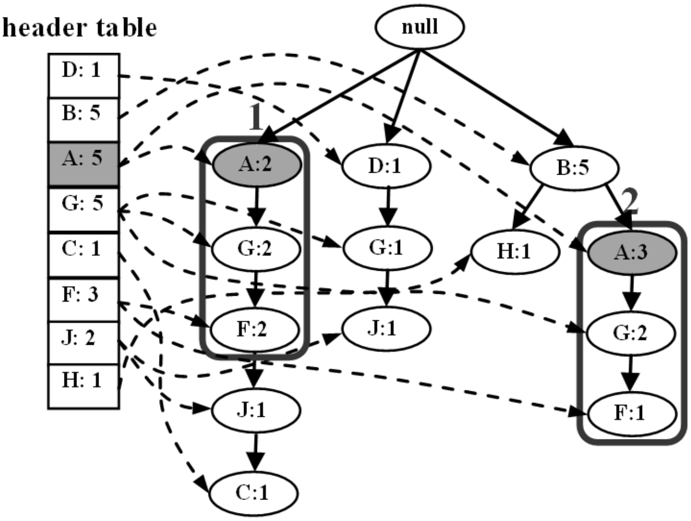


Fig. 2. Finding all maximal patterns starting with the label A

3.4 Optimizations

Having in mind that during the exploration the same nodes in the prefix tree are analyzed several times, we developed the second algorithm, called CPMaxGrowth+. This algorithm omits the exploration of those candidate sequences which are known to be contained in others. Not all redundant sequences will be avoided but the area of analysis will be limited. Please look at Figure 2 again. When we are considering the nodes with the label A, all their parents are obtained and grouped based on the label. For each group, children of the nodes with label A are retrieved. Next, it is checked if a given group is able to generate patterns. Since one of the nodes with label A has the parent with the *null* label, we need to generate all patterns that begin with A and have a successor with label G. The situation gets more interesting when generating all patterns starting with the label G. Nodes with the label G have two different parents, A and D. The parent with the label D is not considered in the analysis since it has the support below the threshold. However, the nodes with label A enable generation of the pattern GF, so the generation ends — pattern GF is generated as part of AGF. Nodes with the label F are handled in a similar manner. As a result we get patterns AGF and BA. Although we do not have sequences contained in others, we have to sort the result set and remove possible repetitions.

The above-mentioned operation is performed only for the first elements of generated patterns. Subsequent recursive calls use the CPMaxGrowth algorithm.

During our research it became clear that the second phase of maximal patterns mining (i.e., filtering redundant sequences) may take more time than the first - a total time may vary by up to two orders of magnitude. Therefore, a simple filtering of redundant patterns is not an option. In order to handle filtering effectively we use a tree-based structure. The developed tree structure is built after the first phase of exploration. The tree construction process is based on the observation: since we rely on the algorithm that generate all frequent patterns, each sequence of a given length may not be contained in other of the same length. Nevertheless, a sequence may be contained in the longer one. This observation implies the following operations. First, sequences have to be sorted in descending order of length. Then, the sorted sequences are inserted into the tree. During this process sequences are being checked whether they are contained in already inserted sequences. To achieve a significant boost we use a similar structure as for the prefix tree, i.e., we used a header table which contains lists of pointers. Each list is associated with one label.

The incremental tree has several characteristic features. The first one is that each node has only one child (because according to our observation a maximum pattern cannot be contained in other). The second feature is that an input sequence which is not contained in the current tree is a new maximal pattern. The proposed structure is not built during the first phase as it would require rebuilding the tree. (There can be inserted sequences that will be contained in later sequences, which is time costly).

Algorithm 1. The maximal pattern mining algorithm

1. Scan a sequence database *SD* once and build the EUCP-Tree.
2. Call the CPMaxGrowth+ algorithm for the EUCP-Tree.
3. Find all maximal patterns accordingly to Definition 5 (use Algorithm 4).

Algorithm 2. The CPMaxGrowth+ algorithm

1) For each item *e* stored in the header table do:

a. If the item *e* has support greater or equal to *minSup*, add *e* to the end of the prefix.

b. Create a new header table *nht* based on the list of nodes *nl*, which are associated with the item *e* of the header table (to build *nht* use Algorithm 3). If *nht* is empty, add prefix to the result set. Otherwise call the CPMaxGrowth algorithm for this table and the prefix.

Algorithm 3. The buildMaxHeaderTable+ algorithm

Input: *hl* - list of nodes for a given item of a header table

Output: *ht* - new header table

1) For each label *l* of children of *hl* count the support. Find all labels that have the support *sup* no less than *minsup* and put these labels into set *fs*.

2) For each node *nd* of *hl* do: Get the parent *pn* of *nd*. If *nd* has children, add them to the item with the label of *pn* (items are stored in a temporal header table *tht*).

3) Check if *tht* contains the root of the EUCP-Tree. If so, get the root-item *ri* from *tht* and check for each node of *ri* if its label is contained in *fs*. If so, add the label to a checking list *cl*.

4) For each item *it* of *tht* do:

a) Count *sup* for each label *l* of nodes of *it* (*sup* is only count for labels contained in *fs*).

b) For each label *l* found in the previous step do: If the support of *l* is different from the support of *l* contained in *fs*, add *l* to *cl*.

5) Create a new empty header table *ht*. For each child *e* of a node from *hl*: If the label of *e* is contained in *cl*, add *e* to *ht* (if necessary create the new item with the label). Return *ht*.

Algorithm 4. The buildResultMaxTree algorithm

```
1) Create an empty result tree $rt$ and a header table $ht$.
2) Sort candidate patterns by length. Add the longest candidate patterns to $rt$. Update $ht$.
3) For each remaining candidate $rc$ do: If $rc$ is not contained in any branch of $rt$, insert $rc$ into $rt$.
4) Get all maximal patterns from $rt$.
```

4 Pseudocodes

In this section we present a pseudocode of the CPMaxGrowth+ algorithm for mining maximal patterns. The pseudocodes of the other algorithms have been omitted because of the limited number of pages. CPMaxGrowth+ is described with the aid of Algorithms 1, 2, 3 and 4. CPMaxGrowth+ uses a EUCP-Tree whose nodes have pointers to their parents.

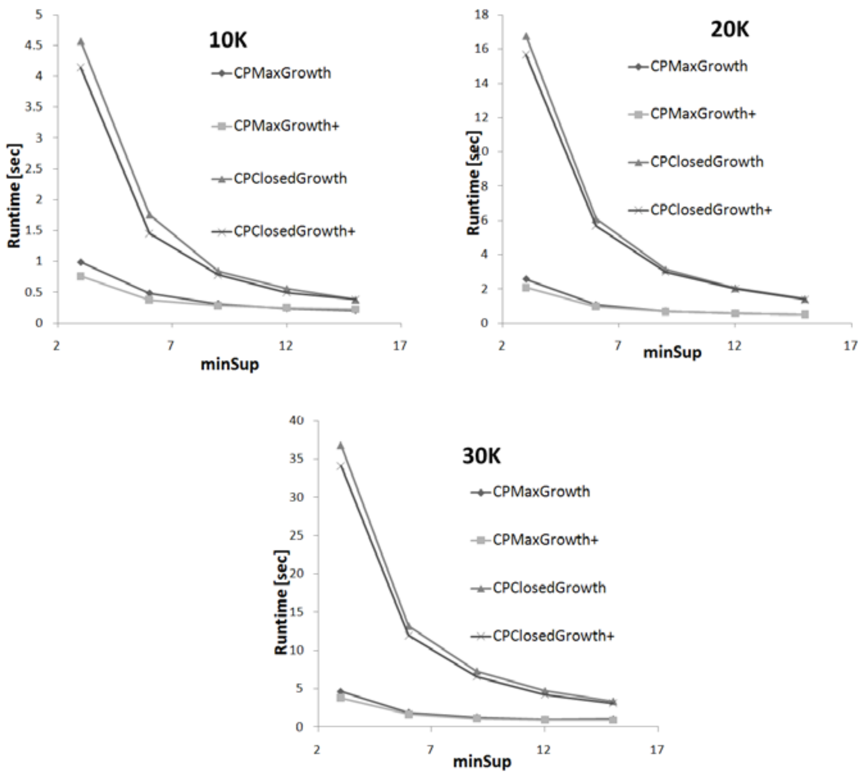


Fig. 3. Performance of algorithms

5 Experiments

During the experiments, we tested two groups of the algorithms. Each group consisted of two algorithms. The algorithms belonging to the first group do not minimize the search space of a prefix tree (CPMaxGrowth and CPClosedGrowth). On the contrary, the second group includes algorithms with optimizations (CPMaxGrowth+ and CPClosedGrowth+). The optimizations reduce the number of generated candidates.

The experiments were carried out on a computer with the following hardware configuration: Intel Core Duo E6550 2.33GHz CPU and 4GB of RAM. Trajectories were generated using the Brinkhoff generator [15]. The experiments were performed on the collection consisted of four data sets (with 10k, 20k and 30k of sequences) which were obtained as described in [12]. In the collection, sequences had an average length of about 20.03. All algorithms were implemented in Java. The experimental results are shown in Figure 3.

The presented results show that the complex algorithms (CPMaxGrowth+ and CPClosedGrowth+), in most cases, have better performance than the basic ones.

6 Conclusion

In this paper we present two algorithms for maximal pattern mining and two for closed pattern mining. These algorithms can help identifying the most useful patterns, so time needed for their analysis can be saved.

In order to improve performance, a number of improvements are applied to the complex algorithms. As shown in the Experiments section, the complex algorithms may improve the search efficiency. The main improvement is to limit the number of passes over the same parts of branches of a prefix tree, which reduces the search space.

Further work will include a detailed comparison of the proposed algorithms and existing approaches. We will also try to improve our algorithms.

References

1. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In: Proc. of the 17th Int. Conf. on Data Engineering, pp. 215–224. IEEE CS, Heidelberg (2001)
2. Zaki, M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning 42, 31–60 (2001)
3. Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential PAttern mining using a bitmap representation. In: Proc. of the 8th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 429–435. ACM, Edmonton (2002)
4. Gouda, K., Zaki, M.J.: Efficiently Mining Maximal Frequent Itemsets. In: Proc. of the 2001 IEEE Int. Conf. on Data Mining, pp. 163–170. IEEE CS, San Jose (2001)
5. Grahne, G., Zhu, J.: High performance mining of maximal frequent itemsets. In: Proc. of the Sixth SIAM Int. Workshop on High Performance Data Mining, pp. 135–143 (2003)
6. Pei, J., Han, J., Mao, R.: CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets. In: ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 21–30 (2000)

7. Zaki, M.J., Hsiao, C.-J.: CHARM: An Efficient Algorithm for Closed Itemset Mining. In: Proc. of the Second SIAM Int. Conf. on Data Mining. SIAM, Arlington (2002)
8. Yan, X., Han, J., Afshar, R.: CloSpan: Mining Closed Sequential Patterns in Large Databases. In: Proc. of the Third SIAM Int. Conf. on Data Mining. SIAM, San Francisco (2003)
9. Wang, J., Han, J.: BIDE: Efficient Mining of Frequent Closed Sequences. In: Proc. of the 20th Int. Conf. on Data Engineering, pp. 79–90. IEEE CS, Boston (2004)
10. Pei, J., Han, J., Mortazavi-Asl, B., Zhu, H.: Mining Access Patterns Efficiently from Web Logs. In: Terano, T., Chen, A.L.P. (eds.) PAKDD 2000. LNCS, vol. 1805, pp. 396–407. Springer, Heidelberg (2000)
11. Tseng, V.S., Lin, K.W.: Efficient mining and prediction of user behavior patterns in mobile web systems. Information and Software Technology 48, 357–369 (2006)
12. Gorawski, M., Jureczek, P., Gorawski, M.: Exploration of continuous sequential patterns using the CPGrowth algorithm. In: The 7-th Int. Conf. on Multimedia and Network Information Systems, pp. 165–172 (2010)
13. Spiliopoulou, M., Faulstich, L.C.: WUM: A Tool for Web Utilization Analysis. In: Atzeni, P., Mendelzon, A.O., Mecca, G. (eds.) WebDB 1998. LNCS, vol. 1590, pp. 184–203. Springer, Heidelberg (1999)
14. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Proc. of the 2000 ACM SIGMOD Int. Conf. on Management of Data, pp. 1–12, Dallas (2000)
15. Brinkhoff, T.A.: A Framework for Generating Network-Based Moving Objects. Geoinformatica, 153–180 (2002)

Logistic Fitting Method for Detecting Onset and Cessation of Tree Stem Radius Increase

Mark J. Brewer[1], Mika Sulkava[2], Harri Mäkinen[3], Mikko Korpela[2,4],
Pekka Nöjd[3], and Jaakko Hollmén[2]

[1] Biomathematics & Statistics Scotland, The Macaulay Land Use Research Institute,
Craigiebuckler, Aberdeen AB15 8QH, Scotland, UK
m.brewer@bioss.ac.uk

[2] Aalto University School of Science, Department of Information and Computer Science, P.O. Box 15400, FI-00076 Aalto, Finland
{mika.sulkava,mikko.v.korpela,jaakko.hollmen}@tkk.fi

[3] Finnish Forest Research Institute, Southern Finland Regional Unit,
P.O. Box 18, FI-01301 Vantaa, Finland
{harri.makinen,pekka.nojd}@metla.fi

[4] University of Helsinki, Department of Computer Science, P.O. Box 68, FI-00014
University of Helsinki, Finland

Abstract. Dendrometers are devices which measure the stem radius of a tree continuously. We studied the use of logistic and generalised logistic models for exploring dendrometer data and for automatically determining the onset and cessation dates of radial increase. We used data measured in two stands in southern Finland to test the performance of the models. In the detection task, the generalised logistic models performed well compared to earlier approaches. In addition, the exploratory analysis revealed distinct differences between growth patterns of trees in different calendar years.

1 Introduction

The timing and rate of wood formation during the growing season are key processes in determining the amount and properties of wood produced. Wood formation depends on genetic signalling, availability of resources, temperature, tree water and nutrient status, and the stage of ontogenic development, e.g. [9]. However, it has been difficult to link wood formation with short term fluctuations in resource availability, partly due to scarcity of direct observations of cambial dynamics and the large number of different processes influencing the growth rate. This lack of knowledge is largely due to difficulties in measuring wood formation across short intervals. Dendrometers have traditionally been used for measuring the intra-annual wood formation of trees with high precision, e.g. [7].

Changes in stem dimensions are not solely a result of wood formation; they are often caused by other processes, especially changes in stem hydration. Because of the large and frequent changes in stem radius associated with fluctuations in stem water potential, it is difficult to use dendrometer measurements to determine

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 204–211, 2011.

the onset, cessation, and rate of wood formation, i.e., radial increment due to formation of new cells, e.g. [6]. Furthermore, the definitions and approaches for identifying the onset and cessation of radial increase have varied across studies, e.g. [2,3].

We present an analysis of data from southern Finland. The data set consists of three groups of trees measured over a period of five years. Five trees have circumferences recorded for each of the whole five years (2001–2005), eight have measurements for the two years 2002–2003 and a final eight for the last two years 2004–2005. This gives 57 “tree-years” in total.

Our first aim in this paper was to explore year-to-year variation in the dendrometer data. The second aim was to objectively and automatically detect the onset and cessation of radial increase period caused by actual wood formation. We used statistical logistic and generalised logistic growth curve models for both purposes. Our previous work [4,10] concentrated on the second aim. The main methods there were cumulative sum (CUSUM) chart, Mann-Kendall test, autoregressive modelling, and linear segmentation. The results of this paper were compared with the earlier studies.

2 Methods

Stainless-steel band-dendrometers [7] were installed on each tree at a height of about 2 m. Changes in tree girth were measured at a resolution of 0.1 mm, corresponding to diameter change of about 0.03 mm. Daily values of stem circumference were calculated as arithmetic mean of values stored as one-hour averages. The circumference changes were converted to radial changes assuming a circular stem cross-section. For more details on the sites, measurement methods, and data, see [4,10]

There were two strands of statistical methodology considered. Firstly, we fitted non-linear random effect models to the combined data for all trees in order to study whether there were different growth patterns present in different calendar years. The approach [8] allowed us to consider information in the whole population of sample trees. Secondly, we fitted separate non-linear models to each tree. In this approach, the onset and cessation date for one tree were predicted without using any of the data from other trees, unlike CUSUM methods for example. This means that there were was no requirement to use cross-validation.

We used standard curves common in the statistical literature: the logistic and generalised logistic curves. The logistic curve is defined by

$$Y = \alpha + \beta \times \left(\frac{1}{1 + \exp\left[-\gamma \times (t - \delta)\right]} \right) \tag{1}$$

for stem radius Y in day t. Of the parameters, α and β represent the initial level and the growth of the radius, respectively; δ is the point of rotational symmetry (the “mid-point” of growth), and γ determines the curvature. The generalised logistic curve has an extra parameter to allow for different curvatures for onset and cessation of growth; the form we used is

$$Y = \alpha + \beta \times \left(\frac{1}{\{1 + \eta \times \exp\left[-\gamma \times (t - \delta)\right]\}^{1/\eta}} \right) \ , \tag{2}$$

including the extra shape parameter η. The generalised logistic curve has no property of rotational symmetry, so this does change slightly the interpretation of the δ parameter.

For a non-linear random effect model, we made the assumption that the parameters for each tree will be different but will come from the same family – as samples from a distribution of parameter values, assuming that the trees for which we have data represent a random sample from the population of trees. Commonly, random effect models assume Normal distributions for parameters; finite mixtures of Normal distributions have been proposed [11] as being a more flexible alternative.

We considered two possibilities for the random effect distribution for a parameter. We denote the Normal distribution with mean μ and variance σ^2 by $N\left(\mu, \sigma^2\right)$. Using β for illustration, we could define either a simpler or a more flexible model. The simpler model involved a standard random effect term such as

$$\beta_i \sim N\left(\mu_\beta, \sigma_\beta^2\right) , \quad \forall i = 1, 2, \ldots, n \ , \tag{3}$$

when there are data for n trees with a global "average" value μ_β. For a more flexible model we could instead define

$$\beta_i \sim N\left(\mu_{\beta,i}, \sigma_\beta^2\right) , \quad \forall i = 1, 2, \ldots, n \ , \tag{4}$$

where $\mu_{\beta,i}$ takes one of m_β values allowing for different subsets of values to have different mean values of the parameter. The latter model is likely to be useful if there are distinct groupings within the data set to give rise to different means. For our current case, trees were measured in different calendar years, so we chose to allow a different mean for selected parameters for different years – see the following Section for details. Of course, since any one tree in the study has measurements over several years, it is true that the assumption of independence may not be satisfied. With a much larger data set, a slightly more complex random effect model would be specified, having two "crossed" (i.e. additive) random effects: one for year, and one for tree. However, two aspects justify the use of year only in our case. Firstly, the data set is small and does not permit the use of a more complex model. Secondly, fitting an alternative model using tree as the random effect factor rather than year proved less successful, as the variation between year was considerably larger than the variation between tree; this can be explained by the fact that we are modelling *changes* in stem diameter from year to year, and there is evidently a strong effect of weather on stem growth which dwarfs between-tree variability.

For greatest flexibility and control in fitting the models we adopted a Bayesian approach and used WinBUGS [5] for analysis. This made it straightforward to consider options such as constraining the left-hand asymptote (α) and/or the year growth parameter (β) to particular values, while allowing the remaining parameters to be estimated in the usual way. We constrained the left-hand asymptotes because we had measurements for each tree for several years. We constrained α to be zero in its first year without loss of generality, and to the right-hand asymptote of the previous year in all other years. For the logistic model, we allowed year-specific means for the parameters β and γ, but not for the mid-point δ; too much freedom in the model parameters only gives rise to unstable estimation. For the generalised logistic model, we accordingly allowed year-specific means for β, γ and η.

3 Experiments and Results

3.1 Random Effect Modelling

The original data are presented in Fig. 1, showing stem radius against the day of year. It appears that most radius increase occurs between day 100 and day 200; some stems fluctuate considerably prior to increase onset. Fitting the random effects logistic model to the data produced the curves shown in Fig. 2a. The thin curves are the random effect predictions for each tree-year and the thick lines are the average curves for each of the five calendar years. These curves are not affected by the starting position at the beginning of the year. The initial levels of the curves are not important, since later years had trees which had been in the experiment for longer. Therefore, we only considered the differences in the curve shapes. In 2003, the average radius increase was more shallow than in the other years – the increase started earlier and ended later, but the total radius increase was less than in 2004. The total radius increase was the smallest in 2001.

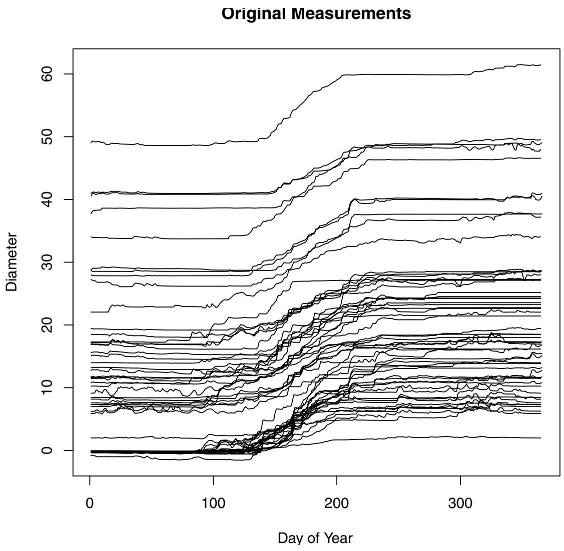
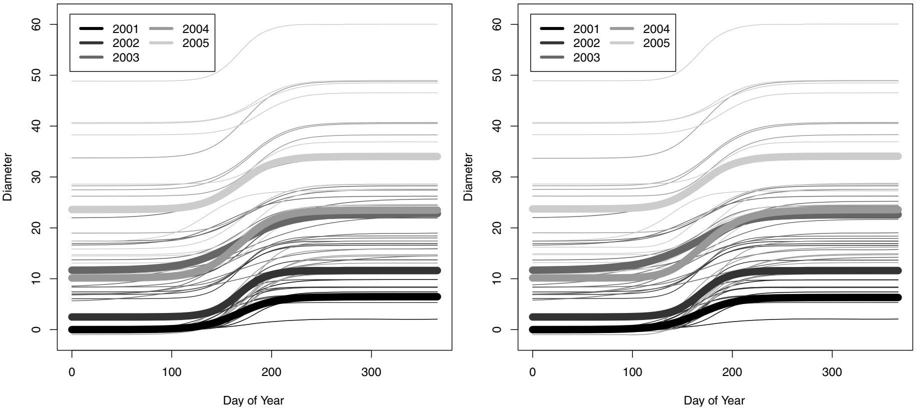


Fig. 1. Stem radius against Day of Year for the 57 tree-years



(a) Logistic models

(b) Generalised logistic models

Fig. 2. (a) Logistic and (b) generalised logistic curves for stem radius against Day of Year. The thick lines are the average curves for each of the five calendar years.

Figure 2b shows the same plot for the generalised logistic model curves. The impression is much the same as for the logistic model, but the extra curve flexibility is apparent. Compared to the logistic model, the radius increase in 2004 seems to start later. With the generalised logistic curve, there is no constraint of rotational symmetry, so the very slow ending period of radius increase has less effect on the starting period.

There were clear differences in the estimated parameters between years (Table 1). For the β parameters, year 2004 (β_4) had the largest average radius increase, and year 2001 (β_1) the least; note the non-overlapping credible intervals. The γ parameters of the logistic model show how close the point of highest curvature is to the mid-point; higher values suggest a shorter increase period. Year 2002 appeared to have the shortest period ($\gamma_2 = 0.072$), and year 2003 the longest ($\gamma_3 = 0.038$). In the generalised logistic model, parameter γ has a similar interpretation, but parameter η allows asymmetry. The large value for year 2003 ($\eta_3 = 1.583$) provided a smoother (or earlier) increase onset relative to cessation, while the small value for year 2004 ($\eta_4 = 0.433$) gave a more rapid increase onset at a later date than for the logistic model.

3.2 Growth Onset and Cessation

The random effect modelling provided us with insights about the data and increase patterns, but for prediction of onset and cessation dates we fitted separate curves for each tree. One apparent problem with the data set is that stems absorb water, thus increasing their radii, so that water absorption and growth are confounded. For this reason, we assumed that there are cut-off points in time, and fixed the start and end parameters α and β. Parameter α was set to the maximum radius recorded before the lower cut-off point, and parameter β was

Table 1. Posterion mean estimates, standard errors and 95 % credible interval estimates of random effect parameter (REP) values from fitting both logistic models

	Logistic Model			Generalised Logistic Model		
REP	Estimate	SE	95 % Interval	Estimate	SE	95 % Interval
$\mu_{\beta,1}$	6.470	1.116	(4.160,8.782)	6.320	1.098	(4.103,8.462)
$\mu_{\beta,2}$	9.138	0.624	(7.857,10.33)	9.124	0.660	(7.758,10.42)
$\mu_{\beta,3}$	11.17	0.649	(9.836,12.44)	11.00	0.654	(9.749,12.28)
$\mu_{\beta,4}$	13.34	0.660	(12.13,14.60)	13.43	0.655	(12.17,14.76)
$\mu_{\beta,5}$	10.43	0.639	(9.145,11.66)	10.36	0.660	(0.045,11.68)
$\mu_{\gamma,1}$	0.054	0.006	(0.043,0.065)	0.055	0.005	(0.045,0.065)
$\mu_{\gamma,2}$	0.072	0.004	(0.064,0.080)	0.070	0.003	(0.063,0.076)
$\mu_{\gamma,3}$	0.038	0.004	(0.031,0.045)	0.046	0.003	(0.040,0.053)
$\mu_{\gamma,4}$	0.053	0.004	(0.047,0.061)	0.043	0.003	(0.037,0.049)
$\mu_{\gamma,5}$	0.056	0.004	(0.049,0.063)	0.054	0.003	(0.048,0.060)
μ_{δ}	167.6	1.462	(164.4,170.2)	167.8	1.369	(164.8,170.3)
$\mu_{\eta,1}$				1.111	0.066	(0.984,1.244)
$\mu_{\eta,2}$				0.960	0.045	(0.873,1.048)
$\mu_{\eta,3}$				1.583	0.048	(1.491,1.677)
$\mu_{\eta,4}$				0.433	0.032	(0.366,0.493)
$\mu_{\eta,5}$				0.868	0.043	(0.789,0.955)

set to be the minimum radius recorded after the upper cut-off point. The two cut-off points have been chosen via empirical exploration (Table 2). Note that different cut-off points should be used for onset and cessation dates.

Given a fitted (generalised) logistic curve, we then define the onset or cessation points to be a particular percentage of the total estimated growth – that is, the value of parameter β for each tree. For onset date, we use 10 % growth as our yardstick for the logistic model, and 5 % growth for the generalised logistic; for cessation date, we use 98 % for the logistic and 95 % for the generalised logistic model. As with the cut-off points, these percentages were chosen by comparison of predicted values with dates subjectively determined by an expert.

Table 3 shows the root mean square errors (RMSEs) for a number of different prediction methods. For more detail on the other methods, see [4]. In the prediction of onset date, the two logistic methods were good competitors, both performing better than the baseline mean predictor, which computes the average onset or cessation date and uses it as the predicted value in a leave-one-out setting. They also both improved prediction of the cessation date compared to

Table 2. Cut-off values in days, both models

	Logistic Model		Generalised Logistic Model	
Type	Lower	Upper	Lower	Upper
Onset	125	210	125	210
Cessation	125	230	135	245

Table 3. RMSEs of predictions for onset and cessation dates by different methods

Method	Onset	Cessation
Logistic model	7.6	17.7
Generalised logistic model	6.4	16.8
Mean predictor [4,10]	8.3	18.2
CUSUM chart [4,10]	13.8	12.9
Segmentation [4]	20.5	18.6
Mann-Kendall test [4]	22.2	62.9
Autoregressive model [4]	35.6	26.5
Autoregressive model, diff. [4]	11.0	21.5

the mean predictor, excluding the CUSUM method of [4], which still appeared preferable. The generalised logistic model did slightly better than the logistic model, most likely due to the extra flexibility in curve shape.

Note that the logistic methods only consider one tree's data at a time. For this reason, there was no need to consider a cross-validated assessment. The cut-offs (Table 2) and the percentages of radius increase used to indicate growth onset and cessation were all chosen crudely – no attempt at formal optimisation took place. We preferred to have simple "rules-of-thumb" at this stage, because we had only a single expert opinion of "truth" in each case. The measures of "error" (Table 3) are only truly accurate if the expert dates are correct.

It does seem possible to improve on the prediction of cessation date for the logistic models. For the generalised logistic model, we noticed a correlation between the error (the difference between the predicted and expert cessation dates) and the estimate of parameter δ, indicating an approximate mid-growth day. This is illustrated in Fig. 3. Three different growth percentages for determining the predicted cessation date are shown. We can see that simply by applying a rule whereby we use a different growth percentage dependent on the estimated value of δ, we can reduce the RMSE considerably. Using 98% if $\delta \leq 160$, 95% if $160 < \delta \leq 170$, and 90% if $\delta > 170$ results in a RMSE of 12.6, better even than the CUSUM method. This rule, marked in the figure with non-shaded areas, is fairly ad hoc. However, it shows that more research involving methods such as Classification and Regression Trees (CART) [1] or other formal optimisation methods have the potential to produce better predictions.

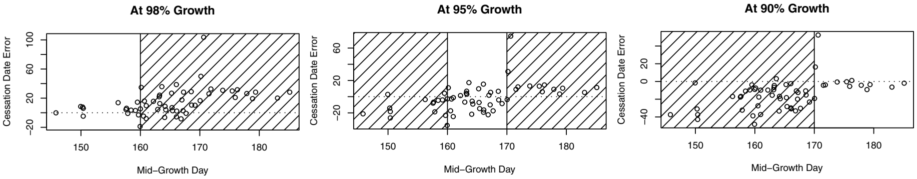


Fig. 3. Errors in cessation date against estimated mid-growth day (δ) for three percentage growths as the predicted cessation point. A rule for decreasing prediction error is to use each percentage when δ is in the corresponding non-shaded area.

4 Summary and Conclusion

The two tested logistic random effect models performed relatively well in describing annual stem radius increase. The more complex generalised logistic curve produced smaller prediction errors than the logistic model. However, other nonlinear growth curves exist and might improve fitting quality further.

Relatively high differences still remained between the dates predicted by generalised logistic curve-fitting and those determined by the expert. Ad hoc tests showed that some improved methods could be able to further improve prediction up to certain limits. However, it seems necessary to seek some other remedy, perhaps comparing the dendrometer measurements with direct measurements on tracheid formation on the stems. This will be the subject of future research.

In conclusion, the results obtained using automated methods should always be checked before using them in further analysis. The amount of manual work needed for identifying the crucial dates can however be reduced by using generalised logistic curve-fitting to assign preliminary onset and cessation labels to trees in other stands.

References

1. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and regression trees. Chapman and Hall, Boca Raton (1984)
2. Deslauriers, A., Rossi, S., Anfodillo, T.: Dendrometer and inta-annual tree growth: What kind of information can be inferred. Dendrochronologia 25, 113–124 (2007)
3. Downes, G., Beadle, C., Worledge, D.: Daily stem growth patterns in irrigated Eucalyptus globules and E. nitens in relation to climate. Trees 14, 102–111 (1999)
4. Korpela, M., Mäkinen, H., Nöjd, P., Hollmén, J., Sulkava, M.: Automatic detection of onset and cessation of tree stem radius increase using dendrometer data. Neurocomputing 73(10-12), 2039–2046 (2010)
5. Lunn, D.J., Thomas, A., Best, N., Spiegelhalter, D.J.: WinBUGS—a Bayesian modelling framework: concepts, structure and extensibility. Statistics and Computing 10, 325–337 (2000)
6. Mäkinen, H., Nöjd, P., Saranpää, P.: Seasonal changes in stem radius and production of new tracheids in Norway spruce. Tree Physiology 23, 959–968 (2003)
7. Pesonen, E., Mielikäinen, K., Mäkinen, H.: A new girth band for measuring stem diameter changes. Forestry 77, 431–439 (2004)
8. Pinheiro, J.C., Bates, D.M.: Mixed-effects models in S and SPLUS. Springer, Heidelberg (2000)
9. Savidge, R.A.: Xylogenesis, genetic and environmental regulation (review). IAWA Journal 17, 269–310 (1996)
10. Sulkava, M., Mäkinen, H., Nöjd, P., Hollmén, J.: Automatic detection of onset and cessation of tree stem radius increase using dendrometer data and CUSUM charts. In: Lendasse, A. (ed.) Proceedings of European Symposium on Time Series Prediction – ESTSP 2008, pp. 77–86. Helsinki University of Technology, Multiprint Oy/Otamedia, Porvoo, Finland (2008)
11. Verbeke, G., Lesaffre, E.: A linear mixed-effects model with heterogeneity in the random-effects population. Journal of the American Statistical Association 91, 217–221 (1996)

Simplifying SVM with Weighted LVQ Algorithm

Marcin Blachnik[1] and Mirosław Kordos[2]

[1] Silesian University of Technology, Department of Management and Informatics,
Katowice, Krasinskiego 8, Poland
marcin.blachnik@polsl.pl

[2] University of Bielsko-Biala, Department of Mathematics and Informatics,
Bielsko-Biała, Willowa 2, Poland
mkordos@ath.bielsko.pl

Abstract. Reduced Set SVMs (RS-SVM) are a group of methods that simplify the internal structure of SVM models, while keeping the SVMs' decision boundaries as similar as possible to the original ones. RS-SVMs are very useful in reducing computational complexity of the original models. They accelerate the decision process by reducing the number of support vectors. They are especially important for large datasets, when lots of support vectors are selected. They also can be very useful for understanding the internal structure of SVM models by the use of prototype-based rules. This paper presents a new method based on the modified version of the LVQ algorithm called WLVQ, which combines both of the objectives: computational complexity reduction and generation of prototype-based rules.

Keywords: SVM, Reduced set methods, LVQ, Prototype-based rules.

1 Introduction

One of the problems faced by support vector machines (SVM) is the speed of the prediction process. This problem is especially important for large datasets and online prediction problems. In such cases usually a large number of support vectors (SV) is chosen, what increases the time required to calculate the kernel matrix and consequently the system response. This problem may be overcome by reduction the number of SV and at the same time preserving original SVM's decision borders.

Reducing the number of SVs may be also beneficial to understand the data properties reflected in the decision boundaries of the SVM model. As we have shown in [1] when the number of SVs gets dramatically reduced in such a way that the prototype positions represent groups of similar instances, the SVM model can be represented as a set of Prototype Based-Rules (P-Rules) [2]. In this approach each SV is treated as a prototype and associated with its similarity or distance function.

Summarizing, there are many benefits that can be obtained from reducing the number of support vectors. There are different techniques that can be used to achieve that goal:

1. Removing some of the original SVs that are linearly dependent leaving other SVs intact [3]

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 212–219, 2011.

2. Applying the “Reduced Set" approach where new support vectors are selected or constructed anywhere in the input space (not necessarily as some of the training samples)[4,5];
3. Modifying the cost function of the original SVM [6,7].

The first approach is very useful, because it does not modify the decision boundary of the SVM classifier, but only reduces its computational complexity. This method usually allows reducing the number of support vector by few or more percents. However, this in many cases is not sufficient, so other techniques have to be used, and these two other groups are examples of the possible solutions.

The second method is based on constructing new set of SVs that is much smaller then the set obtained during the SVM training, but it preserves the shape of the decision boundary and keeps it as similar to the original one as possible. That approach would be further discussed in the next section (2). The last approach is based on reformulating the cost function of the SVM taking into account not only the width of the margin but also the number of SVs.

In this paper a new “Reduced Set" method is presented (the second approach). This method is based on the LVQ algorithm, which is modified to achieve the best possible reconstruction of the decision boundary. The discussed Weighted LVQ algorithm, embed in the prototype positions information about the shape of the decision boundary of the SVM model. The reduced set SVM (RS-SVM) method has an important advantage over other methods. One of the properties of the LVQ algorithm is a selection of prototypes which represent clusters of similar instances that preserves class labels. Such a property allows for better understanding of the model by the use of the P-rules concept of model comprehensibility.

The paper is organized as follows: the next section (2) introduces the SVM training process, and discusses the state of the art in the Reduced Set method. The section (3.1) presents the modified version of the LVQ algorithm called Weighted-LVQ (WLVQ), and an appropriate weighting procedure. The section (4) presents how to determine the appropriate number of SVs. Section (5) shows numerical examples of the new reduced set method (RS) on some artificial and real world problems. The last section (6) concludes the article and draws further research directions.

2 Simplifying SVM Prediction Model

2.1 Introduction to SVM

The SVM is a linear discrimination model defined in the feature space $\mathcal{F}$ after mapping data from the n-dimensional input space χ to this feature space $\phi(\mathbf{x})$. That can be defined as:

$$\mathbf{\Psi} = \sum_{i=1}^{m} \gamma_i \phi(\mathbf{x}_i) \tag{1}$$

According to the kernel trick, it is not necessary to directly map the data into the feature space $\mathcal{F}$ using the mapping function ($\phi(\cdot)$), but rather implicitly map the data using the property of the dot product using the kernel function. The decision function in this case is defined as:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \gamma_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{2}$$

where $\mathbf{x}_i$ are the support vectors with non-zero γ_i coefficients (Lagrangian multipliers), $K(\mathbf{x}, \mathbf{x}_i)$ is the kernel function, and $y_i = C(\mathbf{x}_i) = \pm 1$ are the class labels.

2.2 State of Art of Reduced Set Methods

The idea and the methodology of the reduced set methods was proposed by Burges in [5]. His idea is based on reducing the number of support vectors by minimizing the distance between the original SVM hyperplane $\mathbf{\Psi}$ and the hyperplane $\mathbf{\Psi}'$ obtained with the reduced set model:

$$d = \min ||\mathbf{\Psi} - \mathbf{\Psi}'||^2 \tag{3}$$

To preserve the original decision boundary the distance should be minimized so that the approximation of the new decision function $\mathbf{\Psi}'$

$$\mathbf{\Psi}' = \sum_{i=1}^{m'} \beta_i \phi(\mathbf{z}_i) \tag{4}$$

is as close to the original $\mathbf{\Psi}$ as possible, satisfying the inequality $m' \ll m$, with scalar coefficients β_i, where m' - is the reduced number of SVs.

According to the above statement, in the reduced set model the value of β_i and the positon of $\mathbf{z}_i$ have to be determined and it can be achieved by minimization of (3) over β and $\mathbf{z}$ that can be written as:

$$\begin{aligned} \min_{\beta,\mathbf{z}}(d) = \sum_{i,j=1}^{m} \gamma_i \gamma_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i,j=1}^{m'} \beta_i \beta_j K(\mathbf{z}_i, \mathbf{z}_j) \\ -2 \sum_{i=1}^{m} \sum_{j=1}^{m'} \beta_j \gamma_i K(\mathbf{x}_i, \mathbf{z}_j) \end{aligned} \tag{5}$$

As it was shown in [5], taking the matrix notation $K^{zx}\gamma = K^{zz}\beta$ where $\gamma = [\gamma_1, \gamma_2, \ldots, \gamma_m]^T$, $\beta = [\beta_1, \beta_2, \ldots, \beta_{m'}]^T$, and K^{zx} is matrix of the $m' \times m$ dimensions containing $K(\mathbf{z}_i, \mathbf{x}_j)$ values, the solution of minimization of (5) can be written as:

$$\beta = \left(K^{zz}\right)^{-1} K^{zx} \gamma \tag{6}$$

Now the goal, which is to determine the position of vectors $\mathbf{z}_j$, can be solved in two different ways. In the first solution vectors $\mathbf{z}_j$ can be selected from the vectors $\mathbf{x}_j$ using one of instance selection techniques like ENN or CNN algorithms [8] or by any other systematic search strategy. This is beneficial in terms of interpretation and comprehensibility of the solutions extracted from the SVM model by using prototype based rules. In that case each SV represents an input instance, what in many applications allows for further in depth investigation of these selected cases. On the other hand $\mathbf{z}_j$ vectors can be constructed anywhere in the input space. This approach is used by the majority of already invented algorithms. For example Schölkopf et. al. has proposed a strategy based on clustering in the feature space [4] that can be interpreted as EM iteration for the determination of the center of a Gaussian cluster representing similar vectors that match the sign of y_i and γ_i. Another approach has been proposed by Burges [5] where the author claims that the highest drop in the distance between hyperplanes d can be achieved for vectors $\mathbf{z}$ that are the eigenvectors with the highest absolute eigenvalues $\lambda = \beta \mathbf{z}^2$. Another interesting method has been proposed by Kwok and Tsang [9]. The method is based on the Multidimensional Scaling (MDS) algorithm, which transforms images of the feature space vectors back into the input space. Prototypes derived from these algorithms have no direct counterparts in the instances of the training set. However, the methods based on the construction of the new SVs enable much greater reduction of the number of the original SVs, while preserving a small distance d between hyperplanes. Unfortunately none of described prototypes construction methods allows for data understanding. The prototypes obtained by these methods are not informative, not providing any knowledge of the data structure. That deficiency may be overcome by the use of WLVQ algorithm described in this paper.

3 Reducing Number of Support Vectors with LVQ Algorithm

3.1 Weighted LVQ Algorithm

LVQ can be understood as a special case of an artificial neural network that is based on optimization of the position of codebook vectors, which are also called prototypes. The LVQ algorithm has been applied to RBF neural networks training with very good results [10], so it was also considered as a tool for reduced set methods. In the original LVQ algorithm only the distances between an instance $\mathbf{x}_i$ and the nearest prototypes $\mathbf{p}_k$ are taken into account when updating prototypes position. Therefore in [11] we have introduced a new cost function that also applies external knowledge of the data distribution:

$$\begin{aligned} E(\mathbf{P}) = & \frac{1}{2}\sum_{k=1}^{m'}\sum_{i=1}^{m} \mathbf{1}\left(\mathbf{x}_i \in R_k\right) \mathbf{1}\left(c(\mathbf{x}_i) = c(\mathbf{p}_k)\right) g(\mathbf{x}_i) \left|\left|\mathbf{x}_i - \mathbf{p}_k\right|\right|^2 \\ & -\frac{1}{2}\sum_{k=1}^{m'}\sum_{i=1}^{m} \mathbf{1}\left(\mathbf{x}_i \in R_k\right) \mathbf{1}\left(c(\mathbf{x}_i) \neq c(\mathbf{p}_k)\right) g(\mathbf{x}_i) \left|\left|\mathbf{x}_i - \mathbf{p}_k\right|\right|^2 \end{aligned} \tag{7}$$

where $\mathbf{1}(L)$ is the switching function, which returns 1 when condition L is *true*, and 0 otherwise, $c(\mathbf{x})$ returns class label of vector $\mathbf{x}$, and R_k is the Voronoi area defined for prototype $\mathbf{p}_k$. This cost function can be minimized according to $\mathbf{p}$ by iteratively updating codebook positions:

$$\begin{aligned} \mathbf{p}_k &= \mathbf{p}_k + \alpha(j) g(\mathbf{x}_i) \mathbf{1}\left(c(\mathbf{x}_i) = c(\mathbf{p}_k)\right)\left(\mathbf{x}_i - \mathbf{p}_k\right) \\ \mathbf{p}_k &= \mathbf{p}_k - \alpha(j) g(\mathbf{x}_i) \mathbf{1}\left(c(\mathbf{x}_i) \neq c(\mathbf{p}_k)\right)\left(\mathbf{x}_i - \mathbf{p}_k\right) \end{aligned} \tag{8}$$

where the context factor $g(\mathbf{x}_i)$ describes the external knowledge provided in order to achieve certain properties during the training. The $g(\mathbf{x}_i)$ value can be understood as an instance weight that describes the significance of the instance during the training process.

3.2 Determining Weights Coefficient

As described above the $g(\mathbf{x}_i)$ function can be used to introduce external dependencies that impose additional restrictions on the optimization process. Such dependencies may be defined according to the shape of the decision boundary of the SVM classifier. To achieve that aim all the vectors that are situated close to the decision boundary (2) should have higher weight values and vectors that are far from the boundary should be less significant to the optimization process. According to that we have defined $g(\mathbf{x}_i)$ as:

$$g(\mathbf{x}_i) = 1 - \left|\tanh\left(\sigma \cdot t\left(x_i\right)\right)\right| \tag{9}$$

or

$$g(\mathbf{x}_i) = \exp\left(-\sigma \cdot t\left(x_i\right)^2\right) \tag{10}$$

where σ is a user defined constant, and $t\left(x_i\right)$ is a normalized SVM decision $f\left(x_i\right)$ such that

$$t\left(x_i\right) = \begin{cases} f\left(x_i\right)/std_P & \text{if } f\left(x_i\right) > 0 \\ f\left(x_i\right)/std_N & \text{if } f\left(x_i\right) < 0 \end{cases} \tag{11}$$

where std_P and std_N are normalization factors defined as $std_P = std\left(\forall x_i : \left(f\left(x_i\right) > 0\right) x_i\right)$ and respectively $std_N = std\left(\forall x_i : \left(f\left(x_i\right) < 0\right) x_i\right)$

4 Finding Optimal Number of Support Vectors

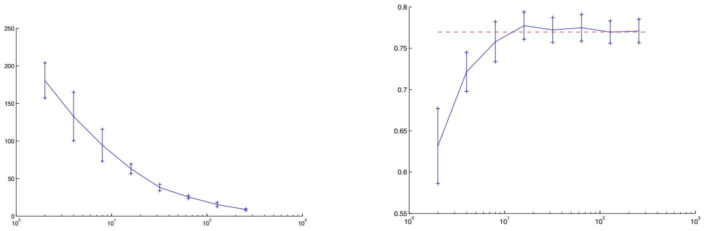
The RS-SVM approach allows for a significant reduction of the number of SVs. However, the problem of determining the correct number of reduced set of SVs remains open. The most natural solution seems to be the optimization of the distance between separating hyperplanes (3):

$$E_1(m') = \|\boldsymbol{\Psi} - \boldsymbol{\Psi}'\| = \tag{12}$$
$$\left(\sum_{i,j=1}^{m} \gamma_i\gamma_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i,j=1}^{m'} \beta_i\beta_j K(\mathbf{z}_i, \mathbf{z}_j) - 2\sum_{i=1}^{m}\sum_{j=1}^{m'} \beta_j\gamma_i K(\mathbf{x}_i, \mathbf{z}_j)\right)^2$$

When facing the comprehensibility problem of the SVM model by the use of P-rules, the cost function can be extended with an additional term $\alpha m'/m$, which represents the model complexity as a ratio of a reduced number of SVs (m') to the original number of SVs (m) multiplied by some constant α. This introduces a punishment into the cost function and promotes the solution with a smaller number of SVs. Because the distance $\|\boldsymbol{\Psi} - \boldsymbol{\Psi}'\|$ may take very high values, α may be rescaled by $1/\|\boldsymbol{\Psi} - \boldsymbol{\Psi}'_1\|$, where $\boldsymbol{\Psi}'_1$ is $\boldsymbol{\Psi}'$ defined with just one SV.

Visualization of the relation between distance $\|\boldsymbol{\Psi} - \boldsymbol{\Psi}'\|$ and the number of SVs is presented in figure (1). The same figure shows the relation between the accuracy and the number of SVs.

The analysis of results presented in figure (1).b shows that the performance of the RS model can overcome the performance of the SVM model. Such situation may happen



(a) Dependence of the cost function E_1 on the logarithm of the number of SVs

(b) Dependence of the mean accuracy (cost function E_3) on the logarithm of the number of SVs. Dashed line represents mean accuracy of the original SVM

Fig. 1. Comparison of the distance (Eq. (13)) and accuracy (Eq. (14)) based cost functions for Pima Indians diabetes data

when the SVM is over-fitted, so the RS-SVM is able to detect the over-fitting condition and propose a simpler model. This leads to two other cost functions defined as a difference between the accuracy of the SVM and RS-SVM model,

$$E_2(m') = \text{acc(SVM)} - \text{acc(RSSVM(m'))} \tag{13}$$

where acc() is the classification accuracy measured using some loss function. Because the $acc(SVM)$ remain constant during optimization of the number of SVs, the function (13) can be simplified omitting the first component:

$$E_3(m') = \text{acc(RSSVM(m'))} \tag{14}$$

5 Numerical Examples

To verify the proposed algorithm we performed a set of numerical tests on real world problems.

In the experiments, four algorithms were compared, the original SVM model (libSVM implementation), Schölkopf and Burges algorithm, both implementations were based on the Spider toolbox for Matlab, and the WLVQ algorithm also implemented as an operator of Spider toolbox `http://www.p-rules.eu`. The datasets used for the comparison were obtained from the UCI repository [12]. For that purpose the most popular datasets were selected, like *wisconsin brest cancer, heart disease, pima indiens diabetes* and *spam base*. In this experiment the number of SVs of all reduced set methods was fixed to 20. All results were obtained with a 10-fold cross-validation test. At the beginning the hyperparameters of the SVM model were optimized and after the selection of the best set of parameters the RS models were generated. The obtained results are presented in table (1)

As it can be seen, the quality of the RS-SVM method is comparable to the others. In almost all cases the Burges algorithm achieved the smallest distance d (3) between

Table 1. Empirical comparison of SVM and three reduced set methods

Dataset	SVM		Burges		Schölkopf		WLVQ	
	Accuracy	$\#SVs$	Accuracy	d	Accuracy	d	Accuracy	d
heart disease	84.50±5.54	130	82.84±5.95	0.500	82.50±5.59	14.39	84.18±4.43	17.02
diabetes	77.61±3.02	430	73.44±4.47	0.078	73.05±4.50	2.80	73.17±5.10	0.825
wbc	97.07±2.19	74	97.21±1.47	0.033	97.21±1.76	0.453	97.07±1.70	0.348
spam	93.84±1.09	734	93.31±1.12	17018	82.92±5.12	41810	90.02±1.64	39421

hyperplanes. However, this did not correlates with the best accuracy, where usually the proposed algorithm based on WLVQ networks obtained the best accuracy - the closest to the SVM model.

A very interesting results were obtained in the case when the SVM model was incorrectly optimized and tended to over-fit the data. The results are presented in table (2).

Table 2. Empirical comparison of over-fitting SVM and 3 reduced set methods

Dataset	SVM		Burges	Schölkopf	WLVQ
	Accuracy	$\#SVs$	Accuracy	Accuracy	Accuracy
heart disease	75.77±8.60	107	73.05±6.36	72.07±8.72	80.47±5.43
diabetes	73.71±5.57	332	64.98±4.03	68.23±5.47	73.70±4.77
wbc	94.15±1.68	58	95.31±2.38	93.55±3.47	95.75±2.63
spam	91.39±1.02	1550	92.00±0.80	91.70±0.84	89.95±0.99

From these results we can see that the accuracy of Burges and Schölkopf algorithms have much smaller values of distance d, however their accuracy is often much worse. That is because these two algorithms that minimize the distance d are concentrating on the most complex and bent part of the separating hyperplane $\boldsymbol{\Psi}$. This results with even a stronger over-fitting. While in the WLVQ-based model the prototypes are representing the centers of instance groups from different classes, which avoids the problem of over-fitting.

6 Conclusions

The proposed RS-SVM algorithm based on weighted LVQ algorithm has proven highly effective. In comparison to other methods the accuracy is of the same level. Similar results were obtained for the $||\boldsymbol{\Psi} - \boldsymbol{\Psi}'||$ distance (except Burges algorithm which outperforms other methods). There are also other advantages of the WLVQ-based SV. During the optimization process the LVQ codebooks are attracted to areas of high density of input vectors. This makes the reduced set of SVs meaningful and useful for understanding of relations hidden in the data, especially for the P-Rules.

An important advantage of the RS-SVM algorithm is the short time required to recalculate the RS-SVM model. In comparison to other methods codebooks position (SVs) are determined independently of the weights β_i, so the process consists of two serial subprocesses. In the first step the WLVQ algorithm is trained and then in the second step the appropriate weights β_i for each codebook are determined.

Our future plans include the investigation of the relation between the number of codebooks and the accuracy of the pure WLVQ algorithm and the $\|\mathbf{\Psi} - \mathbf{\Psi}'\|$ distance. If such relation appears, what seems to be a correct assumption, it could be interesting to apply the dynamic LVQ algorithm (DLVQ) to automatically optimize the number of SVs without recalculating the β coefficients.

Acknowledgement. The work was sponsored by the Polish Ministry of Science and Higher Education, project No. 4421/B/T02/2010/38 (N516 442138).

References

1. Blachnik, M., Duch, W.: Prototype rules from SVM. In: Rule Extraction from Support Vector Machines. Studies in Computational Intelligence Series, vol. 80, Springer, Heidelberg (2008)
2. Duch, W., Grudziński, K.: Prototype based rules - new way to understand the data. In: IEEE International Joint Conference on Neural Networks, pp. 1858–1863. IEEE Press, Washington, D.C (2001)
3. Lin, K., Lin, C.: A study on reduced support vector machines. IEEE Transactions on Neural Networks 14, 1449–1459 (2003)
4. Schölkopf, B., Knirsch, P., Smola, A., Burges, C.: Fast approximation of support vector kernel expansions. Informatik Aktuell, Mustererkennung (1998)
5. Burges, C.: Simplified support vector decision rules. In: ICML, pp. 71–77 (1996)
6. Downs, T., Gates, K., Masters, A.: Exact simplification of support vector solutions. The JMLR 2, 293–297 (2001)
7. Wu, M., Schölkopf, B., Bakur, G.: A direct method for building sparse kernel learning. The JMLR 4, 603–624 (2006)
8. Jankowski, N., Grochowski, M.: Comparison of instances seletion algorithms I. Algorithms survey. In: Rutkowski, L., Siekmann, J.H., Tadeusiewicz, R., Zadeh, L.A. (eds.) ICAISC 2004. LNCS (LNAI), vol. 3070, pp. 598–603. Springer, Heidelberg (2004)
9. Kwok, J., Tsang, I.: The pre-image problem in kernel methods. IEEE Transactions on Neural Networks 15, 408–415 (2003)
10. Palm, F.S.H.K.G.: Three learning phases for radial-basis-function networks. Neural Networks 14, 439–458 (2001)
11. Blachnik, M., Duch, W.: Improving accuracy of LVQ algorithm by instance weighting. In: Diamantaras, K., Duch, W., Iliadis, L.S. (eds.) ICANN 2010. LNCS, vol. 6354, pp. 257–266. Springer, Heidelberg (2010)
12. Merz, C., Murphy, P.: UCI repository of machine learning databases (1998-2004), http://www.ics.uci.edu/~mlearn/MLRepository.html

Novelty Detection for Identifying Deterioration in Emergency Department Patients

David A. Clifton[1], David Wong[1], Susannah Fleming[2], Sarah J. Wilson[3,4],
Rob Way[4], Richard Pullinger[4], and Lionel Tarassenko[1]

[1] Institute of Biomedical Engineering, University of Oxford, Oxford, UK
david.clifton@eng.ox.ac.uk
[2] Department of Primary Health Care, University of Oxford, Oxford, UK
[3] Heatherwood and Wexham Park Hospitals NHS Foundation Trust, Wexham, UK
[4] Oxford Radcliffe Hospitals NHS Trust, Oxford, UK

Abstract. This paper presents the preliminary results of an observational study into the use of novelty detection techniques for detecting physiological deterioration in vital-sign data acquired from Emergency Department (ED) patients. Such patients are typically in an acute condition with a significant chance of deteriorating during their stay in hospital. Existing methods for monitoring ED patients involve manual "early warning score" (EWS) systems based on heuristics in which clinicians calculate a score based on the patient vital signs. We investigate automated novelty detection methods to perform "intelligent" monitoring of the patient between manual observations, to provide early warning of patient deterioration. Analysis of the performance of classification systems for on-line novelty detection is not straightforward. We discuss the obstacles that must be considered when determining the efficacy of on-line classification systems, and propose metrics for evaluating such systems.

Keywords: Novelty Detection, Support Vector Machines.

1 Introduction

1.1 Early Warning Scores

Adverse events in acutely ill hospital patients occur when their physiological condition is not recognised or acted upon early enough [1]. Clinical guidance in the UK [2] recommends the regular observational recording of certain vital signs[1], combined with the use of EWS systems. The latter involve the clinician applying univariate scoring criteria to each vital sign in turn (e.g., "score 3 if heart rate exceeds 140 beats per minute"), and then escalating care to a higher level if any of the scores assigned to individual vital signs, or the sum of all such scores, exceed some threshold.

[1] heart rate (HR) measured in beats per minute, respiration rate (RR) measured in breaths per minute, blood oxygen saturation (SpO_2) measured as a percentage, systolic blood pressure (SysBP) measured in mmHg, etc.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 220–227, 2011.

EWS systems have a number of disadvantages. (i) The scores assigned to each vital sign, and the thresholds against which the scores are compared, are mostly determined heuristically. However, a large evidence base of vital-sign data was used to construct the EWS proposed in [3]. (ii) EWS systems are used with periodic observation of vital signs, which may be made as infrequently as once every few hours in some wards. Patients may deterioriate significantly between observations. (iii) There is a significant error-rate associated with manual scoring, especially in the high-workload setting of the ED. (iv) Each vital sign is treated independently and correlations between vital signs are not taken into account.

2 Novelty Detection

This paper takes a novelty detection approach, in which a model of "normal" patient physiology (for adult in-hospital patients) is constructed. Novelty detection is typically performed in preference to a multi-class approach to classification when there are insufficient data to model abnormal states with any accuracy.

2.1 Manual Clinical Methods

For the purposes of this study, we will consider the performance of the heuristic EWS system that was in place in the ED at the time of data acquisition, which is summarised in table 1. We will also consider the "evidence-based" EWS system described in [3], which is summarised in table 2.

Table 1. EWS system used in the ED at the time of data acquisition

Score:	3	2	1	0	1	2	3
HR		≤ 40	41 - 50	51 - 100	101 - 110	111 - 129	≥ 130
BR	≤ 8			9 - 18	19 - 24	25 - 29	≥ 30
SpO_2	≤ 92			≥ 93			
SysBP	≤ 90	91 - 99		100 - 179			≥ 180

Table 2. Evidence-based EWS system proposed in [3]

Score:	3	2	1	0	1	2	3
HR	≤ 42	43 - 49	50 - 53	54 - 104	105 - 112	113 - 127	≥ 128
BR	≤ 7	8 - 10	11 - 13	14 - 25	26 - 28	29 - 33	≥ 34
SpO_2	≤ 84	85 - 90	91 - 93	≥ 94			
SysBP	≤ 85	86 - 96	97 - 101	102 - 154	155 - 164	165 - 184	≥ 185

2.2 Automated Methods - Estimation of the Joint Density

Previous work [6] has modelled the joint pdf $f(\mathbf{x})$ of vital signs $\mathbf{x} \in \mathbb{R}^4$, for the vital signs shown in table 1. Each vital sign was standardised with respect to its own mean and variance, $x' = (x - \mu)/\sigma$. The joint distribution of the (normalised) training data was estimated using a mixture of Gaussian distributions, obtained as a Parzen window estimate with 400 components. The process used to estimate this distribution involved first summarising (using the k-means clustering algorithm) a set of approximately 2.3×10^6 data, corresponding to over 3,000 hours of vital-sign data acquired from acutely-ill hospital patients. The width parameter σ shared by all of the isotropic Gaussian distributions was set using an independent validation set [6].

The likelihood $f(\mathbf{x}|\boldsymbol{\theta})$ of previously-unseen test data $\mathbf{x}$ is then evaluated with respect to the Parzen window estimate (parameterised by $\boldsymbol{\theta}$) and used to generate a corresponding novelty score, $z(\mathbf{x}) = -\ln f(\mathbf{x}|\boldsymbol{\theta})$. This novelty score takes high values when the test data are "abnormal" with respect to f, and which thus take low probability densities as $f \to 0$.

A threshold κ is defined on z such that test data $\mathbf{x}$ are deemed "abnormal" with respect to the joint pdf if $z(\mathbf{x}) > \kappa$. In order to avoid false-positive alerts caused by transient noise and other artefact of short duration, this method only generates a novelty alert when $z(\mathbf{x}) > \kappa$ for four minutes out of any five-minute window of data. The value of κ was similarly selected using an independent validation set, selected from over 18,000 hours of vital-sign data acquired from acute patients [6].

2.3 Automated Methods - One-Class Support Vector Machine

We also consider the use of a one-class support vector machine (SVM), trained using the same data as that from which the density estimate described above was obtained. We used the method proposed by [7], in which the objective function is defined by separating the training data from the origin in the feature space defined by the SVM kernel, for which we use the Gaussian distribution[2].

The degree to which the SVM objective function is penalised by misclassifications (and thus the flatness of the decision boundary) is controlled by the C parameter, the value of which, along with the width parameter σ shared by all of the isotropic Gaussian kernels in the model, was selected using cross-validation, and was performed using the same independent validation set as was used for the density estimate described above [6].

The SVM produces a novelty score $z(\mathbf{x})$, which represents the distance between test data $\mathbf{x}$ and its decision boundary in the feature space defined by the Gaussian kernel[3]. One-class classification is performed according to the sign of z; i.e., test data $\mathbf{x}$ are classified "abnormal" if $z(\mathbf{x}) < 0$, and "normal" otherwise. To avoid false-positive alerts due to transient noise and artefact, as with the probabilistic method, an alert was generated if test data were classified "abnormal" for four minutes in any five-minute window of test data.

[2] This method typically performs similarly to the other popular one-class SVM formulation, the *support vector data description*, as proposed by [8].

[3] where that distance is normalised by the distance of the support vectors to the boundary [9].

3 Clinical Study

3.1 Overview

Vital-sign data were acquired from 472 adult patients during their stay in the ED of the John Radcliffe hospital, Oxford, using existing hospital bed-side monitors. These monitors provide measurements of HR, RR, and SpO_2 at a sampling interval of approximately 20 secs, and measurements of BP whenever the patient's blood-pressure cuff is inflated. Following [6], it was assumed that a blood-pressure measurement was valid for a period of 30 minutes after acquisition.

The total amount of data acquired from the 472 patients was approximately 1,708 hours. Patients were admitted to study (on a random basis) between January, 2009 and January, 2010, and patient consent was gained in accordance with approval from the Medical Research Ethics Committee (MREC).

3.2 Clinical Labels

To evaluate the performance of novelty detection, we would ideally have accurate labels of "normal" and "abnormal" episodes of data. The "gold standard" in classification problems is often a set of labels provided by domain experts - here, ED clinicians. However, such exhaustive labelling is typically not possible in practice due to the size of the datasets and the difficulty in determining patient abnormality from retrospective review of the vital signs. Furthermore, intra- and inter-expert variability makes the labelling process inaccurate.

An approach in which clinical experts are asked to review only vital signs from periods of suspected patient abnormality is often adopted. Here, *clinical escalations* have been taken as being indications of patient abnormality. These escalations are events that took place during the patient's stay in the ED, and they were identified retrospectively from the patient's written clinical notes.

There are many reasons for which a patient's care may be escalated in practice, only some of which will be associated with abnormal vital signs, and which could therefore be expected to be identified by an automatic method. Two clinical experts independently reviewed the patient notes and identified those periods during the patient's stay in the ED that corresponded to escalations that would be expected to be associated with abnormal vital signs. Any differences in opinion between the two experts were resolved by a third clinical expert, independently.

4 Methodology and Results

4.1 Obstacles to Evaluating Classifier Performance

The evaluation of the performance of classifiers with respect to discrete events within a time-series is a complex topic, which is reviewed in this section.

Independence. The classical method for evaluating classifier performance is to construct a confusion matrix, which quantifies the number of true and false, positive and negative classifications (TP, FP, TN, and FN) made by the classifier, with respect to the "ideal" classification. The sensitivity and specificity of the classifier may then be plotted as a function of some variable of its operation (typically a parameter that controls the decision threshold), to give the receiver operating characteristic (ROC) curve. Some EWS systems [10] have been constructed by maximising the area under this curve (AUROC). A *loss function* may be defined to assign different weights to false-positive and -negative errors if one is deemed more costly than the other. However, such weights are typically difficult to assign in practice.

The equivalent Bayesian methodology for evaluating performance in this manner is to integrate over the loss function, and select the classification parameters that minimise the expected loss ("risk") for each decision.

This approach is appropriate in the diagnostic context, as in mammography or blood chemical analysis, but are problematic when the classifier is used to analyse time-series data, as samples and events are not independent, but correlated. The results could be biased, for example, by a small number of "abnormal" patients with long hospitals stays (which is a common example, given that length-of-stay often correlates with abnormality), which contribute a large proportion of data to the set of "positives", but which are largely dependent. The performance of the classifier would be skewed towards how well it performed on this small number of patients.

We suggest that there is no simple answer to this problem, and that it is inappropriate to reduce the evaluation of classifier performance to a single metric (such as accuracy / AUROC).

Patient-Based Analysis. To use ROC-based performance metrics in time-series analysis, we must select a basic unit of analysis other than individual samples. To avoid breaking the independence assumption between basic units, we perform the analysis on a per-patient basis. In this study, we adopt the following convention:

"Event" patients: This group comprises all patients with one or more "events", the latter defined according to the criteria given in section 3 (i.e., those events with associated changes in vital signs). This corresponds to 34 (7%) of the 472 patients in our study[4].

"Normal" patients: This group comprises all patients who had no clinical escalation of any kind, and corresponds to 217 (46%) of the 472 patients in our study.

We define a TP classification to be an "event patient" for whom the first event was successfully detected; conversely, we define a FN classification to be an "event patient" for whom the first event was not successfully detected. The first event is used because the number of events from "event" patients varies between

[4] 75 (16%) of the 472 patients had no corresponding vital-sign data.

1 and 4 in our dataset. We define an event to have been detected if the method under evaluation generates an alert within some time w ahead of that event. We will consider the performance of our novelty detection systems as w is varied from [0 60] minutes, representing "early warning" up to an hour ahead of the event in the context of patient vital-sign monitoring in the ED.

We define a TN classification to be a "normal patient" for whom there were no alerts generated; conversely, we define a FP classification to be a "normal patient" for whom one or more alerts are generated by the classification system under test.

4.2 Results

Figure 1 shows the TP and FP results for candidate novelty detection methods, when evaluated on the per-patient basis described above. We compare (i) the density-based estimation method, (ii) the SVM, (iii) the heuristic EWS system that was used in the hospital at the time of the study, and (iv) the "evidenced-based" EWS system proposed in [3].

The two EWS systems (used with paper charts, in practice) were evaluated when applied to continuous data at frequencies of 30 minutes and 2 hours. The SVM and density-based methods were both trained on data obtained from a previous clinical study, as described in section 2. These training data were acquired from another set of acutely ill hospital patients. The density-based method allows the possibility of adapting its normalisation parameters (μ, σ for each vital sign) to those observed in the ED population, while still retaining the parameterisation $\boldsymbol{\theta}$ obtained from the original training set.

The results shown in the figure indicate that this "localisation" of the density-based method (whereby local population normalisation coefficients are used with an existing model) causes the performance of the model to improve (TPs increasing from 20 to 33). This increase in sensitivity is, however, matched by an increase in FPs from 34 to 66 for the original and "localised" density-based methods, respectively.

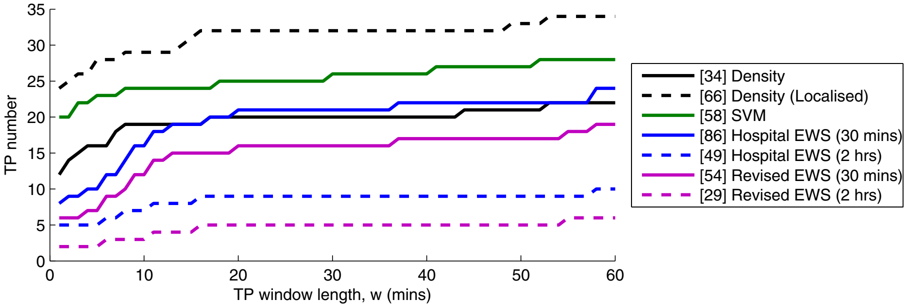


Fig. 1. TP numbers for each novelty detection method, shown as a function of w, the window-length used for determining if a physiological event was "detected". Numbers in square brackets (in the legend) indicate the number of FP classifications for each method when applied to 217 "normal" patients.

By comparison, the one-class SVM method outperforms the original density estimate, and approaches the performance of the "localised" density estimate, while having a lower FP rate, even without "localisation".

It may be seen from the figure that the manual EWS systems perform poorly in comparison with the more principled techniques described above. To match the number of TPs obtained with the density-based method, the hospital EWS system must be applied very frequently (every 30 minutes) - not a practical proposition in the clinical environment, due to the workload of clinical staff. Furthermore, at this level of TP classification, the number of FPs is particularly high (86). The FP number may be decreased by taking vital-sign observations less frequently: increasing the hospital EWS observation interval from 30 minutes to 2 hours reduces the FP number to 49, but this results in a very low sensitivity.

The effect of the "evidence-based" EWS system proposed in [3] is to significantly reduce the number of FPs in comparison to the hospital EWS system, with a small decrease in sensitivity as a result.

5 Conclusions

Analysing the performance of classifiers operating on time-series data in which discrete "abnormal events" occur is difficult. We have proposed a method to provide evaluation of classifier performance on a per-patient basis, and presented preliminary results in the context of a clinical study described in the ED.

We have demonstrated that paper-based EWS systems can be improved upon significantly by using automated methods when patients are continuously monitored with bed-side monitors (as in the high-acuity areas of the ED). We have examined the performance of density-based and one-class SVM approaches, and have shown that both provide an increase in sensitivity and specificity over existing EWS systems. Based on a training set acquired from a previous study, the SVM method outperforms the density-based method, although the latter can be improved by "localising" its normalisation coefficients to the ED population (whereas the SVM method must retain the existing normalisation coefficients of the input data, in order to retain the validity of its decision boundary in the high-dimensional feature space associated with the kernel).

It is possible that an on-line approach, which adapts to the new training data observed in the ED, would outperform both methods. Moving from a population-based to a patient-based modelling approach may also improve sensitivity to abnormalities. However, smaller quantities of training data would be available if patient-specific models were constructed, introducing significant model uncertainty, in which case a Bayesian approach is likely to be more appropriate.

Acknowledgements. The work described in this paper was funded by the NIHR Biomedical Research Centre Programme, Oxford. Dr. David Clifton was supported by the Wellcome Trust and the EPSRC under grant number WT 088877/Z/09/Z.

References

1. Safer Care for Acutely Ill Patients: Learning from Serious Accidents. Technical Report. National Patient Safety Association (2007)
2. Recognition of and Response to Acute Illness in Adults in Hospital. Technical Report. National Institute for Clinical Excellence (2007)
3. Tarassenko, L., Clifton., D.A., Pinsky, M.R., Hravnak, M.T., Woods, J.R., Watkinson, P.J.: Centile-Based Early Warning Scores Derived from Statistical Distributions of Vital Signs. Resuscitation (2011), doi:10.1016/j.resuscitation.2011.03.006
4. Williams, C.K.I., Quinn, J., McIntosh, N.: Factorial Switched Kalman Filters for Condition Monitoring in Neonatal Intensive Care. In: Advances in Neural Information Processing Systems, vol. 18, pp. 1513–1520. MIT Press, Cambridge (2006)
5. Tarassenko, L., Hann, A., Young, D.: Integrated Monitoring and Analysis for Early Warning of Patient Deterioration. Brit. J. Anaesthesia 98(1), 149–152 (2007)
6. Hann, A.: Multi-parameter Monitoring for Early Warning of Patient Deterioration. Ph.D. Thesis. University of Oxford (2008)
7. Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the Support of a High-Dimensional Distribution. Neural Computation 13, 1443–1471 (2001)
8. Tax, D.M.J., Duin, R.P.W.: Data Domain Description using Support Vectors. In: Proc. ESANN, pp. 215–256 (1999)
9. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
10. Prytherch, D.R., Smith, G.B., Schmidt, P.E., Featherstone, P.I.: ViEWS - Towards a National Early Warning Score for Detecting Adult In-Patient Deterioration. Resuscitation 81, 932–937 (2010)

A Hybrid Model to Favor the Selection of High Quality Features in High Dimensional Domains

Laura Maria Cannas, Nicoletta Dessì, and Barbara Pes

Università degli Studi di Cagliari, Dipartimento di Matematica e Informatica,
Via Ospedale 72, 09124 Cagliari, Italy
{lauramcannas,dessi,pes}@unica.it

Abstract. Feature selection is a widely recognized challenging task in dealing with application problems with a large number of features and a limited number of training samples. Filters and wrappers are the most popular feature selection strategies, but recent literature shows the emergence of hybrid approaches aiming at combining the strengths of filters and wrappers while avoiding their drawbacks. This paper proposes a new hybrid model for feature selection that takes advantage of a filter method to weight the relevance of each feature. Top-ranked features are selected, in an incremental way, resulting in a set of nested feature spaces of relatively small size. An evolutionary wrapper further refines each space by extracting small subsets of highly predictive features. Extensive experiments on a benchmark microarray dataset state the effectiveness of the proposed approach.

Keywords: Feature Selection, Hybrid Methods, Genetic Algorithms, Microarray Data.

1 Introduction

Curse of dimensionality is a major problem associated with a growing number of domains that produce data with a large number of features while the number of samples is limited. Notable examples are the microarray datasets that store thousands of gene expression profiles measured on a few dozen of samples, due to the high cost associated with the procedure and the sample availability.

Research on microarray data analysis is aimed at building an efficient model for predicting the class membership of data in order to produce a correct label on training data and predict the label for any unknown data correctly. Although the abundance of genes, it has been anticipated that only a limited number of them are informative for prognostic purposes about cancer. Thus, one demanding challenge is to identify a small subset of representative genes (features) that are potentially relevant for distinguish the sample classes.

Thanks to feature selection, classification algorithms improve their accuracy as the chance of over-fitting increases with the number of the features. Moreover, classification models based on a lower number of features are easier to understand by biologists that are interested in a limited number of informative genes.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 228–235, 2011.

Feature selection algorithms can be broadly divided in two categories: filters and wrappers [1]. Filter approaches evaluate the relevance of each single feature based on the capacity of that feature to separate the classes. Although simple and fast, filters lack robustness against correlations between features and it is not clear how to determine the optimal subset of features to be used for the purpose of classification. Conversely, wrappers compare different feature subsets and evaluate them using the classification algorithm that will be employed to build the final classifier. Being exhaustive search impractical, greedy procedures (such as forward selection or backward elimination) are usually employed to guide combinatorial search through the space of candidate feature subsets looking for a good trade-off between performance and computational cost.

During the past decade, the use of genetic algorithms (GAs) has become increasingly important [2] [3] [4] [5] as advanced type of wrapper for microarray data analysis. GAs differ from the above mentioned greedy wrappers since they select features non linearly by generating feature subsets randomly. Being efficient in detecting non linear relationships among features, GAs demonstrated their potential in exploring large feature spaces [6]. However, because the search process is guided by a stochastic model, GAs are frustrated by their instability in selecting features and face the risk to be trapped into local optimal solutions as the feature size increases. Moreover, as many wrapper selection strategies, GAs can suffer from over-fitting [1].

In this paper we present and test a hybrid model for feature selection that would overcome these limitations and be better able to explore the vast solution space. It takes advantage of both filter and wrapper approaches that, as recent literature witnesses [1] [7], can successfully complement each other.

In detail, a filter is first used to rank the initial set of features. Then, features with the highest ranking values are selected, in an incremental way, resulting in a set of nested feature spaces of relatively small size. This ensures that useful features are unlikely to be screened out but, differently from most filter-based approaches, the ranked feature list is not cut off according to a single (somewhat arbitrary) threshold value. Since the features in a filtered space can be highly correlated with each other, resulting in a non-optimal predictive performance, our model considers refining the selection process by employing a GA that searches for optimal feature subsets in the filtered solution space. For assessing each candidate solution, a classifier is used within the GA-based wrapper. A distinctive aspect of the approach is the parallel exploration of different feature spaces looking for solutions that are not necessarily alternative but complementary from a biological point of view.

As a test-bed for evaluating the proposed model we choose the Colon Tumor dataset [8], which is recognized as one of the noisiest microarray benchmarks. The experimental results compare well with other state-of-art methods and state the effectiveness of our hybrid approach in obtaining a trade-off between classification accuracy and computational cost.

The rest of the paper is structured as follows. Section 2 details the proposed model. The experimental analysis is presented in Section 3, while Section 4 contains a discussion and some concluding remarks.

2 The Proposed Model

Hybrid feature selection strategies basically involve (i) filtering the initial dataset to define a space of potentially informative features, and then (ii) refining the selection process by a wrapper method that optimizes classification accuracy. In this study, we propose a new hybrid model featured by:

a) using a filtering algorithm to define multiple feature spaces;
b) exploring each feature space through a GA that acts as wrapper selector;
c) fusing information from the different spaces to identify the features most relevant to the classification task at hand.

In more detail, the proposed approach involves decomposing the overall feature selection process into different steps that are described in what follows.

First, a particular ranking criterion (of statistical or entropic nature) is applied to assess individual features and assign them weights according to their relevance to the target class. A feature with a high ranking value indicates higher discrimination of this feature compared to other categories and means that the feature contains information potentially useful for classification. The ranking process results in an ordered list where features appear in descending order of relevance. As there is little theoretical support, it is crucial to establish a suitable threshold to cut-off the ranked list and retain only the informative features. This choice is somewhat arbitrary in the common practice [1].

In the spirit of Structural Risk Minimization [9], the ranking can be used to define nested subsets of features, and select an optimum subset with a model selection criterion by varying a single parameter: the number of features. This is the basis for a number of greedy algorithms inspired to forward selection or backward elimination search strategies. In this study, we also build a set of nested subsets, called feature spaces (FSs), but we use them as a starting point for a more advanced search and not just to train and test a classifier.

Specifically, our algorithm uses the ranked list to generate a sequence of M nested feature spaces whose size is incremented, in an iterative way, of a fixed amount k, namely: $FS_k \subset FS_{2*k} \subset \ldots \subset FS_{M*k}$. The first space, FS_k, includes the first k top-ranked features and the last one, FS_{M*k}, includes the first M*k features. A wrapper approach is then applied to further explore each space looking for solutions, i.e. feature subsets, that are optimal in terms of classification performance. This optimization is performed by a GA that extracts different subsets from the given input space and evaluates them using a classification algorithm.

In more detail, at the start of the GA, a population of individuals (i.e. feature subsets) is initialized randomly. Each individual is encoded by a N-bit binary vector, where N is the size of the provided feature space. The bit value {1} represents a selected feature, whereas the bit value {0} represents a non-selected feature; any number of features smaller than N can be selected. First, each individual is evaluated in terms of a fitness function that we assume as the accuracy of a classifier learnt on that individual (more complicated multi-objective fitness functions don't guarantee better results in this specific domain, as witnessed by our previous research [5]). Then, the current population undergoes genetic operations, i.e. selection, mutation and crossover, resulting in a new population whose individuals are again evaluated in terms of classification accuracy. This evolution process is repeated until a pre-defined number of generations is reached, and the output is a "best individual" representing the most predictive feature subset.

The above genetic search is performed separately on the M feature spaces FS_{i*k}, i = 1, …, M, as the subsets selected from such spaces may provide different ways to combine features into useful predictors. Besides, given the stochastic nature of the GA, different runs of the algorithm may select different subsets even from the same space, thus resulting in a potentially high number of distinct predictors. Hence, after a number T of trials on each space, our approach involves computing the frequency of membership of features in the resulting M*T predictors, among which there may be a certain number of replicates. This enables to evaluate the relative importance of the selected features, distinguishing among the features that play a primary role in discriminating the target class and the features that give a complementary, yet not negligible, contribution.

3 Experimental Analysis

We verified the proposed model with Colon [8] which is a popular public microarray dataset containing 62 samples, belonging to tumor and normal colon tissues, and 2000 genes.

Our model is quite general and its evaluation can be supported by a variety of filter methods as well as classification techniques. Based on our previous experience [5],[10],[11], we choose χ^2 as filter metric and SVM and K-NN (with K=1) classifiers for fitness evaluation. As parameters for building the nested feature spaces FS_{i*k} (i = 1, 2, ..., M) we set k =10 and M = 5 in that we consider the first 50 top-ranked genes. For the purpose of simplicity, we denote the first space FS_k as TOP10 (i.e. the first 10 top-ranked features), the second FS_{2*k} as TOP20 (i.e. the first 20 top-ranked features) etc. In this study, we also evaluated two additional feature spaces: TOP80 and TOP100.

The overall analysis was implemented using the Weka data mining environment [12], whose libraries provide support for ranking genes, as well as for genetic search and classification. In particular, genetic operations were carried out by roulette wheel selection, single point crossover, and bit-flip mutation. Leveraging on our previous studies [10][11] where we performed a tuning of the GA parameters, we set the following values: population size = 30, number of generations = 50, probability of crossover = 1, probability of mutation = 0.02. In fitness evaluation, error estimation was performed by a 10-fold cross-validation for both SVM and K-NN classifiers.

The experiments were organized in the following classes:

1. *Baseline experiments.* To get an evaluation of classification accuracy without considering the proposed model, each classifier (i.e. K-NN and SVM) was trained directly on each TOPN. The related accuracy was also estimated by a 10-fold cross-validation and was considered as a baseline accuracy.
2. *GA experiments.* We applied the proposed model to each TOPN and evaluated the fitness by SVM (namely, GA/SVM experiments) and by K-NN (namely, GA/K-NN experiments).

The behavior of both GA/SVM and GA/K-NN was evaluated considering three aspects: (i) classification accuracy, (ii) dimensionality of the selected subset and (iii) computational cost. Since the GA performs a stochastic search, we considered the average results over 10 trials.

Results given in Table 1 show that, for each feature space, the proposed GA/SVM outperforms in terms of accuracy the corresponding baseline SVM, with an increment of between 2,5% and 9%. With regard to the average size of the selected subsets, results show the effectiveness of the proposed approach in reducing the dimensionality of the provided feature space. In particular, GA/SVM selects the smallest subsets from TOP10 and, on average, the selected subsets becomes larger as the size of the search space increases. As well, the computational cost increases as more features are included in the search space.

Table 1. SVM: Baseline vs. GA/SVM

	Baseline accuracy (%)	Average accuracy (%)	Average subset size	Average time (sec)
Top10	82,3	87,1	4,0	334,7
Top20	88,7	90,9	6,9	1241,8
Top30	87,1	90,5	8,4	1943,9
Top40	85,5	91,5	13,9	2475,2
Top50	83,9	91,5	15,0	2919,7
Top80	85,5	91,9	23,5	3274,3
Top100	87,1	93,1	32,6	3507,9

As Table 2 shows, GA/K-NN also outperforms the corresponding baseline K-NN, with an increment of between 12,5% and 19,7%. Moreover, it turns out to be more effective in selecting predictive gene subsets, outclassing in every feature space the accuracy values obtained using GA/SVM. While the GA/K-NN average subset size is comparable with results in Table 1, GA/K-NN outperforms significantly GA/SVM in terms of computational cost.

Table 2. K-NN: Baseline vs. GA/K-NN

	Baseline accuracy (%)	Average accuracy (%)	Average subset size	Average time (sec)
Top10	80,6	90,8	5,2	3,1
Top20	82,3	95,3	5,9	9,9
Top30	83,9	94,5	9,2	20,0
Top40	83,9	94,5	12,4	31,1
Top50	80,6	92,7	13,0	37,1
Top80	79,0	94,6	22,5	54,8
Top100	79,0	93,2	25,2	68,3

The above experiments allow to evaluate how the GA-based feature selection is affected by the size and the composition of the provided search space, that is also a neglected issue in recent microarray works based on genetic algorithms [2] [3] [4]. As shown in Tables 1 and 2, the choice of parameters k and M affects in a different way the performance of GA/SVM and GA/K-NN.

In fact, GA/SVM average accuracy increases as the size of the space increases. On the other hand, the increase of the search space implies increasing the average size of selected subsets. Indeed, the subset with the highest accuracy (94,2%) is selected from the feature space TOP100 and contains 39 features.

The behavior of GA/K-NN is quite different from several points of view. First, GA/K-NN is much more effective in selecting predictive subsets, irrespective of the size of the provided feature space. In particular, GA/K-NN selects the most predictive subset from TOP20, reaching an accuracy of 98,4% with only 4 features. The average size of the selected subsets, instead, depends significantly on the number of genes included in the TOPN, leading to predictive subsets of larger size in larger feature spaces. Finally, GA/K-NN turns out greatly superior in terms of computational cost, leading to a more effective feature selection in a very efficient way. This seems to suggest that a similarity-based classification approach, like K-NN, may be more suitable in this specific domain, where SVM has been so far considered the "best class" classification algorithm [13].

A further aspect to discuss is the relative importance of the selected genes. According to the model illustrated in Section 2, we have computed the frequency of membership of features in the subsets selected by GA/SVM and GA/K-NN. Table 3 shows the identification number of the features most frequently occurring and, in brackets, their position in the original ranked list.

Table 3. Frequency of features (identification number) selected by GA/SVM and GA/K-NN; in brackets, the ranking position of each feature

GA/SVM algorithm		GA/K-NN algorithm	
Features	Frequency	Features	Frequency
66 (15)	64,3%	1772 (10)	75,7%
493 (3)	61,4%	765 (4)	72,9%
1423 (5)	60,0%	1423 (5)	41,4%
1771 (6)	58,6%	267 (8)	35,7%
1772 (10)	57,1%	415 (21)	35,7%
897 (14)	52,9%	513 (7)	34,3%
1042 (19)	48,6%	1892 (20)	34,3%
765 (4)	47,1%	1771 (6)	32,9%
581 (35)	45,7%	897 (14)	32,9%
780 (12)	42,9%	822 (16)	32,9%

Some interesting considerations can be drawn from the above results. First of all, the two lists have 5 features in common (out of 10), revealing that GA/SVM and GA/K-NN quite agree in evaluating the relevance of genes, although selecting different gene combinations. Besides, features reported in Table 3 that appear only in the GA/SVM list are also selected by GA/K-NN with lower frequency. Vice versa, features appearing only in the GA/K-NN list are also selected by GA/SVM with lower frequency.

Further, the genes most frequently selected are not necessarily the top-ranked ones. For example, genes that exhibit the highest frequency (gene 66 for GA/SVM and gene 1772 for GA/K-NN) are placed at ranking position 15 and 10, respectively. In turn, some top-ranked genes such as 1671 and 249 don't appear at all in Table 3 even if they are at positions 1 and 2 of the ranked list. This confirms that, while useful to reduce the dimensionality of the initial problem, the ranking process is not by itself a suitable feature selection technique for microarray data.

4 Discussion and Concluding Remarks

The experimental results illustrated in Section 3 confirm the effectiveness of our hybrid feature selection. Differently from most filter-based approaches, where quite an arbitrary threshold is used to cut-off the list of ranked genes, we explore different threshold values in order to gain insight into the best trade-off between accurate results and computational cost. Moreover, by exploring feature spaces of different size and composition, our approach suits the broader search ability of the GA method since the extended search over multiple solution spaces results in different combinations of genes that achieve superior classification performance.

Tables 4 and 5 summarize our best results for Colon dataset in comparison with results of different state-of-art methods presented in recent microarray literature. In particular, in Table 4 the comparison is limited to GA-based feature selection approaches, while Table 5 refers to hybrid feature selection strategies not involving a genetic search. The conventional criteria are used to compare the results, i.e. the classification accuracy and the number of selected genes. As it can be observed, our GA/K-NN method achieves better accuracy than all other methods, except for [4]; in [4], however, the number of selected genes is greater than the one obtained with our method. The approach reported in [14] results in the smallest gene subset, but the corresponding accuracy is worse than that achieved by other approaches.

Table 4. Comparison with GA-based methods

	GA / K-NN	**[15]**	**[3]**	**[4]**
Accuracy (%)	98.4	93.6	97.0	99.4
Subset size	4	12	7	10

Table 5. Comparison with hybrid methods without GA

	GA / K-NN	**[7]**	**[16]**	**[14]**
Accuracy (%)	98.4	95.2	93.6	91.9
Subset size	4	6	4	3

As an additional point, our approach is able not only to select small subsets of informative genes but also to assess the relative importance of these genes. Indeed, many predictors are obtained, after which the frequency of gene selection is examined in order to identify the genes most relevant for cancer diagnosis. From this point of view, our approach may support the extraction of biological knowledge through a two-fold contribution, i.e. by discovering different alternative markers and by assessing the role of each gene within such markers.

Further experiments with our method are currently in progress on multiple microarray datasets. In particular, the analysis will be extended to consider different threshold values in the filter process, as well as different ranking criteria for weighting features.

References

1. Saeys, Y., Inza, I., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
2. Li, L., Weinberg, C.R., Darden, T.A., Pedersen, L.G.: Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. Bioinformatics 17, 1131–1142 (2001)
3. Reddy, A.R., Deb, K.: Classification of two-class cancer data reliably using evolutionary algorithms. BioSystems 72(2003), 111–129 (2003)
4. Huerta, E.B., Duval, B., Hao, J.K.: A hybrid GA/SVM approach for gene selection and classification of microarray data. In: Rothlauf, F., Branke, J., Cagnoni, S., Costa, E., Cotta, C., Drechsler, R., Lutton, E., Machado, P., Moore, J.H., Romero, J., Smith, G.D., Squillero, G., Takagi, H. (eds.) EvoWorkshops 2006. LNCS, vol. 3907, pp. 34–44. Springer, Heidelberg (2006)
5. Dessì, N., Pes, B.: An Evolutionary Method for Combining Different Feature Selection Criteria in Microarray Data Classification. Journal of Artificial Evolution and Applications. Article ID 803973 (2009)
6. Kudo, M., Sklansky, J.: Comparison of algorithms that select features for pattern classifiers. Pattern Recognition 33, 25–41 (2000)
7. Leung, Y., Hung, Y.: A multiple-filter-multiple-wrapper approach to gene selection and microarray data classification. IEEE/ACM Transaction on Computational Biology and Bioinformatics 7(1), 108–117 (2010)
8. Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D., Levine, A.: Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probed by Oligonucleotide Arrays. PNAS 96, 6745–6750 (1999)
9. Vapnik, V.N.: Statistical Learning Theory. Wiley Interscience, Hoboken (1998)
10. Cannas, L.M., Dessì, N., Pes, B.: A filter-based evolutionary approach for selecting features in high-dimensional micro-array data. In: Shi, Z., Vadera, S., Aamodt, A., Leake, D. (eds.) IIP 2010. AICT, vol. 340, pp. 297–307. Springer, Heidelberg (2010)
11. Cannas, L.M., Dessì, N., Pes, B.: Tuning evolutionary algorithms in high dimensional classification problems. In: SEBD 2010, Rimini, Italy, pp. 142–149 (2010)
12. Hall, M., et al.: The WEKA data mining software: an update. SIGKDD Explorations 11(1) (2009)
13. Statnikov, A., Wang, L., Aliferis, C.F.: A comprehensive comparison of random forests and support vector machines for microarray-based cancer classification. BMC Bioinformatics 9, 319 (2008)
14. Wang, Y., Makedon, F., Ford, J.C., Pearlman, J.D.: Hykgene: a hybrid approach for selecting marker genes for phenotype classification using microarray gene expression data. Bioinformatics 21(8), 1530–1537 (2005)
15. Peng, S., Xu, Q., Ling, X.B., Peng, X., Du, W., Chen, L.: Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. FEBS Letters 555(2), 358–362 (2003)
16. Yu, L., Liu, H.: Redundancy Based Feature Selection for Microarray Data. In: 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD 2004), pp. 737–742 (2004)

A Modified Apriori Algorithm for Analysing High-Dimensional Gene Data

Claudia Pommerenke[1], Benedikt Friedrich[2], Thorsten Johl[3], Lothar Jänsch[3], Susanne Häussler[3], and Frank Klawonn[2]

[1] Infection Genetics, Helmholtz Centre for Infection Research, Brunswick, Germany
claudia.pommerenke@helmholtz-hzi.de
[2] Computer Science, Ostfalia University of Applied Sciences, Wolfenbüttel, and Bioinformatics and Statistics Group, Helmholtz Centre for Infection Research, Brunswick, Germany
[3] Cellular Proteomics, Helmholtz Centre for Infection Research, Brunswick, Germany
[4] Chronic Pseudomonas Infections, Helmholtz Centre for Infection Research, Braunschweig and Twincore, Centre for Experimental and Clinical Infection Research, Hanover, Germany

Abstract. Modern high-throughput technologies allow the systematic characterisation of an organism but provide excessive amounts of data such as results from microarray gene expression experiments. Combining the information from various experiments will help to expand the knowledge about an organism. However, the analysis of a data set comprising measurements for thousands of genes under many conditions, requires efficient techniques to be feasible at all. Here, we refine a frequent itemset mining approach for scanning a high-throughput data set in order to identify subsets of genes and subsets of conditions with similar data patterns. As a use case, screenings of 4699 mutant clones of *Pseudomonas aeruginosa* each with a disrupted gene were considered under 109 conditions. We found an unexpected gene group with highly overlapping phenotypes. Therefore our approach is suitable to simultaneously find objects with similar pattern in high-dimensional data sets and their key characteristics within reasonable time.

Keywords: Apriori algorithm, high-dimensional data, phenomics.

1 Introduction

In the past decade, advances in high-throughput sequencing technologies have provided hundreds of complete microbial genomes. Despite assigning gene functions by bioinformatic predictions, one third of the genes lack sequence homology or information from genomic context and their interrelations still remain uncharacterised [3,13]. Large amounts of experimental data are also available providing explicit or implicit information about the role of genes. However, to derive functional relations between genes from such excessive data is often a difficult and challenging task. The algorithm we present here is motivated by a study on

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 236–243, 2011.

Pseudomonas aeruginosa, a highly versatile human pathogen with strong resistance to antibiotic treatment, for which phenotypes screening proved to be powerful for identifying genes with similar functions [15]. However, the concepts and ideas can be applied to other organisms as well when measurements for a larger number of genes and a larger number of experiments is available. In order to cope with a larger data set—in our specific example a table with more than 4500 rows and more than one hundred columns—we concentrate on genes with minimal overlap to the wildtype and simplify the data in a first step to a binary table. In order to characterise the phenotypic profile of the mutants that are extremly different to the wildtype we apply techniques from frequent itemset mining to find associations between genes and the key phenotypes between similar gene groups.

2 Formal Problem Definition

The data to be analysed can be structured as a table containing the results from experiments for an organism of the form as shown in Table 1. For each gene corresponding to a row several measurements from different experimental settings are available indicated as columns and it is assumed that the table has a larger number of rows and columns ($n \gg 200$, $k \gg 20$).

The table could contain gene expression values or other information. But it can also be other data, for instance measurements from phenotypic assays as in the case of our *Pseudomonas aeruginosa* data.

The task of identifying groups of related genes consists of the finding:

- suitable subgroups of genes and
- subsets of conditions under which the genes in a subgroup are similar.

Table 1. General structure of the data table

Gene name	Condition 1	...	Condition k
Gene 1	a_{11}	...	a_{1k}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
Gene n	a_{n1}	$\vdots$	a_{nk}

2.1 An Example Data Set

Pseudomonas aeruginosa is a human pathogen of high clinical interest. Its strong intrinsic resistance to antibiotics is attributed to many different mechanisms in the complex cellular machinery. Studying distinct phenotypic characteristics of knock-out mutants defective in one single gene will provide functional relations of genes and thereby potentially related genes acting in similar or even same intracellular pathways.

Our *P. aeruginosa* data set is based results from a novel combination of high-throughput phenotype assays and a mutant library [12] for which *P. aeruginosa* PA14 single mutants defective in 70% of all genes were tested and we found that these phenotype screenings were useful to support known classification schemes for *P. aeruginosa* genes [15].

Here, distinct to the previous study, we focus on the mutants that strongly differ from the wildtype and inquire for mutants with similar phenotpye behaviour and for the key tests characterising them. Each row in our data table refers to a mutant clone for which the corresponding gene had been deactivated. Altogether, we had 4699 of these mutant clones. The columns in the table correspond to 109 different conditions under which the phenotypes of the mutant clones were tested. These 109 tests were based on on phenotype screenings that recorded growth curves after treatment with different antibiotics (see Fig. 1) and curves indicating metabolic conversion or growth due to substrate addition [15].

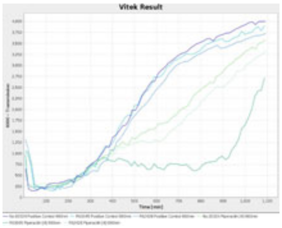


Fig. 1. Measured growth curves for three different mutants. Control curves without treatment (upper, curves) and curves measured for 4 μg/ml Piperacillin addition (lower, curves).

Since the task of finding functionally related mutants with similar patterns requires to consider more than half a million growth curves ($4699 \cdot 109$), brute force search techniques must fail due to an explosion of the computational costs. Therefore, a crucial part of this work is to find tractable and efficient algorithms that can cope with such large data sets. This requires preprocessing of the data, which we will first discuss for the general case (i.e. high-throughput gene expression analysis) and then for our specific data set.

2.2 Preprocessing of Data

In order to simplify the analysis and make it computationally tractable, we restrict our considerations to saying that two entries in the data table are either similar or not similar, i.e. for each pair of genes and each condition, we define whether the two genes behave similar in the corresponding condition or not. The definition of similar behaviour depends on the problem and the type of data. For gene expression data one could choose a threshold and say that a gene is (significantly) expressed compared to another experimental setting when its fold change expression (entry in the data table) exceeds the threshold.

A common choice for the threshold is the two-fold differential expression which is applied globally on the whole data set. Another option for the definition of similar behaviour could be based on the difference of the absolute expression values in each single condition. When the expression levels of two genes in one single condition differ less than a given threshold, the behaviour of these two genes is considered to be similar in this condition.

For the *P. aeruginosa* data set, we consider in this study, we have to deal with growth curves instead of gene expression levels. How we define similarity between growth curves will be explained in the next section.

2.3 Preprocessing of the Example Data Set

We aim to find related genes (mutants) with similar curves for a significant number of conditions, but not necessarily for all conditions. For this purpose, a similarity or distance measure for the curves is required to decide whether two curves can be considered to be similar. In our study, different from the previous study, after some necessary preprocessing steps applied to the curves, including value correction to defined time intervals, antibiotic curves correction to control curves, and subsequently normalisation by median polish [9] [15], the distance measure between two curves of two mutants x and y with n timepoints, is calculated from the sum of (absolute) pointwise differences at equidistant intervals:

$$d_{x,y} = \sum_{1}^{n} |x_i - y_i| \tag{1}$$

However, other distance measures could also be used. In principle, it would be possible to use the more informative distance measure itself instead of a simplified discretised version reducing the quantitative distance value to a binary value, i.e. similar or not similar. However, this would add another dimension of complexity and would make the problem even more intractable as being discussed in Section 3.

We define a threshold value for the distance up to which two curves are considered to be similar. The threshold is derived from nine repeated measurements of one mutant. These measurements indicate, how much the same curve might vary due to noise coming from the measurement, biological factors and inexact adjustments of the experimental setting. The threshold is set to the highest distance value of the replicate pairs r_{max}. The discretised distance between two mutants x and y for one phenotypic test is as follows:

$$dd_{x,y} = \begin{cases} 1 & \text{if } d_{x,y} > r_{max}, \\ 0 & \text{if } d_{x,y} \leq r_{max}. \end{cases} \tag{2}$$

Since we are interested in the mutants with maximal difference in the phenotypic profile in comparison to the wildtype clone, these mutants are filtered at first. 555 mutants are selected for further analysis that are different to the wildtype in more than about 40 phenotypic tests. The aim is to find mutant groups for these preselected mutants that exert similar pattern and to identify the key characteristics for these groups.

Consequently, the data to be analysed consists of 109 (48 for biochemical activities plus 61 for antibiotics susceptibility tests) tables with 555 rows and 555 columns, one for each mutant (gene) (no selection on mutants of the same gene or with outlier values). Each binary entry in such a table indicates, whether the curves of the two corresponding mutants are considered to be similar (entry 1) or not similar (entry 0) under the test condition that the table represents:

$$s_{x,y} = \begin{cases} 0 & \text{if } d_{x,y} > r_{max}, \\ 1 & \text{if } d_{x,y} \leq r_{max}. \end{cases} \tag{3}$$

3 Algorithm for Finding Groups of Genes with Similar Characteristics

The identification of groups of genes with similar behaviour requires to find subsets of genes and subsets of conditions under which the genes show similar behaviour.

In principle, one could cluster the genes based on their behaviour in the different conditions. In the case of gene expression data, one could directly apply a clustering algorithm. For our example of growth curves, we define a similarity or distance measure between curves. In any case, one would use all k conditions as attributes for clustering. But since the intention of this study is to find subsets of conditions in which certain subsets of genes show similar behaviour, standard clustering taking all features (conditions) into account is not suitable. One could try algorithms that incorporate individual feature selection for each cluster [11]. However, this clustering algorithm suffers like more or less all clustering algorithms from the curse of dimensionality [1,8,14], which refers to the problem that distances of data points in high-dimensional spaces tend to get indistinguishable and therefore cause difficulties for clustering nearest neighbour searches.

Even if we choose a fixed subset of test conditions, finding groups of genes with similar behaviour under these conditions corresponds to the maximum cliques problem which is known to be NP-complete [6]. To see the connection to the maximum cliques problem, a graph can be constructed from the tables corresponding to the fixed subset of conditions. The vertices of the graph are the genes and two genes are connected by an edge if and only if they have similar curves under all test conditions considered for the graph. A (maximum) subset of similar genes under these conditions corresponds to a (maximum) clique in the graph. If we require that groups of similar genes should behave similar in at least half of the considered test conditions, we would have to find maximum cliques in 2^{108} such graphs, so that the maximum clique approach is not applicable here.

In order to solve the problem, we borrow ideas from finding frequent itemsets, a task well-known in data mining. In the simplest case, frequent itemset mining is applied to market basked analysis to find groups of products (items) that are frequently bought together. Algorithms for frequent itemset mining start with single items that are bought with a sufficient frequency and then find step by step larger itemsets. The Apriori algorithm [2] for frequent itemset mining is based on breadth first search, whereas the Eclat algorithm [16] relies on depth first search. A variety of improvements of these algorithm have been proposed, for instance to minimize accesses to the database [7].

Instead of starting with finding frequent single items, in the first step we consider single test conditions. For each condition, a list of groups of genes is constructed. Each group is a maximal set of genes whose behaviour is pairwise similar in the corresponding condition. Then we combine test conditions step by step and assign to each combination of conditions lists of groups of genes, such that all genes in a group show similar behaviour in all considered conditions. Of course, for certain combinations of conditions, the search will terminate early, since there is no (large enough) group of genes with similar behaviour for these conditions. It should be noted that the candidate generation process is much more complicated than in the case of frequent itemset mining. Candidate generation refers to the construction of possible larger itemsets from smaller itemsets. In our case, it is not sufficient to combine simple sets, but lists of sets. Therefore, even efficient adaptations of the Apriori or the Eclat algorithm cannot handle larger data sets.

Therefore, for larger data sets as in the case of our *P. aeruginosa* example, we have chosen another approach that does not try to find arbitrary groups of mutants with similar genes. Instead, we first choose a gene (of interest) g and then find groups of genes with similar behaviour to this specific gene g. We now start with a list of conditions for each gene. The list contains those conditions under which the gene shows a similar behaviour as g. Although we have 4698 (number of mutants $-$ 1) lists in the beginning in the *P. aeruginosa* example data set, the maximum length of each list in the beginning is 109 (number of test conditions). In the next step, we consider lists of pairs conditions, then sets of three conditions etc.

We additionaly restrict the search during the initial procedures to those genes whose lists contain at least a minimum number of conditions.

4 Results and Conclusions

The heuristic Apriori algorithm is applied to a subset of mutants defective in one or two single genes (555) that are maximal different to the wildtype. Due to time and memory limitations, the algorithm is conducted on maximal 20 possible related mutant members.

The top one of the identified groups with high similarity contains 18 mutants (see Tab. 2A) which are either known to be involved in diverse biological functions or yet unidentified. The composition of mutants in one group for the ten top groups varies in one or two different mutants only, which is also observed for the corresponding phenotypic tests (see Tab. 2B). These unexpected but clearly defined mutants of the top ten groups that share a common phenotype profile have not been described to be functionally related so far in other high-throughput experiments for our microbial pathogen [4,5,10]. However, it will be

Table 2. A: List of mutants with similar characterstics identified by the heuristic Apriori algorithm. B: Overlapping phenotypic tests of the top one mutant group (see Tab. 2A). AB: Antibiotic susceptibility, BC: Biochemistry test.

A

Mutant (gene) name	Biological relevance
mexA	Multidrug efflux
exoT	Exoenzyme
gcdH	Diverse metabolic processes
phzM	Phenanzine biosynthesis
phzC1	Phenanzine biosynthesis
cupB3	Outer membrane protein
epd	Vitamin B6 metabolism
pcaT	Dicarboxylic acid transporter
pilU	Twitching motility
PA14_00780	Unknown function
PA14_03760	Unknown function
PA14_05370	Unknown function
PA14_07480	Unknown function
PA14_07490-PA14_07510	Unknown function
PA14_09760	Unknown function
PA14_11530	Unknown function
PA14_11840-PA14_11850	Unknown function
PA14_15540	Unknown function

B

Phenotype test	Type
Ampicillin, 4	AB
Ampicillin, 8	AB
Cefpodoxime, 0.5	AB
Cefoxitin, 16	AB
Cefoxitin, 8	AB
Ceftazidime, 1	AB
Cefuroxime, 2	AB
Cefuroxime, 8	AB
Tigecycline, 4	AB
β-N-Acetyl-Galactosaminidase	BC
Ala-Phe-Pro-Arylamidase	BC
L-Pyrrolydonyl-Arylamidase	BC
Tyrosine-Arylamidase	BC
L-Histidine-Assimilation	BC
γ-Glutamyl-Transferase	BC
Fermentation/Glucose	BC
Citrate (Sodium)	BC
Lipase	BC

intriguing to test these mutants for their relatedness in the bacterial signaling network.

One half of the common phenotypic profile of the identified mutant group is attributed to the antibiotic susceptibility testing and the other half to the biochemical testing despite the contribution of 1/3 biochemical to 2/3 antibiotic screenings for this analysis, hence slightly more similar biochemical screenings are yielded than expected. This may be due to several reasons: a high sensitivity of the antibiotic susceptibility tests, or more intriguingly, a more distinct characterisation of the identified mutants by biochemistry screenings at least for the identified mutant group.

Further improvements could include different distance or similarity measures for the mutant pairs or an intelligent summarisation of similar mutant groups in order to find further interesting groups of related mutants.

In conclusion, the heuristic Apriori algorithm reveals novel potential relationships between mutants or genes and identifies the key common characteristic features of these yet unknown functional interrelations. In this way, complex phenotypes may be identified and hence functional characterisation of specific genes may be discovered in tractable time.

References

1. Aggarwal, C.C., Hinneburg, A., Keim, D.A.: On the surprising behavior of distance metrics in high dimensional space. In: Van den Bussche, J., Vianu, V. (eds.) ICDT 2001. LNCS, vol. 1973, pp. 420–434. Springer, Heidelberg (2000)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proc. 20th Int. Conf. Very Large Data Bases (VLDB), pp. 487–499. Morgan Kaufmann, San Francisco (1994)

3. Bochner, B.R., Giovannetti, L., Viti, C.: Important discoveries from analysing bacterial phenotypes. Mol. Microbiol. 70, 274–280 (2008)
4. Breidenstein, E.B.M., Khaira, B.K., Wiegand, I., Overhage, J., Hancock, R.E.W.: Complex ciprofloxacin resistome revealed by screening a Pseudomonas aeruginosa mutant library for altered susceptibility. Antimicrob. Agents Chemother. 52, 4486–4491 (2008)
5. Fajardo, A., Martínez-Martín, N., Mercadillo, M., Galán, J.C., Ghysels, B., Matthijs, S., Cornelis, P., Wiehlmann, L., Tümmler, B., Baquero, F., Martínez, J.L.: The neglected intrinsic resistome of bacterial pathogens. PLoS ONE 3, e1619 (2008)
6. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
7. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Chen, W., Naughton, J., Bernstein, P.A. (eds.) ACM SIGMOD Int. Conf. Management of Data, pp. 1–12. ACM Press, New York (2000)
8. Hinneburg, A., Aggarwal, C.C., Keim, D.A.: What is the nearest neighbor in high dimensional spaces? In: Abbadi, A.E., Brodie, M.L., Chakravarthy, S., Dayal, U., Kamel, N., Schlageter, G., Whang, K.-Y. (eds.) Proc. Very Large Databases Conf. (VLDB), pp. 506–515. Morgan Kaufmann, San Francisco (2000)
9. Hoaglin, D.C., Mosteller, F., Tukey, J.W.: Understanding Robust and Exploratory Data Analysis. Wiley Classics Library. Wiley, NY, USA (2000)
10. Johnson, D.A., Tetu, S.G., Phillippy, K., Chen, J., Ren, Q., Paulsen, I.T.: High-throughput phenotypic characterization of Pseudomonas aeruginosa membrane transport genes. PLoS Genet. 4, e1000211 (2008)
11. Keller, A., Klawonn, F.: Fuzzy Clustering with Weighting of Data Variables. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 8, 735–746 (2000)
12. Liberati, N.T., Urbach, J.M., Miyata, S., Lee, D.G., Drenkard, E., Wu, W., Villanueva, J., Wei, T., Ausubel, F.M.: An ordered, nonredundant library of Pseudomonas aeruginosa strain PA14 transposon insertion mutants. Proc. Natl. Acad. Sci. USA 103, 2833–2838 (2006)
13. Oti, M., Huynen, M.A., Brunner, H.G.: Phenome connections. Trends Genet. 24, 103–106 (2008)
14. Pestov, V.: On the geometry of similarity search: Dimensionality curse and concentration of measure. Inf. Process. Lett. 73, 47–51 (2000)
15. Pommerenke, C., Müsken, M., Becker, T., Dötsch, A., Klawonn, F., Häussler, S.: Global Genotype-Phenotype Correlations in Pseudomonas aeruginosa. PLoS Pathog 6, e1001074 (2010)
16. Zaki, M., Parthasarathy, S., Ogihara, M., Li, W.: New algorithms for fast discovery of association rules. In: Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining KDD 1997, Newport Beach, CA, pp. 283–296. AAAI Press, Menlo Park (1997)

Evolving Recurrent Neural Models of Geomagnetic Storms

Derrick T. Mirikitani, Lisa Tsui, and Lahcen Ouarbya

Department of Computer Science
Goldsmiths College, University of London
New Cross, London SE14 6NW

Abstract. Genetic algorithms for training recurrent neural networks (*RNN*s) have not yet been considered for modeling the dynamics of magnetospheric plasma. We provide a discussion of the previous state of the art in modeling $\boldsymbol{D_{st}}$. Then, a recurrent neural network trained by a genetic algorithm is proposed for geomagnetic storm forecasting. The exogenous inputs to the *RNN* consist of three parameters, $\boldsymbol{b_z}$, $\boldsymbol{n}$, and $\boldsymbol{v}$, which represent the southward and azimuthal components of the interplanetary magnetic field ($\boldsymbol{IMF}$), the density of electromagnetic particles, and the velocity of the particles respectively. The proposed model is compared to a model used in operational forecasts on a series of geomagnetic storms that so far have been difficult to forecast. It is shown that the proposed evolutionary method of training the *RNN* outperforms the operational model which was trained by gradient descent.

Keywords: Geomagnetic Storms, Recurrent Neural Networks, Genetic Algorithms.

1 Introduction

Neural Networks have established themselves as effective tools in the prediction of geomagnetic disturbances [13,14,17]. In this paper we extend the work in the field of geomagnetic storm forecasting by proposing a genetic algorithm for estimation of recurrent neural network (*RNN*) parameters [2]. The advantage of our approach is improved forecasts through the use of a global optimization algorithm. Previous work in the area has relied on gradient based optimization [14,17] which is susceptible to becoming trapped in local minima on the error surface, leading to suboptimal solutions. Unlike gradient based learning algorithms which progress downhill until becoming trapped in a minima, genetic algorithms use stochastic search operators, such as mutation and crossover, which allows the algorithm to move, in one step, from inside a local minima across a ridge to an even lower local minima with no more difficulty than descending directly into the local minima itself [16]. Genetic algorithms are efficient optimization procedures as they are able to reduce the forecast errors of the model monotonically. This study found that the use of the genetic algorithm can lead to improved out of sample performance on $\boldsymbol{D_{st}}$ forecasting tasks over models estimated with gradient descent. In the following sub-section we provide an overview of geomagnetospheric distrubances and a brief review of neural forecasting of geomagnetic storms.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 244–251, 2011.

1.1 Geomagnetic Storms

The magnetosphere is a magnetic field surrounding the Earth that shields and deflects charged particles emitted from the Sun from hitting the Earth. The sun also has a magnetic field, however the Sun's magnetic field is much more complex than the Earths. It is well known that the variation in the Sun's magnetic field influences the structure of the magnetic field surrounding the Earth [1,3,8]. An Interplanetary Magnetic Field (IMF) is formed when the solar wind expands the reach of the Sun's magnetic field, which can extend to hit Earth's magnetic field. The IMF can cause energetic particles to enter into the Earth's magnetic field which results in magnetosphere disturbances. Disruption of the magnetosphere takes place when a transfer of energy from the solar wind opposes the Earth's magnetic field. A magnetospheric storm occurs if this transfer of energy persists for several hours [8].

Geomagnetic storms can have many negative effects on Earth resulting in widespread problems and damage to electric power grids, gas pipelines, power generations facilities, and Global Positioning System (GPS) disruption. Forecasting the earth's magnetic field can provide vital information about the intensity of future magnetospheric disturbances. At mid-latitudes, these magnetic storms are measured in relation to the horizontal component of the Earth's magnetic field [8]. The mean of this horizontal component is used to form an index known as the D_{st} index. There have been various studies that have shown a correlation between the value of the D_{st} index and the magnetic storm's intensity [5,9], where the more negative the D_{st} index the greater the intensity of the magnetic storm. The physical interaction between the IMF and the magnetosphere takes place at the magnetopause boundary where a detailed understanding of this interaction is not yet fully understood. Previous researchers have built non parametric predictive models based on recurrent neural networks (*RNN*s) to model this interaction [13,17].

Previous work in modeling the relationship between IMF and D_{st} with *RNN*s have relied heavily on first-order gradient based methods for parameter estimation of the model [12,17] which has resulted in long training times, uncertain convergence, and possibly vanishing gradients. In this paper we investigate solutions to this problem through the use of the genetic algorithms for *RNN* training [2]. The advantage of our approach is a framework based on global search strategies resulting in superior convergence and accurate forecasts. The main results of the paper are as follows: 1) a stochastic global search strategy for *RNN* parameter estimation for D_{st} forecasting, 2) improved forecast accuracy over previously demonstrated results.

2 Recurrent Neural Networks

In this study, the recurrent architecture known as the Elman network (*RNN*) [4] is chosen as previous studies have found successful results with *RNN*s. Feed-forward networks are not considered in this study due to poor performance in modeling the recovery phase dynamics [6]. This is mostly likely due to the limitation of the feed-forward architecture, i.e. limited temporal memory, bounded by the dimension of the input window. The Elman *RNN* consists of a feed-forward multi-layer perceptron architecture, augmented with a context layer which connects every hidden neuron output to every hidden neuron input. The context layer allows for a memory of past states of the network. The network weights for the hidden layer of size H can be represented as a matrix defined as

$$\mathbf{W}_h = [\mathbf{w}^1, \mathbf{w}^2, \ldots, \mathbf{w}^H]^T \tag{1}$$

where $\mathbf{w}^i = [w_{i,0}, w_{i,1}, \ldots, w_{i,j}]^T$ $i = 1, 2, \ldots, H$, $j = I + H$, and I is the size of the input layer. The hidden state $\mathbf{s}(t)$ is connected to the network output $y(t) \in \mathbb{R}^1$ via the output weight vector

$$\mathbf{w}^{out} = [w_0, w_1, \ldots, w_H]^T \tag{2}$$

The operation of the Elman network is described by the following discrete time equations:

$$\begin{aligned} \mathbf{s}(t) &= \mathbf{g}\Big(\mathbf{W}_h[b, \mathbf{x}(t)^T, \mathbf{c}(t)^T]\Big) \\ \mathrm{y}(t) &= \mathbf{f}(\mathbf{w}_{out}^T[b, \mathbf{s}(t)^T]^T) \end{aligned} \tag{3}$$

where $\mathbf{c}(t) = \mathbf{s}(t-1) \in \mathbb{R}^H$ is the context vector, $\mathbf{u}(t) = [\boldsymbol{b_z}(t)/30,\ \boldsymbol{n}(t)/80 - 1,\ \boldsymbol{v}(t)/400 - 1.5]$ is the exogenous $\boldsymbol{IMF}$ and solar wind input quantities, and b is the bias. The functions $\mathbf{g}(\cdot)$ and $\mathbf{f}(\cdot)$ are typically logistic sigmoidal nonlinearities $\sigma(a) = 1/(1 + \exp(-a))$ which map the input a from $\mathbb{R}$ into a bounded interval $\Omega = (0, 1)$ of length $|\Omega| = 1$ where $\Omega \subset \mathbb{R}$. All weight vectors $\mathbf{w}_i$ $i = 1, 2, \ldots, H$ and the output weight vector from Equation 2 can be arranged into a single vector as follows:

$$\begin{aligned} \mathbf{C}_1 &= [(\mathbf{w}_1^1)^T, (\mathbf{w}_1^2)^T, \ldots, (\mathbf{w}_1^H)^T, (\mathbf{w}_1^{out})^T]^T \\ &= [w_{1,0}, w_{1,1}, \ldots, w_{1,j}, w_{2,0}, w_{2,1}, \ldots, w_{2,j}, \ldots, w_0, w_1, \ldots, w_H] \end{aligned} \tag{4}$$

The total number of weights in the *RNN* is $\mathfrak{n} = (I \times H) + H^2 + 2H + 1$

2.1 Evolutionary Training of the *RNN*

It has become commonplace in the geomagnetic storm forecasting literature to build recurrent neural models based on backpropagation of errors. The backpropagation algorithm computes model derivatives which are used to optimize the neural model via gradient descent optimization. However, it is well known that gradient based learning algorithms are susceptible to becoming trapped in local minima on the model error surface [10]. This generally leads to poor model forecasts on out of sample data. The standard approach to alleviate problems of this nature is to use global search strategies such as genetic algorithms.

Genetic algorithms (GAs) are stochastic global optimization procedures motivated by our understanding of the process of evolution found in nature [7,11]. Genetic algorithms can promote learning or adaptation in *RNN*s through evolution of the *RNN* model parameters (i.e. *RNN* weights). This is typically accomplished by encoding the *RNN* model parameters into chromosome like structures. As evolution works on populations of individuals, the algorithm starts off by randomly generating a population of chromosomes (*RNN*s). Each chromosome in the population represents an *RNN* which can compute a solution to the forecasting problem. The quality of the solution can be though of as the chromosome's fitness. This is measured by the insample error of the *RNN* in terms of mean squared error (MSE).

The driving force behind evolution is selection. The evolutionary process favors solutions that carry higher relative fitness. In the genetic algorithm, it is the chromosomes with higher fitness that are more likely to be selected for reproduction, leading to the survival of organisms with fitter characteristics in the next generation. Here we favor individuals which have lower error (higher fitness) on the training set (insample data). Roulette wheel selection is used to probabilistically choose the parents that will create the offspring that survive in the next generation. The offspring are created by applying the standard genetic operators, crossover and mutation, on the probabilistically selected parents. The offspring are then placed into a population representing the next generation. Performing these steps over multiple generations tends to lead to successive populations of increasing fitness, converging toward the globally optimal solution.

The genetic algorithm for training *RNN*s can be summarized as follows:

1. Randomly generate the initial population of chromosomes.
2. Compute the fitness of each member of the population by running each *RNN* over the training data and measuring the error.
3. If at least on of the members of the population have lower fitness (higher MSE) than the predetermined requirements then stop. Otherwise, continue on to the next step.
4. Copy the chromosome with the highest fitness into the population of the next generation (elitism).
5. Continue to fill the population of the next generation by applying the selection operator to select the parents and then apply the genetic operators of crossover and mutation.
6. Go to step 2

The genetic algorithm is used to find a set of weights that minimizes the error between the training targets and the network output.

2.2 Fitness Function

To measure the fitness of each chromosome in the population, the mean squared error is used to measure the errors committed by each *RNN* on the training data. The error measure used is the mean squared error:

$$mse = \frac{1}{\tau}\sum_{i=1}^{\tau}(d_i - y_i)^2 \tag{5}$$

where τ is the number of in sample data, d_i are the targets and y_i are the network outputs. To evaluate the fitness of each chromosome in the population, the chromosome is transformed into a *RNN*; each weight in the chromosome is placed in its corresponding position in the connectivity graph of the network. The training set is then presented to the *RNN* which processes the data and provides a one step ahead forecast. The errors are measured via the MSE error measure (Equation 5).

2.3 Cross over

The crossover operator takes two parent chromosomes and creates two children. By randomly selecting a point $i \sim \mathcal{U}(1, \mathfrak{n} - 1)$, from the uniform distribution, on the chromosome and then cutting and swapping the genetic material from each parent, two child chromosomes can be made.

Assume that $\mathbf{C}_1 = (c_1^1, \dots, c_1^{\mathfrak{n}})$ and $\mathbf{C}_2 = (c_2^1, \dots, c_2^{\mathfrak{n}})$ are two chromosomes that have been selected for reproduction. The crossover operator is used to create the offspring. We apply simple cross over where a position on the chromosome $i \in \{1, 2, \dots, \mathfrak{n} - 1\}$ is randomly chosen and two new offspring are generated:

$$\begin{aligned} \acute{\mathbf{C}}_1 &= \{c_1^1, c_1^2, \dots, c_1^i, c_2^{i+1}, \dots, c_2^{\mathfrak{n}}\} \\ \acute{\mathbf{C}}_2 &= \{c_2^1, c_2^2 \dots, c_2^i, c_1^{i+1}, \dots, c_1^{\mathfrak{n}}\} \end{aligned} \tag{6}$$

2.4 Mutation

The mutation operator may be invoked on the offspring. Assume that the offspring $\acute{\mathbf{C}}_1 = \{c_1^1, \dots, c_1^{\mathfrak{n}}\}$ is selected for mutation. Then a position c^i on the chromosome is chosen to be mutated, where the position of the mutation is an integer selected from the uniform distribution $i \sim \mathcal{U}(1, \mathfrak{n})$. The gene c^i is mutated by adding by a random number r to itself. The new gene is

$$c^i(new) = c^i + r \tag{7}$$

where $r \sim \mathcal{N}(0, \sigma)$ is a number drawn from the Gaussian distribution with mean 0 and standard deviation σ^2. The new gene $c^i(new)$ then replaces the old gene c^i at the i^{th} position on the chromosome.

3 Experimental Results

This section presents the results of both Lundstedt's algorithm [14,15] and the genetic algorithm and provides a comparison of the model forecasts. It shows how genetic algorithms can improve the forecast. Lundstedt, uses both $\boldsymbol{IMF}$ ($\boldsymbol{b_z}$) and solar wind data ($\boldsymbol{n}, \boldsymbol{v}$) as inputs to the neural network to forecast the $\boldsymbol{D_{st}}$ index. The inputs for the Lund based algorithms were scaled by the following constants $\boldsymbol{b_z}(t)/30,\ \boldsymbol{n}(t)/80 - 1,\ \boldsymbol{v}(t)/400 - 1.5$. The output of the *RNN* was scaled by $\hat{\boldsymbol{D}}_{\boldsymbol{st}}(t+1) = 150y(t) - 100$. The same scaled inputs were also used for the genetic algorithm trained *RNN* in order to make a direct comparison of the forecast.

The training set was constructed from 6 storms from the dates of 01-01-2001 to 06-01-2002. To speed up training, the quiet periods were largely omitted. The evolutionary algorithm was allowed to run until there was an average of 12nT error per hour.

Figure 1 illustrates the forecast of a moderate storm that occurred between the period of 31^{st} October - 4^{th} November 2001. The green line indicates the target values $\boldsymbol{D_{st}}$. From 0 hours there is a steep decrease in $\boldsymbol{D_{st}}$ to around -100nT. During the period of 10 hours to 20 hours the $\boldsymbol{D_{st}}$ fluctuates and reaching a minimum of around -107nT. This is followed by a slow recovery phase.

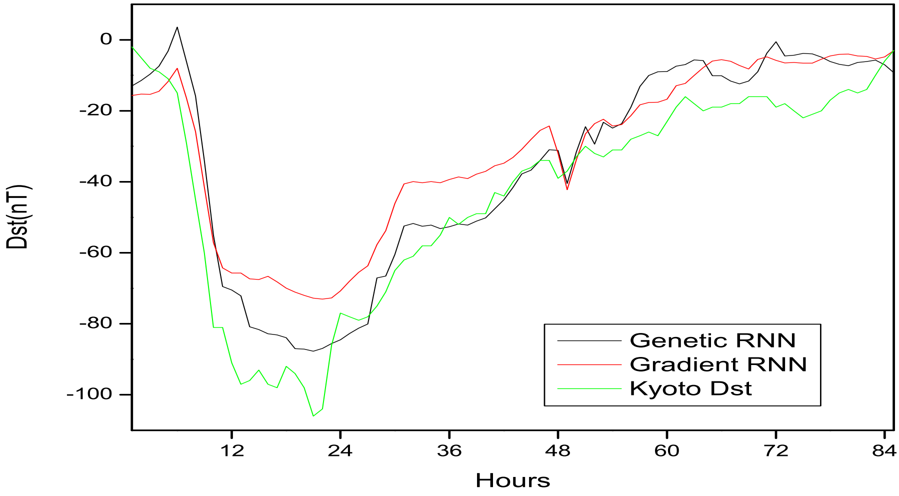


Fig. 1. Prediction of D_{st} from 31st October - 4th November 2001

Table 1. Performance of RNN models on out of sample data

Type	Lund RMSE	Genetic RMSE
Mild Storm	14.92	11.38
Strong Storm	55.31	16.37

Both algorithms behave similarly during the initial drop of D_{st} and fail to capture the initial compression of the storm. After 12 hours Lund's algorithm (red line) stops reproducing the target line and goes through a flat minimum at around -70nT compared to the genetic algorithm (blue line), which continues further downward toward the target. Both algorithms fail to capture the peak of the storm; however the genetic algorithm provides a more close forecast to the target during this period in comparison to Lund's algorithm. The recovery phase of the storm begins at around -105nT. The recovery phase of Lund's algorithm commences earlier than expected at around -73nT compared to the genetic algorithm at around -93nT which is closer to the target line. The second row of Table 1 shows the performance of both models in terms of root mean squared error (RMSE). The proposed algorithm slightly outperforms the gradient based algorithm of Lund. Figure 2 presents the forecast of a strong storm that occurred between the period of 24th November - 28th November 2001. The figure highlights that overall the genetic algorithm outperformed Lund's algorithm especially during the recovery phase.

At the beginning of the storm, Lund's algorithm does not initially capture the start of the storm until later on at around -20nT. Lund's algorithm fails to reproduce the minimum of the storm at around -220nT, in comparison to the genetic algorithm. Although the genetic algorithm begins to capture the storm earlier than expected, it reaches a minimum of around -225nT compared to Lund's algorithm of around -180nT. The recovery phase of this storm highlights the performance differences in both algorithms. Figure 2

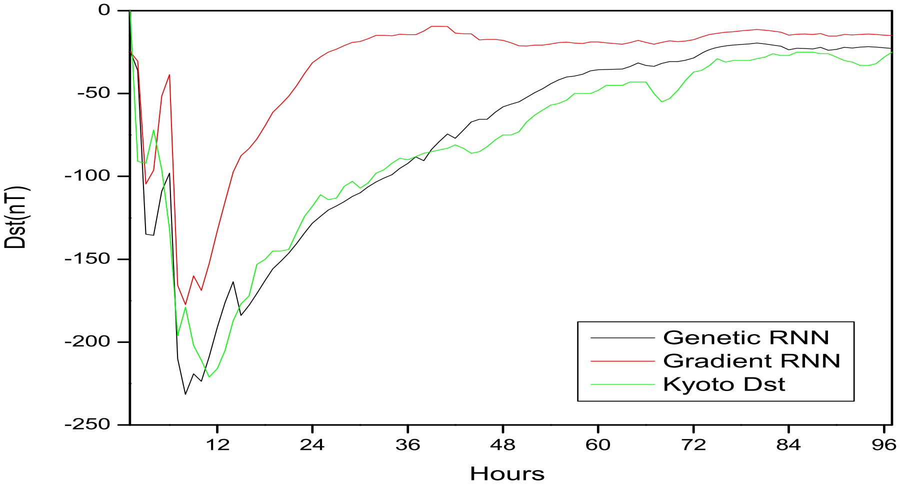


Fig. 2. Prediction of D_{st} from 24th November - 28th November 2001

highlights how the genetic algorithm reproduces a better representation of the target line of the storm compared to Lund's algorithm, and this is emphasized in the recovery phase of the storm. The genetic algorithm recovery follows the target line closely compared to the Lund's algorithm which produces a sharp and rapid recovery phase. In the third row of Table 1, the performance of both models in terms of RMSE are presented for the strong storm. On the strong storm, the genetic trained *RNN* significantly outperforms the gradient based model.

4 Future Work

Neural networks have shown encouraging results in modeling D_{st}. The research presented here demonstrate the ability of evolutionary trained *RNN*s to accurately capture the behavior of D_{st}. Future work will focus on using various inputs to the RNN, as it may be that the limiting factor now holding back the performance is the quality of the inputs. We plan to look into transformations on the inputs to the *RNN*, as well as using IMF data only [17].

5 Concluding Remarks

This paper introduced genetic algorithms for training *RNN*s for modeling geomagnetic activity induced through IMF plasma. The proposed model implements a global search strategy for parameter estimation of the *RNN* [2]. Through a comparison between the proposed models and [15], we have observed an increase in prediction accuracy of the *RNN* trained with the Genetic Algorithm. The advantage of the genetic algorithm trained *RNN* is due to the GA's ability to move out of local minima during training, where as the previously employed learning algorithm (gradient descent) was unable to escape local minima. It is known that the geomagnetic forecasting literature has heavily utilized first order gradient based methods. This paper has shown improvement with genetic algorithm trained *RNN*s.

References

1. Axford, W.I., Hines, C.O.: A unifying theory of high-latitude geophysical phenomena and geomagnetic storms. Can. J. Phys. 39, 1433–1464 (1961)
2. Blanco, A., Delgado, M., Pegalajar, M.C.: A real-coded genetic algorithm for training recurrent neural networks. Neural Networks 14, 93–105 (2001)
3. Dungey, J.W.: Interplanetary magnetic field and the auroral zones. Phys. Rev. Lett. 26, 47–48 (2000)
4. Elman, J.L.: Finding Structure in Time. Cognitive Science 14, 179–211 (1990)
5. Farrugia, C.J., Freeman, M.P., Burlaga, L.F., Lepping, R.P., Takahashi, K.: The earth's magnetosphere under continued forcing - Substorm activity during the passage of an interplanetary magnetic cloud. J. Geophys. Res. 98, 7657–7671 (1993)
6. Gleisner, H., Lundstedt, H., Wintoft, P.: Predicting Geomagnetic Storms From Solar-Wind Data Using Time-Delay Neural Networks. Ann. Geophys. 14, 679–686 (1996)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison Wesley Pub. Co., Reading (1989)
8. Gonzales, W.D., Joselyn, J.A., Kamide, Y., Kroehl, H.W., Rostoker, G., Tsurutani, B.T., Vasyliunas, V.M.: What is a geomagnetic storm? J. Geophys. Res. 99, 5771–5792 (1994)
9. Gosling, J.T., McComas, D.J., Phillips, J.L., Bame, S.J.: Geomagnetic activity associated with earth passage of interplanetary shock disturbances and coronal mass ejections. J. Geophys. Res. 96, 7831–7839 (1991)
10. Gupta, J.N.D., Sexton, R.S.: Comparing backpropagation with a genetic algorithm for neural network training. Omega 27(6), 679–684 (1999)
11. Holland, J.H.: Adaptation in natural and artificial systems. MIT Press, Cambridge (1992)
12. Lundstedt, H.: Neural Networks and prediction of solar-terrestrial effects. Planet. Space Sci. 40, 457–464 (1992)
13. Lundstedt, H., Wintoft, P.: Prediction of geomagnetic storms from solar wind data with the use of a neural network. Ann. Geophys. 12, 19–24 (1994)
14. Lundstedt, H., Gleisner, H., Wintoft, P.: Operational forecasts of the geomagnetic Dst index. Geophys. Res. Lett. 29, 34-1–34-4 (2002)
15. Lund Space Weather Center, http://www.lund.irf.se/rwc/dst/models/
16. Mitchell, M.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1996)
17. Pallocchia, G., Amata, E., Consolini, G., Marcucci, M.F., Bertello, I.: Geomagnetic Dst index forecast based on IMF data only. Ann. Geophys. 24, 989–999 (2006)

Comparing Multi-class Classifiers: On the Similarity of Confusion Matrices for Predictive Toxicology Applications

Mokhairi Makhtar, Daniel C. Neagu, and Mick J. Ridley

School of Computing, Informatics and Media, University of Bradford,
Bradford, BD7 1DP, UK
{M.B.Makhtar,D.Neagu,M.J.Ridley}@Bradford.ac.uk

Abstract. Calculating the similarity of predictive models helps to characterize the models diversity and to identify relevant models from a collection of models. The relevant models are considered based on their performance, calculated using their confusion matrix. In this paper, we propose a methodology to measure the similarity for predictive models performances by comparing their confusion matrices. In this research, we focus on multi-class classifiers for toxicology applications. The performance measures of confusion matrices of multi-class classifiers are regrouped into a binary classification problem. Such approach may result in selecting multi-class classifiers with lower False Negative Rate (FNR) for example. Consequently, the methodology for model comparison based on the similarity of confusion matrices provides a working way to select models from a collection of classifiers.

Keywords: Similarity of Confusion Matrices, Classifiers Comparison, Multi-Class Classifiers.

1 Introduction

Predictive models comparison helps in finding how similar models are. But relying only on standard performance indicators such as accuracy may not give much clue on the overall or specific quality of a predictive model. Sometimes the accuracy might be biased for a certain class and this may not provide a good indication of the overall performance for the predictive model. In this case the accuracy is not necessarily the best measurement for predictive models, whereas the confusion matrix is still the most valuable source of performance indicators from classifiers to be analyzed.

Our motivation is given by the need of analyzing the multi-class classifier models for selected classes. In toxicology, we are mostly interested in the toxic class being predicted correctly. Using the confusion matrix as the information source of classifiers performance, we can adapt more useful measurements related to our objective. The classifiers can be either binary class or multi-class models. In our case, we want to predict if the chemical compound is toxic or non-toxic where all our

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 252–261, 2011.

classifiers are in a multi-class format. The multi-class classifiers can be used as binary class models. It is done by combining the multi-class dataset into a new dataset with only binary classes of toxic and non-toxic output [1, 2] and re-generate new predictive models related to the new datasets. But the solution requires much effort in converting datasets to new binary class sets and retraining the models with the new datasets. To be more practical because there are thousands of models in a collection of models, we propose to use the multi-class classifiers confusion matrices as new binary class classifiers confusion matrices. The practical method is to transform the multi-class confusion matrices into binary confusion matrices without updating the datasets and re-generating the models. This will confirm that the original structures and information the predictive models learned remain unchanged. We will demonstrate the proposed technique in section 3.

In this paper we propose a technique to compare multi-class predictive models' performance measures based on confusion matrices. Our methodology addresses model selection, where comparing the classifiers' performance for each class will lead to usefully diverse predictive models for the class of interest from model ensembles.

The rest of the paper is structured as follows: Section 2 presents related work on reducing multi-class into binary class problem. Section 3 defines the technique proposed for comparison of confusion matrices for multi-class (toxicology) models. In Section 4 we introduce and exemplify the technique to calculate the performance measures of output for multi-class predictive models represented by their confusion matrix. Experiments and results are discussed in Section 5. The paper ends with conclusions on current work and further research directions.

2 Reducing Multi-class to Binary Classification Problems

Sometimes, the multi-class classification problems can still be solved with binary classifiers. Such a solution may divide the original multi-class dataset into two class subsets, learning a different binary model for each subset. These techniques are known as binarisation strategies. There are three main approaches: *One-vs-All* (OVA), *One vs-One* (OVO), and *Error Correcting Output Codes* (ECOC) [2].

All of these techniques decompose a complex multi-class to a simpler binary class problem. Hence this strategy may improve the performance because the classifiers have an easier task to distinguish between only two classes rather than many classes.

In this paper we want to investigate whether there are any differences in performance between binarisation strategies by regenerating new binary classifiers from multi-class classifiers. We calculate the performance measures using multi-class classifiers confusion matrices without retraining new binary classifiers.

In the next section, we will discuss on the performance measures related to binary classification classifiers and propose a methodology to reduce multi-class problems to a binary version while calculating the performance measures of the multi-class classifiers with a focus on lower False Negative Rate (FNR) for example, as required in toxicity prediction problems.

3 Performance Measures and Confusion Matrix for Multi-class Classifiers

The confusion matrix contains information about learned and predicted classifications done by a classification model [3]. Table 1 shows the confusion matrix for a two class classifier. The performance measures for two-class classifiers can be calculated from the confusion matrix [3],[4]: sensitivity *TPR = TP/(TP+FN)* is the rate of correct predictions for the positive output (e.g. Yes or True), *FPR = FP/(FP+TN)* is the rate of incorrect predictions for the positive output (e.g. No or False), specificity *TNR =TN/(TN+FP)* is the rate of correct predictions for the negative output, and *FNR = FN / (TP+FN)* is the rate of incorrect predictions for the negative output. Accuracy *ACC = (TP+TN) / (TP+FP+FN+TN)* measures the correct predictions for all classes.

Table 1. Confusion Matrix of Binary Classification: True Positive (TP), True Negative (TN), False Negative (FN) and False Positive (FP)

		Actual	
	Classes	Toxic	Non-toxic
Predicted	Toxic	TP	FP
	Non-toxic	FN	TN

The confusion matrix for a multi-class classification problem is a generalization of the binary case. Below we discuss the properties and the performance measures derived from a multi-class confusion matrix. Table 2 is an example of a multi-class confusion matrix. For the first column (Class A) the intersection with the first row is the True Positive (TP) value for Class A. The sum of values from remaining cells of the column is the False Negative (FN) value for Class A. True positives for second and third columns are the diagonal values of the confusion matrix.

Table 2. Confusion Matrix for a 3-Class Classifier

	Class A	Class B	Class C
Class A	$TP_{A\ (1,1)}$	$e_{AB\ (1,2)}$	$e_{AC\ (1,3)}$
Class B	$e_{BA\ (2,1)}$	$TP_{B\ (2,2)}$	$e_{BC\ (2,3)}$
Class C	$e_{CA\ (3,1)}$	$e_{CB\ (3,2)}$	$TP_{C\ (3,3)}$

The classification accuracy of a multi-class classifier is the ratio of the sum of the principal diagonal values to the total of values in the confusion matrix. If *C* indicates the confusion matrix, the classification accuracy ACC_c can be defined [5] as:

$$ACC_C = \left(\sum_{i=1}^{N} C_{ii} \Big/ \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij} \right) \tag{1}$$

where *N* is the number of classes, *i* refers to the rows index and *j* refers to the columns index for the confusion matrix *C*. The Error Rate (ER) for the classifiers is the complement of the accuracy: *ER = (1 – ACC).*

Beside the accuracy ACC_C and the error rate (*ER*), we can derive other performance measures that can be used to measure multi-class classifiers' quality. Moreover the performance measures of the two-class classification problem can be applied by regrouping the multi-class confusion matrix into two-class classification measures.

In predictive toxicology applications, there is more interest in the false negative rate (*FNR*) measurement of the cases the model fails to correctly classify the instances to the appropriate classes. To give more flexibility for such applications for multi-class classifiers comparison, we propose that the positive (toxic) class and negative (non-toxic) class can be selected by regrouping them into a two class problem. Furthermore this technique is also highly recommended in classifier ensembles where good combination of classes and models will make the binary prediction more accurate [5, 6].

The performance measures for the positive (toxic) class in predictive multi-class classifiers are described below. Let's say the selected toxic classes are Class A (e.g. Very Toxic, column 1) and Class B (e.g. Toxic, column 2) in Table 2. The selected class indexes are stored into the one-row vector *V*. Thus *V= (1, 2)*. The proposed *TPR* and *FNR* measures for the selected classes are as follow:

$$TPR_{Ma} = \left(\sum_{\substack{x=1,\\ j=V_x}}^{C} \sum_{\substack{y=1,\\ i=V_y}}^{C} R_{ij} \Bigg/ \sum_{\substack{x=1,\\ j=V_x}}^{C} \sum_{i=1}^{N} R_{ij} \right) \tag{2}$$

$$FNR_{Ma} = \left(\sum_{\substack{x=1,\\ j=V_x}}^{C} \sum_{\substack{y=1,\\ i\neq V_y}}^{N} R_{ij} \Bigg/ \sum_{\substack{x=1,\\ j=V_x}}^{C} \sum_{i=1}^{N} R_{ij} \right) \tag{3}$$

where: *N* is the number of samples of all classes in the confusion matrix *R*, *C* is the number of selected class samples for the confusion matrix *R*, *i* is the row index in the confusion matrix *R*, *j* is the column index in the confusion matrix *R*, *x* and *y* are counters for columns and rows, and *V* is a vector of selected class indexes.

The performance measures for the non-toxic class, False Positive Rate (*FPR*) and True Negative Rate (*TNR*), can be derived by adapting equation (2) and equation (3).

Table 3. Confusion Matrix (M_{M1}) for Model M1

	Class A	Class B	Class C
Class A	$10_{(1,1)}$	$21_{(1,2)}$	$33_{(1,3)}$
Class B	$24_{(2,1)}$	$53_{(2,2)}$	$26_{(2,3)}$
Class C	$17_{(3,1)}$	$18_{(3,2)}$	$19_{(3,3)}$

Let's say Model M1 produced a confusion matrix M_{M1} (see Table 3). Referring to the equation 2 and equation 3, we demonstrate how to calculate the *TPR* and *FNR* of toxic classes. For these examples we select two classes as toxic classes (Class A and Class B). The index for Class A is 1 and the index for Class B is 2. Thus the vector

$V = (1, 2)$. For example, from Table 3: $TPR_{M1} = ((10 + 24) + (21+53)) / ((10+24+17) + (21+53+18)) = 0.76$ and $FNR_{M1}= ((17) + (18)) / ((10+24+17) + (21+53+18)) = 0.24$.

From the results above, *TPR* and *FNR* complement each other in the confusion matrix. In the next section we will demonstrate the methodology to measure the similarity between confusion matrices for multi-class classifiers.

4 Similarity of Confusion Matrices for Multi-class Classifiers

In this section, we apply the technique proposed in [1] to compare multi-class classifiers' confusion matrices**.** Let's say we have three predictive models generated by different classifiers using the same dataset. The model M1 generates the confusion matrix M_{M1} (see Table 3), the model M2 generates the confusion matrix M_{M2}, and the model M_3 generates confusion matrix M_{M3} (see Table 4).

Table 4. Confusion Matrices for Model M2 and Model M3

Confusion matrix for model M2			
Class	A	B	C
A	$24_{(1,1)}$	$18_{(1,2)}$	$33_{(1,3)}$
B	$10_{(2,1)}$	$53_{(2,2)}$	$19_{(2,3)}$
C	$17_{(3,1)}$	$21_{(3,2)}$	$26_{(3,3)}$

Confusion matrix for model M3			
Class	A	B	C
A	$34_{(1,1)}$	$4_{(1,2)}$	$9_{(1,3)}$
B	$10_{(2,1)}$	$80_{(2,2)}$	$10_{(2,3)}$
C	$7_{(3,1)}$	$8_{(3,2)}$	$59_{(3,3)}$

Table 5 shows the performance measures *TPR*, *FNR* and *ACC* calculated using equations 1, 2 and 3. The values of performance measures were calculated by grouping the selected toxic classes A and B. Thus, $V = (1, 2)$. From the results depicted in Table 5, model M3 is the better model compared to M1 and M2: *TPR* is the highest value and *FNR* is the lowest value for model M3.

Table 5. Performance Measures (*TPR* and *FNR*) for Models M1, M2 and M3

Models	*TPR*	*FNR*	*ACC*
M1	0.76	0.25	0.37
M2	0.73	0.27	0.47
M3	0.90	0.10	0.78

For the similarity measurement, in this example we chose *FNR* to measure the distance between the models' performances. For the performance measures in Table 5, let's use the notations $k_{1...n}$. In this case k_1 is *FNR*. The following steps illustrate the calculation of the distance between the confusion matrices between two predictive models:

Step 1: Save the selected performance measure/s in a 1-dimension (vector).

We save the selected performance measures into two rows vectors; in this case the vectors for M1 (V_{M1}) and M2 (V_{M2}) have just 1 element: V_{M1} = (0.25) and V_{M2} = (0.27).

Step 2: Calculate the distance between the vectors.

The distance between the vectors V_{M1} and V_{M2} is calculated using the Euclidean Distance. The distance O (Output) between model *M1* and model *M2* is the average of distances between the confusion matrix elements. Similarity and distance measures are complementary. In our case, the similarity of output *O (SimO)* between two models will be:

$$SimO_{(M1,M2)} = 1 - \left(\sqrt{\sum_{k=1}^{n} (V_{M1k} - V_{M2k})^2} \Big/ n \right) \tag{4}$$

where: k is the order of performance measures selected, n equals to number of k, V_{M1} is the index vector for model *M1*, and V_{M2} is the index vector for model *M2*. The value for $SimO_{(M2,M2)}$ in the example above is 0.98. Table 6 contains the values for $SimO_{(M1,M2)}$ related to the similarity of the three classifiers using *FNR*. The result shows that models M1 and M2 are 98% similar on their FNR.

Table 6. Similarity Matrix for models M_1, M_2 and M_3

	M_1	M_2	M_3
M_1	1	0.98	0.85
M_2	0.98	1	0.83
M_3	0.85	0.83	1

5 Experiments and Results

For this study, we generated collections of models using a series of classification algorithms implemented in Weka [7], such as k-nearest neighbors classifier (weka.classifiers.lazy.IBk), decision trees (weka.classifiers.trees.J48) and numerical prediction algorithms (weka.classifiers.rules.JRip). The predictive models were applied to various toxicology data sets such as Demetra [8] (Bee, Daphnia, Oral Quails, Dietary Quails and Trout) and TETRATOX [9]. Each dataset had originally more than two classes to predict the toxicity levels for each compound. Table 7 is an example of the confusion matrix. We mapped the old multi-classes onto binary classes. We want to study how the relation of different class categories will affect the performance of classifier algorithms (refer to Table 12 in [10]).

Over 1,300 predictive models were generated with different combinations of datasets, algorithms, and model parameters. The feature selection algorithm applied to the original full datasets was Correlation-based Feature Selection (CFS). We used feature selection to find sets of attributes that are highly correlated with the target classes [10, 11]. Each data set was processed using Weka with 10-fold cross validation and classifiers weka.classifiers.lazy.IBk, weka.classifiers.trees.J48, and weka.classifiers.rules.Jrip). In Table 7 the confusion matrix for a decision tree applied to the Bee dataset with 5 classes is provided.

Table 7. A confusion matrix generated using multi-class dataset with feature selection (CFS), 10-fold cross validation and using classifiers (weka.classifiers.trees.J48)

	Class1	Class2	Class3	Class4	Class5
Class1	7	4	2	3	0
Class2	4	7	4	8	2
Class3	0	2	1	4	0
Class4	2	10	4	23	4
Class5	0	0	2	4	8
Total Instances	13	23	13	42	14

Considering the fusion of Class1, Class2 and Class3 as toxic classes, the performance for a randomly chosen model M154c are as follows:

Table 8. Performance measures calculated based on the confusion matrix using table 7

Performance Measures	Results
TPRate (All Classes) and Accuracy (See Eq. 1 and 2)	0.44
Error Rate (All Classes)	0.56
FNRate (selected toxic class; 1, 2, 3, 4) (See Eq. 3)	0.07

Experiment 1:

In this experiment we want to compare the use of error rate for all classes vs. false negative rate for selected toxic classes in multi-class classifiers. We are interested in FNR because in predictive toxicology good models should have lower rate of false negatives (FN) for toxic class results. For Table 9 (ER vs. FNR results measured using the selected classes) we can find that models with similar ER Rate can exhibit a range of FNR values:

Table 9. Error Rate (ER) and FNR of multi-class classifiers applied to the DEMETRA datasets

Datasets	Toxic Classes (Low FNR)	All Classes (ER)	Toxic Classes (High FNR)	All Classes (ER)
Bee	0.04 *–M304c*	0.60 *–M304c*	0.12 *-M1c*	0.61 *-M1c*
Daphnia	0.07 *-M334c*	0.56 *-M334c*	0.20 *–M31c*	0.56 *–M31c*
Dietary Quail	0.19 *–M364c*	0.59 *–M364c*	0.25 *-M211c*	0.61 *-M211c*
Oral Quail	0.30 *-M91c*	0.60 *-M91c*	0.52 *-M244c*	0.61 *-M244c*
Trout	0.12 *M271c*	0.51 *-271c*	0.17 *-M274c*	0.52 *-M274c*

Experiment 2:

For the second experiment, we want to study if the relationship between the numbers of toxic classes will affect the performance of the classifier. In this experiment we mapped the toxic class into two categories: binary class (Toxic and Non-toxic) and multi-class (class A, class B .. class N). From the results shown in Table 10 we conclude that:

- Datasets with feature selection algorithms (such as CFS) applied are better in FNR performance measurement compared to datasets with no feature selection. Examples of such models are *M4a* and *M1a*.
- The classifiers perform best in Bee dataset and worst in Oral Quail dataset.
- Some performance (FNR) of models with selected class for more than 1 toxic class (e.g. *M4c*) is poor compared to binary model with only 1 toxic class (e.g. *M4a*), but in contrast some of the multi-class classifiers are better than binary classifiers (e.g. M34c vs. M34a and M271c vs. M271a).
- On average, models that applied binarisation strategies (models number ended with '*a*') are better than multi-class classifiers that apply calculation of FNR to their confusion matrices (models named ending in '*c*'). This proved that multi-class classifiers for Daphnia datasets such as M334c are better than binary classifiers (e.g. M331a). For Oral Quail dataset, both binary and multi-class were having the same performance (0.30) of FNR (e.g. M91c vs. M244a).

From the results shown in Table 10, if the objective is to discriminate between two binary classes, in our case Toxic and Non-toxic, then the classifiers with binary class format have better performance compared to multi-class classifiers. But the difference between both categories is very small (between 0.02 – 0.04). For some models, regrouping classes in a single toxic class may increase the accuracy as compared to re-generating binary class classifiers.

Table 10. Results of FNR for all datasets with feature selection algorithms (CFS) and without CFS generated (None) using classifiers (IBK, J48 and JRip)

	IBK	J48	JRip
Datasets	FNR –Model_ID	FNR – Model_ID	FNR –Model_ID
Bee (None)	0.12 – *M1a* 0.12 – *M1c*	0.06 - *M151a* 0.09 - *M151c*	0.06 - *M301a* 0.04 – *M301c*
Bee (CFS)	0.04 – *M4a* 0.11 – *M4c*	0.02 - *M154a* 0.07 - *M154c*	0.04 - *M304a* 0.04 – *M304c*
Daphnia (None)	0.19 – *M31a* 0.20 – *M31c*	0.19 - *M181a* 0.20 - *M181c*	0.10 - *M331a* 0.11 – *M331c*
Daphnia (CFS)	0.20 – *M34a* 0.16 - *M34c*	0.12 - *M184a* 0.14 - *M184c*	0.12 - *M334a* 0.07 – *M334c*
Dietary Quail (None)	0.19 - *M61a* 0.19 - *M61c*	0.20 - *M211a* 0.25 - *M211c*	0.23 - *M361a* 0.24 – *M361c*
Dietary Quail (CFS)	0.15 - *M64a* 0.19 - *M64c*	0.13 - *M214a* 0.15 - *M214c*	0.20 - *M364a* 0.19 – *M364c*
Oral Quail (None)	0.32 - *M91a* 0.30 - *M91c*	0.36 - *M241a* 0.34 - *M241c*	0.54 - *M391a* 0.62 – *M391c*
Oral Quail (CFS)	0.37 - *M94a* 0.36 - *M94c*	0.30 - *M244a* 0.52 - *M244c*	0.47 - *M394a* 0.61 – *M394c*
Trout (None)	0.14 - *M121a* 0.16 - *M121c*	0.17 - *M271a* 0.12 - *M271c*	0.10 - *M421a* 0.09 – *M421c*
Trout (CFS)	0.12 - *M124a* 0.14 - *M124c*	0.07 - *M274a* 0.17 - *M274c*	0.05 - *M424a* 0.12 – *M424c*

Experiment 3:

In this experiment, models from Table 10 were selected to calculate their similarity. From the results in Table 11 we can see that the models have a large spread of performance value of FNR. The similarity values between confusion matrices shows that similar FNR values between models indicate similar performance among them although using different classifier algorithms. Example of such models are model M4a and model M304a, and model M31c and model M181c.

However the results only show a single element of the similarity evaluation for predictive models' performance. To have more accurate results of similarity of predictive models, the comparison of multi-class confusion matrices can be applied using our proposed methodology for calculating the similarity of binary predictive models [1].

Table 11. Similarity Matrix for Models (M4a, M304A, M151c and M154c)

Model ID	M4a	M304a	M151c	M154c
M4a	1	1	0.95	0.97
M304a	1	1	0.97	0.97
M151c	0.95	0.97	1	0.98
M154c	0.97	0.97	0,98	1

6 Conclusions

This study shows that comparing predictive models' confusion matrices will help users to choose similar models based on FNR performance measure. We studied whether there are any differences in performance measures between binarisation strategies by converting the multi-class datasets into binary classes, compared to calculating the performance measure on the fly using their confusion matrices.

From the experiments presented, regrouping multi-class classifiers' confusion matrices to binary problem is a simple solution to analyze and categorize the performance of the multi-class classifiers from a collection of models. This methodology can be integrated in ensembles of classifiers by further analysing diversity of classes of selected models.

Our experiments also show that the similarity of confusion matrices will help for further analysis and customized selection of the relevant models according to the user's needs. In future, we will integrate the methodology in a models management system and evaluate various ways to characterize and use their performances.

Acknowledgments. This work is partially supported by BBSRC, TSB and Syngenta through the Knowledge Transfer Partnerships (KTP) Grant "Data and Model Governance with Applications in Predictive Toxicology". The first author acknowledges the financial support received from the University Sultan Zainal Abidin (UniSZA), Malaysia.

References

1. Makhtar, M., Neagu, D.C., Ridley, M.J.: Binary classification models comparison: On the similarity of datasets and confusion matrix for predictive toxicology applications. In: Khuri, S., Lhotská, L., Pisanti, N. (eds.) ITBAM 2011. LNCS, vol. 6865, pp. 108–122. Springer, Heidelberg (2011)
2. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. J. Pattern Recognition 44, 1761–1776 (2011)
3. Kohavi, R., Provost, F.: Glossary of Terms. Editorial for the Special Issue on Applications of Machine Learning and the Knowledge Discovery Process. J. Machine Learning 30, 271–274 (1998)
4. Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers. HP Laboratories, http://www.hpl.hp.com/techreports/2003/HPL-2003-4.pdf
5. Prasanna, S.R.M., Yegnanarayana, B., Pinto, J.P., Hermansky, H.: Analysis of Confusion Matrix to Combine Evidence for Phoneme Recognition. IDIAP Research Report, IDIAP-RR-27-2007 (2007)
6. Freitas, C.O.A., Carvalho, J.M.D., Jose Josemar Oliveira, J., Aires, S.B.K., Sabourin, R.: Confusion Matrix Disagreement for Multiple Classifiers. In: Proceedings Of The Congress On Pattern Recognition 12th Iberoamerican Conference on Progress in Pattern Recognition, Image Analysis and Applications, pp. 387–396 (2007)
7. Witten, I.H., Frank, E., Trigg, L., Hall, M., Holmes, G., Cunningham, S.J.: Weka: Practical Machine Learning Tools and Techniques with Java Implementations. In: Proceedings of the ICONIP/ANZIIS/ANNES 1999 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems, pp. 192–196 (1999)
8. DEMETRA Project, http://www.demetra-tox.net/
9. TETRATOX, http://www.vet.utk.edu/TETRATOX/index.php
10. Neagu, D., Guo, G.: A Data-Driven Approach for Improved Effective Classification in Predictive Toxicology. In: Proceeding of IEEE International Conference on Computational Cybernetics ICCC 2006, pp. 193–198 (2006)
11. Trundle, P.: Hybrid Intelligent Systems Applied to Predict Pesticides Toxicity - a Data Integration Approach. PhD Thesis. School of Informatics. University of Bradford, UK (2008)

Modeling Design and Flow Feature Interactions for Automotive Synthesis

Michael Rath and Lars Graening

Honda Research Institute Europe,
Carl-Legien-Straße 30, 63073 OffenbachMain, Germany
lars.graening@honda-ri.de
http://www.honda-ri.de

Abstract. In the automotive industry Computational Fluid Dynamics (CFD) simulations have become an important technology to support the development process of a new automobile. During that process, individual simulations of the air flow produce a huge amount of information about the design characteristic, where mostly only a minority of information is used. At the same time knowledge about the relationship between design modifications and their aerodynamic consequences provides valuable insight into the entire aerodynamic system. In this work a computational framework is introduced, providing means to identify relevant interactions within the aerodynamic system based on existing design and flow data. For an efficient modeling, the raw flow field data is reduced to a set of relevant flow features or phenomena. Applying interaction graphs to the aerodynamic data set unveils interacting and redundant structures between design variations and observed changes of flow phenomena. The general framework is applied to an exemplary aerodynamic system representing a 2D contour of a passenger car.

Keywords: Data Mining, Structural Modeling, Information Theory, Interaction Information, Aerodynamic Design, Flow Field Feature.

1 Introduction

In the automotive industry computer aided engineering (CAE) tools have become an important technology for improving the design development process. Physical experiments are replaced by computational tools to reduce development costs, see [10]. Hundreds and thousands of different geometric models of the designs and flow fields are simulated before an actual physical model of a design is build. However, usually the resulting flow field is reduced to a single number, defining the performance of the design. In order to get a deeper insight into the aerodynamic system at hand, we developed a framework for identifying relevant interaction structures between shape, flow field phenomena and performance, based on these otherwise unused data.

After reviewing related research activities in the subsequent section, the detailed framework proposed is depicted in section 3, including details about data

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 262–270, 2011.

reduction, the modeling of the interaction structure and the visualization of the results. The general framework has been applied to an exemplary aerodynamic system modeling the 2D contour of a passenger car, depicted in section 4.

2 Related Work

In the domain of aerodynamics, computational approaches for data mining and knowledge extraction are rarely being reported. Nevertheless, most of the existing works approach the modeling of the relationship between design variations and performance only. As an example, Obayashi et. al [18,1] utilized self-organizing map (SOM) and analysis of variance (ANOVA) techniques to identify relevant relationships between design and performance, and Graening et. al [4] introduced a knowledge extraction framework discovering a set of If-Then rules, which illuminate causal relations between design modifications and performance changes. Further, the need of a universal geometric design representation within the framework is stated. However, the flow field produces much more information about the design concept than kept in the performance number. Nevertheless, it is impossible to handle the whole flow field. Therefore we extract flow field features commonly used for the visualization of flow field data. An overview of the sate of the art in flow visualization are given by [19]. De-composition methods, like the proper orthogonal decomposition (POD) [14], are applied to reduce the dimensionality of the flow field while keeping the majority of the energy contained in the flow. Often, decomposition methods are used in a pre-processing step, e.g. before flow phenomena like vortices are extracted, see [6]. While decomposition methods come up with features that have no direct physical meaning, other researchers aim at explicitly quantifying the position, rotation and elongation of physical artifacts like vortices or attachment and detachment lines [11]. Due to the similarity to optical flow, some attempts are derived from the computer vision domain to detect flow patterns, e.g. see [20]. In the context of applying data mining technologies to flow field features, Depardon et. al [3] use multi-dimensional scaling (MDS) for the classification of flow topologies. However, the relationship between flow fields and design properties has not been considered.

3 Interaction Modeling Framework

Based on earlier work from Graening et al. [5], the authors aim at deriving a general framework for identifying interaction structures in aerodynamic systems. Given the design process (e.g. targeting the development of a new car design), various design shapes together with its related flow field data are generated. Each of the shapes is represented by a low dimensional geometric representation of the actual continuous surface. Spline surfaces like NURBS (Non-uniform Rational B-Spline) or FFD (Free-From Deformation) are exemplary representations often being used. In a pre-processing step the geometric representations have to be unified and reduced to managable number of design features. Given the surface representation and the size of the computational area in which to model the flow,

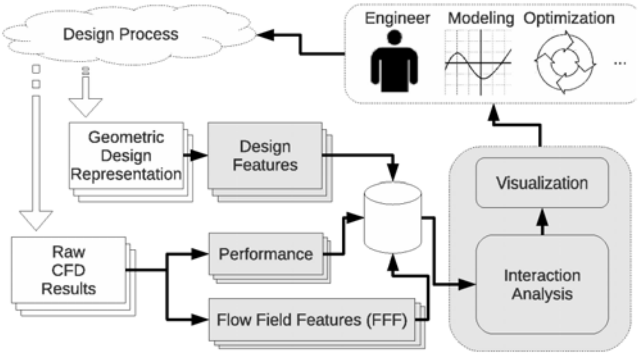


Fig. 1. Overall framework for the identification of structural interaction patterns with an aerodynamic system

a discrete volume mesh is generated discretizing the computational area, with a typical mesh size in the order of 10^6 or more cells. The volume mesh together with the pre-defined wall conditions make up the initial setup for the CFD simulator. After simulation, detailed local information about the flow direction, velocity, pressure, temperature and so on is available. The resulting amount of data is too huge for an adequate analysis of flow effects and its relation to design parameter variations. A feature extraction step is introduced reducing the raw flow field to a low dimensional representation as depicted in Fig. 1. The choice of the flow field features strongly depend on the given task. Alongside the flow features, individual performance indicators are calculated, to quantify the overall quality of the shape.

After the pre-processing, interaction analysis is applied to identify the intrinsic structure of the aerodynamic system at hand. Based on a probabilistic attempt from information theory, described in the following section, the most relevant structures in the system of design features, flow features and performance are identified. Finally, interaction graphs are deployed to transfer the extracted information to the aerodynamic engineer, thus influencing the design process.

3.1 Information Theoretic Interactions

Following Krippendorff [12], we define interaction as *a unique dependency from which all relations of lower ordinality are removed*. Information theoretic attempts for quantifying interactions are founded based on the formulation of the Shannon entropy. Given a discrete random variable X_i, the Shannon entropy, denoted as $H(X_i)$, is a non-negative measure of uncertainty quantifying the amount of randomness contained in X_i. It is formally defined as: $0 \leq H(X_i) = -\sum_{n=1}^{N} p(x_n) \log p(x_n) \leq \log N$, with N being the number of discrete intervals. The logarithm is commonly chosen with the base two, resulting in $H(X_i)$ being measured in bits.

For two variables X_i and X_j, the mutual information [2], $I(X_i; X_j)$, 2-way interaction or transmitted information can be considered as the amount of information shared among both variables. The mutual information can be written as the difference between maximum entropy, assuming independence among the variables, and the actual joint entropy [12] observed:

$$I(X_i; X_j) = H(X_i) + H(X_j) - H(X_i, X_j). \tag{1}$$

$I(X_i; X_j)$ is only equal to zero if X_i and X_j are statistical independent. In case that a dependency between X_i and X_j exists, $I(X_i; X_j)$ is always larger zero, bounded by the maximum of the marginal entropies $H(X_i)$ and $H(X_j)$. Based on the work of McGill [15], Jakulin and Bratko [7] defined the interaction information for multiple attributes. The interaction information for three variables X_1, X_2 and X_3 evaluates the information gain resulting from the 3-way interaction which is not present in any of the 2-way interactions:

$$I_J(X_1; X_2; X_3) = I(X_1, X_2; X_3) - I(X_1; X_3) - I(X_2; X3). \tag{2}$$

$I_J(X_1; X_2; X_3)$ quantifies the information shared among the variables X_1 and X_2 with X_3, reduced by the information shared between variable X_1, X_3 and X_2, X_3. It is important to note that in contrast to the mutual information the interaction information can have negative values. $I_J(X_1; X_2; X_3)$ gets negative if the joint information of X_1 and X_2 on X_3 is smaller than the product of $I(X_1; X_3)$ and $I(X_2; X3)$. This is the case if the information added to the system through interaction is smaller compared to the amount of redundant information in the system that e.g. X_1 and X_2 have in common about X_3, see [13]. In consequence, $I_J(X_1; X_2; X_3) = 0$ not necessarily reflects the absence of an interaction rather that information and redundancy cancel each other out. However, a positive I_J indicates a surplus of information due to interaction and a negative I_J indicates a surplus of redundancy. Higher order interactions (larger three), as defined by Jakulin [8], are not considered throughout this study.

3.2 Interaction Graph

An interaction graph [7] is a visualization of the identified interaction structure from observed variables characterizing the system. In this work we adopt supervised graph structures visualizing 2-and 3-way interactions relative to an a priori chosen dependent variable Y with the target to explain the uncertainty of the dependent variable. Thus, all information quantities are normalized by the uncertainty of the dependent variable, $H(Y)$. The graph consists knots labeled by the relative mutual information $I(X_i, Y)/H(Y)$ and edges between two nodes of X_i and X_j corresponding to the relative interaction information $I_J(X_i, X_j, Y)/H(Y)$. Edges with a negative relative interaction information are drawn with dashed lines and with solid lines otherwise. For the sake of readability, only the most significant 3-way interactions are drawn into an interaction graph.

4 Application to a Passenger Car Model

The introduced framework is applied to a two-dimensional model of a passenger car to unveil interactions between car shape deformations, flow field features and performance measures. A parametric Free-Form Deformation (FFD) [21] with 20 control point parameters (design parameters), see Fig. 2, is applied to model the design contour of the car. 1000 variations of the car design are generated by applying a latin hypercube sampling (LHS) [16] to the design parameters. The computational fluid dynamics simulation of the passenger car is carried out by OpenFOAM®[1]. For each car, the stationary flow field is computed using the simpleFOAM solver, which iteratively computes the steady-state solution of the incompressible Navier-Stokes equations.

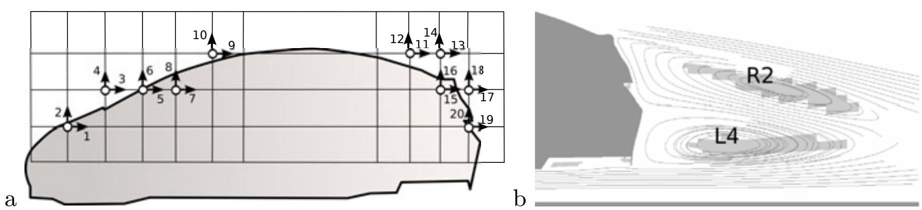


Fig. 2. a: Contour of the initial passenger car together with the FFD control point lattice, defining the position of the control point parameters used to vary the car shape, b: Visualization of the identified upper $R2$ and lower $L4$ wake vortex behind the car

4.1 Flow Feature Extraction and Performance Evaluation

The resulting flow fields are reduced to a small set of flow features and performance indicators. A common objective in car design is to reduce the drag force D, acting opposite to the flow direction, while reducing lift force L, perpendicular to the drag force component. Beside drag and lift and without loss of generality our performance indicator is defined by a superposition of lift and drag:

$$Q = \frac{L}{\sqrt{\mathrm{var} L}} + \frac{D}{\sqrt{\mathrm{var} D}}. \tag{3}$$

The major part of drag on road vehicles (form drag) is the result of flow separations [9], especially at the aft section of the vehicle. The flow separation at the back of the car is always attended by an upper and lower vortex sheet behind the vehicle. Hence, the size and orientation of the emerging vortices is linked to an change of the drag value. Vortices are characterized by discontinuities observed in the vector field. To identify those discontinuities the vortex identification algorithm of Michard [17] has been adopted,

[1] OpenFOAM: open source CFD, http://www.openfoam.com/

$$\Gamma_1(P) = \frac{1}{N} \sum_{S} (\theta_M) \tag{4}$$

The dimensionless scalar $\Gamma_1(P)$ integrates over the angles θ_M between the velocity vectors for each point $M \in S$, where S defines the neighborhood around P and the vector PM. $|\Gamma_1|$ becomes close to one in the range of vortex centers. Applying a threshold to the calculated values of $|\Gamma_1|$ allows to identify vortex regions as shown in Fig. 2b. Gray areas indicate the position of the upper $R2$ and lower wake vortex $L4$ behind the car. The results of Γ_1 are used to model the vortices with ellipses, estimating the orientation, the size and eccentricity of the vortices. For all 1000 generated flow fields the corresponding vortices between different flow fields are identified using the Euclidean distance between the vortex centers. Flow fields that have no correspondence in any of the generated flow fields have to be treated differently what is not considered in this work. We limit the following studies to the vortices $R2$ and $L4$, where a correspondence for any flow field is found.

4.2 Interaction Analysis

Given the data of all 1000 designs the interactions are modeled and visualized using interaction graphs. Only the design parameter modifying the design in y-direction have be considered in this study. The interaction graphs for modeling the relationship between a: design features and drag D, b: design features and Q, c: design features and the size of the upper wake vortex $R2$ as well as flow features and Q are depicted. Regarding the influence of the design parameters d_i on the drag D, Fig. 3a, variations of the rear part of the car have a stronger influence compared to variations on the frontal part. Especially parameter $d18$ and $d12$ share a majority of information with the drag. $d18$ already explains about 26% of $H(D)$. Further, the interaction between $d12$ and $d18$ seems to be relevant for explaining variations in drag. Interactions between the frontal and rear part of the car model seem to be negligible. The interaction graph concerning lift L, not presented here, provides qualitative similar results as for the drag. Concerning the influence of the flow field features on the drag, the size of the upper wake vortex $R2$ turned out to be most relevant. The interaction structure for the size of $R2$, Fig. 3c, is qualitatively the same as, Fig. 3a, what consolidates the importance of the flow feature. Interestingly, in all interaction graphs comprising design parameters, no strong redundant connections are observed, possibly due to the chosen representation, where the influence of individual design parameter on the shape does not overlap.

While usually the effect of design variations and flow phenomena is studied regarding aerodynamic properties like drag and lift, the influence on combined objectives like the performance measure Q is seldom be regarded. The graphs showing the interaction structure between design features and Q as well as between flow features and Q are depicted in Fig. 3b and d respectively. Comparing the interaction graph from Fig. 3a, showing the interactions between design parameters and D, with the interaction graph from Fig. 3b, interactions are of

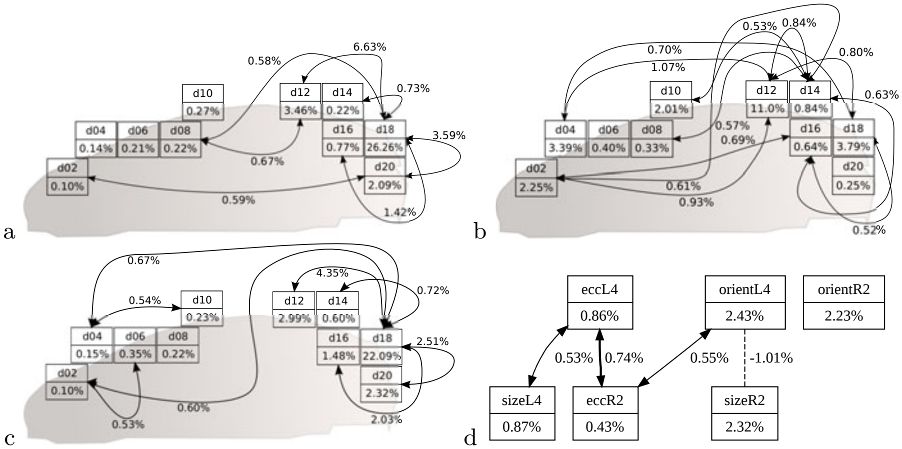


Fig. 3. Resulting interaction graphs modeling the interactions between a: design parameter and drag D, b: design parameter and objective Q, c: design parameter and the size of the upper wake vortex $R2$, d: flow features and the objective Q.

relevance which are not important for either of the individual objectives D or L. E.g. the influence of $d04$, located at the frontal part of the car, becomes more relevant. Please consider that the graph for L is not shown here but is qualitative similar to the interaction graph for D. Further, the frontal part and its interaction with the rear part of the car becomes more relevant concerning the combined objective Q. Finally, the interrelation between the flow features and Q are under investigation. Fig. 3d shows that the orientation of the lower wake vortex $L4$, the orientation of $R2$ and the size of $R2$ seemingly have a strong influence on Q. Redundancy between the orientation of $L4$ and the size of $R2$ are observed. This might be the result of interference between the lower and upper vortex for certain configurations of the size of $R2$ and the orientation of $L4$.

Overall, the interaction analysis results in hypothesis about the relationship between design-, flow properties and objectives valuable for the ongoing design process, once consolidated with real physical experiments. E.g. with the given knowledge, selected design parameters can be modified for manipulating distinct flow phenomena with respect to a given objective.

5 Conclusion

In this work we presented a general framework for identifying interaction structures in aerodynamic systems, by deploying techniques from information theory. For the investigation of aerodynamic systems that constitute high-fidelity flow simulations, the flow field data has to be reduced to a manageable amount of flow features before modeling interactions. As pointed out, the choice of flow features depends on the defined objective. The framework is applied to the automotive domain, exemplary to the 2D contour of a passenger car model.

Investigating the interactions between design-, flow features and objectives provides valuable knowledge about the aerodynamic system. The knowledge can directly be utilized by the design process by filtering design parameters and interactions affecting distinct flow features relevant for a defined objective. The application of the framework to 3D non-stationary flow field data is considered for future work. This requires to discover different techniques for identifying and tracking flow field features and discrete information like the absence of a flow feature has to be processed properly. Finally, the result of the interaction analysis need to be consolidated by real physical experiments.

References

1. Chiba, K., Jeong, S., Obayashi, S., Morino, H.: Data mining for multidisciplinary design space of regional-jet wing. In: IEEE Congress on Evolutionary Computation, vol. 3, pp. 2333–2340 (2005)
2. Cover, T.M., Thomas, J.A., Wiley, J.: Elements of information theory, vol. 1. Wiley Online Library (1991)
3. Depardon, S., Lasserre, J., Brizzi, L., Bore, J.: Automated topology classification method for instantaneous velocity fields. Experiments in Fluids 42, 697–710 (2007)
4. Graening, L., Menzel, S., Hasenjäger, M., Bihrer, T., Olhofer, M., Sendhoff, B.: Knowledge extraction from aerodynamic design data and its application to 3d turbine blade geometries. Mathematical Modelling and Algorithms 7, 329–350 (2008)
5. Graening, L., Olhofer, M., Sendhoff, B.: Interaction detection in aerodynamic design data. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 160–167. Springer, Heidelberg (2009)
6. Graftieaux, L., Michard, M., Grosjean, N.: Combining PIV, POD and vortex identification algorithms for the study of unsteady turbulent swirling flows. Measurement Science and Technology 12, 1422 (2001)
7. Jakulin, A.: Machine learning based on attribute interactions. Unpublished doctoral dissertation. University of Ljubljana 28, 252 (2005) (retrieved March)
8. Jakulin, A., Bratko, I.: Quantifying and visualizing attribute interactions. Arxiv preprint cs/0308002 (2003)
9. Katz, J.: Race Car Aerodynamics: Designing for Speed. Bentley Publishers (1995)
10. Keane, A., Nair, P.: Computational Approaches for Aerospace Design: The Pursuit of Excellence. Wiley, Chichester (2005)
11. Kenwright, D.N.: Automatic detection of open and closed separation and attachment lines. In: Proceedings of Visualization 1998, pp. 151–158. IEEE, Los Alamitos (2002)
12. Krippendorff, K.: Information theory: structural models for qualitative data. Sage Publ., Thousand Oaks (1986)
13. Krippendorff, K.: Information of interactions in complex systems. International Journal of General Systems 38, 669–680 (2009)
14. Liang, Y.C., Lee, H.P., Lim, S.P., Lin, W.Z., Lee, K.H., Wu., C.G.: Proper orthogonal decomposition and its application - Part I: Theory. Journal of Sound and Vibration 252(3), 527–544 (2002)
15. McGill, W.: Multivariate information transmission. IRE Professional Group on Information Theory 4(4), 93–111 (2002)

16. McKay, M.D., Beckman, R.J., Conover, W.J.: A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 42(1), 55–61 (2000)
17. Michard, M., Graftieaux, L., Lollini, L., Grosjean, N.: Identification of vortical structures by a non local criterion-Application to PIV measurements and DNS-LES results of turbulent rotating flows. In: 11th Symposium on Turbulent Shear Flows, Grenoble, France, pp. 25–28 (1997)
18. Obayashi, S., Sasaki, D.: Visualization and data mining of pareto solutions using self-organizing map. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 796–809. Springer, Heidelberg (2003)
19. Post, F.H., Vrolijk, B., Hauser, H., Laramee, R.S., Doleisch, H.: Feature extraction and visualization of flow fields. In: Eurographics 2002 State-of-the-Art Reports, pp. 69–100 (2002)
20. Schlemmer, M., Heringer, M., Morr, F., Hotz, I., Hering-Bertram, M., Garth, C., Kollmann, W., Hamann, B., Hagen, H.: Moment invariants for the analysis of 2D flow fields. IEEE Transactions on Visualization and Computer Graphics, 1743–1750 (2007)
21. Sederberg, T.W., Parry, S.R.: Free-form deformation of solid geometric models. ACM Siggraph Computer Graphics 20(4), 151–160 (1986)

The Importance of Precision in Humour Classification

Joana Costa[1], Catarina Silva[1,2], Mário Antunes[1,3], and Bernardete Ribeiro[2]

[1] Computer Science Communication and Research Centre
School of Technology and Management, Polytechnic Institute of Leiria, Portugal
{joana.costa,catarina,mario.antunes}@ipleiria.pt
[2] Department of Informatics Engineering,
Center for Informatics and Systems of the University of Coimbra (CISUC), Portugal
{catarina,bribeiro}@dei.uc.pt
[3] Center for Research in Advanced Computing Systems (CRACS), Portugal

Abstract. Humour classification is one of the most interesting and difficult tasks in text classification. Humour is subjective by nature, yet humans are able to promptly define their preferences.

Nowadays people often search for humour as a relaxing proxy to overcome stressful and demanding situations, having little or no time to search contents for such activities. Hence, we propose to aid the definition of personal models that allow the user to access humour with more confidence on the precision of his preferences.

In this paper we focus on a Support Vector Machine (SVM) active learning strategy that uses specific most informative examples to improve baseline performance. Experiments were carried out using the widely available Jester jokes dataset, with encouraging results on the proposed framework.

Keywords: Support Vector Machine, Active Learning, Text Classification, Humour classification.

1 Introduction

Humour classification is one of the most interesting and difficult tasks in text classification. However, despite the attention it has received in fields such as philosophy, linguistics, and psychology, there have been few attempts to create computational models for humour classification [1].

Modern societies turn human course of life a *fast forward* version of itself. It is not only overwhelming work times, but also the pressure they convey. Most people feel that there is not enough time to de-stress, and even when there is, the mind is constantly being overstimulated by the mass media, that take part of everyday life and expose people to so much information.

While it is merely considered a way to induce amusement, humour also has a positive effect on the mental state of those using it and has the ability to improve their activity [1,2].

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 271–278, 2011.

With these constraints in mind, we propose a framework to aid the definition of personal models that allow the user to access humour with more confidence on the precision of his preferences. When searching for amusing content with little or no time, the user is more interested in spending a nice time than grasping everything that would be possible. In other words, in a computer science point of view, the precision of the displayed content is more relevant than its recall performance.

Active learning designs and analyses learning algorithms that can effectively filter or choose the samples to be labeled by a supervisor (a.k.a. oracle or teacher). The reason for using active learning is mainly to expedite the learning process and reduce the labeling efforts required by the teacher [3]. Another strong reason is the possibility for each user to define personal labels, thus constructing a customized learning model that better fits his preferences.

The SVM active learning framework we propose is a certainty-based method using the definition of the specific most informative examples to improve baseline performance, with two major guidelines: (i) the number of active examples has to be necessarily small; and (ii) precision is a critical factor.

The rest of the paper is organized as follows. We start in Section 2 by describing the background on SVM, active learning and humour classification and proceed into Section 3 by presenting the active learning framework for humour classification. Then, in Section 4 we introduce the Jester benchmark and discuss the results obtained. Finally, in Section 5 we delineate some conclusions and present some directions for future work.

2 Background

In what follows we will provide the background on Support Vector Machine (SVM), active learning and humour classification, which constitute the generic knowledge for understanding the approach proposed ahead in this paper.

2.1 Support Vector Machines

SVM is a machine learning method introduced by Vapnik [4], based on his Statistical learning Theory and Structural Risk Minimization Principle. The underlying idea behind the use of SVM for classification, consists on finding the optimal separating hyperplane between the positive and negative examples. The optimal hyperplane is defined as the one giving the maximum margin between the training examples that are closest to it. Support vectors are the examples that lie closest to the separating hyperplane. Once this hyperplane is found, new examples can be classified simply by determining on which side of the hyperplane they are.

The output of a linear SVM is $u = \mathbf{w} \times \mathbf{x} - b$, where $\mathbf{w}$ is the normal weight vector to the hyperplane and $\mathbf{x}$ is the input vector. Maximizing the margin can be seen as an optimization problem:

$$\begin{aligned} &minimize \quad \frac{1}{2}||\mathbf{w}||^2, \\ &subjected\ to\ \ y_i(\mathbf{w}.\mathbf{x}+b) \geq 1, \forall i, \end{aligned} \tag{1}$$

where $\mathbf{x}$ is the training example and y_i is the correct output for the ith training example. Intuitively the classifier with the largest margin will give low expected risk, and hence better generalization.

To deal with the constrained optimization problem in (1) Lagrange multipliers $\alpha_i \geq 0$ and the Lagrangian (2) can be introduced:

$$L_p \equiv \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{l} \alpha_i(y_i(\mathbf{w}.\mathbf{x}+b)-1). \tag{2}$$

In fact, SVM constitute currently the best of breed kernel-based technique, exhibiting state-of-the-art performance in diverse application areas, such as text classification [5,6,7]. In humour classification we can also find the use of SVM to classify data sets [8,1].

2.2 Active Learning

The key idea behind active learning is that a machine learning algorithm can achieve greater accuracy with fewer training labels if it is allowed to choose the data from which it learns. An active learner may pose queries, usually in the form of unlabeled data instances to be labeled by an oracle [9].

Active learning methods can be grouped according to the selection strategy: committee-based and certainty-based [10]. The first group determines the active examples combining the outputs of a set of committee members. As in [11], most effort is done in determining the examples in which the members disagree the most as examples to be labeled. The certainty-based methods try to determine the most uncertain examples and point them as active examples to be labeled. The certainty measure depends on the learning method used.

2.3 Humour Classification

Humour research in computer science has two main research areas: humour generation [2, 12] and humour recognition [8, 1, 13]. With respect to the latter, research done so far considers mostly humour in short sentences, like *one-liners*, that is jokes with only one line sentence, and the improvement of interaction between applications and users.

Humour classification is intrinsically subjective. Each one of us has its own perception of fun, hence automatic humour recognition is a difficult learning task that is gaining interest among the scientific community.

Classification methods used thus far are mainly text-based and include SVM classifiers, *naïve Bayes* and less commonly decision trees.

In [8] a humour recognition approach based in *one-liners* is presented. A dataset was built grabbing *one-liners* from many websites with an algorithm and the help of web search engines. This humorous dataset was then compared with non-humorous datasets like headlines from news articles published in the Reuters newswire and a collection of proverbs.

Another interesting approach [13] proposes to distinguish between an implicit funny comment and a not funny one. A 600,000 web comments dataset was used, retrieved from the Slashdot news Web site. These web comments were tagged by users in four categories: funny, informative, insightful, and negative, which split the dataset in humorous and non-humorous comments.

3 Proposed Approach

This section describes the proposed SVM active learning strategy. The SVM active learning framework we propose is a certainty-based method, i.e. it determines the most uncertain examples and point them as active examples to be labeled. As the certainty measure depends on the learning method used, for SVM we used the margin as the determining factor. When an SVM model classifies new unlabeled examples, they are classified according to which side of the Optimal Separating Hyperplane (OSH) they fall. As can be gleaned from Fig. 1, not all unlabeled points are classified with the same distance to the OSH. In fact, the farther from the OSH they lie, i.e. the larger the margin, more confidence can be put on their classification, since slight deviations of the OSH would not change their given class.

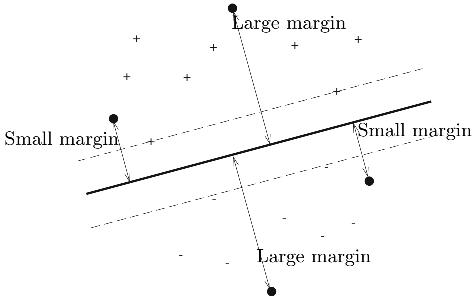


Fig. 1. Unlabeled examples (black dots) with small and large margins

Our active learning approach includes a certain number of unlabeled examples from the testing set (only the features, not the classification) in which the SVM has less confidence (smaller margin, see Fig. 1) after they are correctly classified by the supervisor. Thus, an example $(\mathbf{x}_i, y_i)$ will be included if Equation (3) holds.

$$(\mathbf{x}_i, y_i) : \rho(\mathbf{x}_i, y_i) = \frac{2}{\|w\|} < \varDelta \tag{3}$$

This number of examples can not be large, since the supervisor will be asked to manually classify them. After being correctly classified, they are integrated in the training set. This approach can be regarded as a form of active learning, where the information introduced by each example in the classification task is inversely proportional to its classification margin.

Despite not being fully automated, the active learning method has the potential to efficiently improve classification performance, since a user must classify the margin-based chosen examples. These examples help customize the learning machine regarding personal classification preferences.

4 Experimental Setup

4.1 Data Set

The Jester dataset contains 4.1 million continuous ratings (-10.00 to +10.00) of 100 jokes from 73,421 users and is available at: http://eigentaste.berkeley.edu. It was generated from Ken Goldberg's joke recommendation website, where users rate a core set of 10 jokes and receive recommendations from other jokes they could also like. As users can continue reading and rating and many of them end up rating all the 100 jokes, the dataset is quite dense. The dataset is provided in three parts: the first one contains data from 24,983 users who have rated 36 or more jokes, the second one data from 23,500 users who have rated 36 or more jokes and the third one contains data from 24,938 users who have rated between 15 and 35 jokes. The experiments were carried out using the first part as it contains a significant number of users and rates for testing purposes, and for classification purposes was considered that a joke classified on average above 0.00 is a recommendable joke, and a joke classified below that value is non recommendable. The jokes were split into two equal and disjoint sets: training and test. The data from the training set is used to select learning models, and the data from the testing set to evaluate performance.

4.2 Pre-processing Methods

A joke is represented as the most common, simple and successful document representation, which is the vector space model, also known as *Bag of Words*. Each joke is indexed with the *bag* of the terms occurring in it, i.e., a vector with one component for each term occurring in the whole collection, having a value that takes into account the number of times the term occurred in the joke. It was also considered the simplest approach in the definition of term, as it was defined as any space-separated word. Considering the proposed approach and the use of text-classification methods, pre-processing methods were applied in order to reduce feature space. These techniques, as the name reveals, reduce the size of the joke representation and prevent the mislead classification as some words, such as articles, prepositions and conjuctions, called *stopwords*, are non-informative words, and occur more frequently than informative ones. These words could also mislead correlations between jokes, so *stopword* removal technique was applied.

Stemming method was also applied. This method consists in removing case and inflection information of a word, reducing it to the word stem. Steaming does not alter significantly the information included, but it does avoid feature expansion.

4.3 Performance Metrics

In order to evaluate a binary decision task we first define a contingency matrix representing the possible outcomes of the classification, as shown in Table 1.

Table 1. Contingency table for binary classification

	Class Positive	Class Negative
Assigned Positive	a (True Positives)	b (False Positives)
Assigned Negative	c (False Negatives)	d (True Negatives)

Several measures have been defined based on this contingency table, such as, error rate ($\frac{b+c}{a+b+c+d}$), recall ($R = \frac{a}{a+c}$), and precision ($P = \frac{a}{a+b}$), as well as combined measures, such as, the van Rijsbergen F_β measure [14], which combines recall and precision in a single score:

$$F_\beta = \frac{(\beta^2+1)P \times R}{\beta^2 P + R}. \tag{4}$$

F_β is one of the best suited measures for text classification used with $\beta = 1$, i.e. F_1, an harmonic average between precision and recall (5).

$$F_1 = \frac{2 \times P \times R}{P + R}. \tag{5}$$

4.4 Results and Discussion

To test the proposed approach an experimental setup with three different experiments was defined:

1. Baseline SVM
2. Active Learning SVM with random active examples (Random AL SVM)
3. Active Learning SVM with margin-based active examples (Margin AL SVM)

Keeping in mind our initial guidelines: (i) the number of active examples has to be necessarily small; and (ii) precision is a critical factor, we defined a set of only 10 active examples, following the initial dataset construction procedure (see Section 4.1).

In the first experiment the SVMLight[1] package was used with linear kernels and default parameters. For the second experiment 30 runs were carried out, by randomly selecting 10 active examples and average values are presented. For the third experiment, the proposed SVM margin-based active learning strategy was deployed (see Section 3). Table 2 summarizes the performance results obtained.

Table 2. Performances of Baseline and Active Learning Approaches

	TP	FP	TN	FN	Precision	Recall	F1
Baseline SVM	35	8	4	3	81.40%	92.11%	86.42%
Random AL SVM	32	6	6	6	84.36%	84.74%	83.81%
Margin AL SVM	36	5	7	2	87.80%	94.74%	91.14%

Focusing on precision values, we can see that there is a trend for improvement: 81.40%, 84.36% and 87.80%. This can become a determining factor in humour classification, since users are typically more interested in a strong confidence of amusement (low false positive values) than in the guarantee of getting all jokes (low false negative values).

Comparing both active learning strategies, we can see that although both present improvements in precision, the random approach achieves it at the expense of recall values, while the proposed margin-based active learning permits the improvement of both recall and precision.

5 Conclusions and Future Work

In this paper we have described a framework for humour classification, based on an SVM active learning strategy. Our aim was to evaluate the use of such a strategy to increase the overall humour classification precision. For that purpose we have conducted a set of experiments with the Jester benchmark data set, by comparing the baseline SVM model with a two-fold active learning approach: (i) using a set of arbitrary examples; and (ii) using a set of the most relevant examples.

The preliminary results obtained are very promising. We were able to observe that the proposed active learning strategies have increased the overall precision measure, i.e. have reduced the false positive examples, when compared with the baseline SVM classification.

Regarding the recall, we have also observed that only in the active learning approach with the most relevant examples we were able to maintain an appropriate false negative rate, hence not worsening the overall classification results (e.g. $F1$). That is, for the specific case of joke classification with the Jester data set, using an active learning approach, we increase the amount of jokes correctly classified as having fun and thus *recommended* for reading. Such an active

[1] http://svmlight.joachims.org/

learning approach may also benefit other different application domains, like the recommendation systems for books or movies.

Our research is now focused on introducing crowdsourcing information into the active learning processing. That is, instead of using the results obtained from a supervisor we intend to use knowledge acquired from the end-users volunteer participation.

References

1. Mihalcea, R., Strapparava, C.: Making computers laugh: investigations in automatic humor recognition. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, HLT 2005, pp. 531–538. Association for Computational Linguistics, Stroudsburg (2005)
2. Stock, O., Strapparava, C.: Getting serious about the development of computational humor. In: IJCAI 2003, pp. 59–64 (2003)
3. Baram, Y., El-Yaniv, R., Luz, K.: Online choice of active learning algorithms. In: Proceedings of ICML-2003, 20th International Conference on Machine Learning, pp. 19–26 (2003)
4. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1999)
5. Joachims, T.: Learning Text Classifiers with Support Vector Machines. Kluwer Academic Publishers, Dordrecht (2002)
6. Tong, S., Koller, D.: Support vector machine active learning with applications to text classification. The Journal of Machine Learning Research 2, 45–66 (2002)
7. Antunes, M., Silva, C., Ribeiro, B., Correia, M.: A hybrid AIS-SVM ensemble approach for text classification. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part II. LNCS, vol. 6594, pp. 342–352. Springer, Heidelberg (2011)
8. Mihalcea, R., Strapparava, C.: Technologies That Make You Smile: Adding Humor to Text-Based Applications. IEEE Intelligent Systems 21(5), 33–39 (2006)
9. Settles, B.: Active learning literature survey. CS Technical Report 1648. University of Wisconsin-Madison (2010)
10. Silva, C., Ribeiro, B.: On text-based mining with active learning and background knowledge using svm. Soft Computing-A Fusion of Foundations, Methodologies and Applications 11(6), 519–530 (2007)
11. McCallum, A.K., Nigam, K.: Employing EM and pool-based active learning for text classification. In: Proceedings of ICML-1998, 15th International Conference on Machine Learning, pp. 350–358. Morgan Kaufmann Publishers, San Francisco (1998)
12. Binsted, K., Ritchie, G.: An implemented model of punning riddles. arXiv.org, vol. cmp-lg (June 1994)
13. Reyes, A., Potthast, M., Rosso, P., Stein, B.: Evaluating Humor Features on Web Comments. In: Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010) (May 2010)
14. van Rijsbergen, C.: Information Retrieval. Butterworths ed. (1979)

A Hybrid Approach to Feature Subset Selection for Brain-Computer Interface Design

John Q. Gan, Bashar Awwad Shiekh Hasan, and Chun Sing Louis Tsui

School of Computer Science and Electronic Engineering,
University of Essex, Colchester CO4 3SQ, UK
jqgan@essex.ac.uk

Abstract. In brain-computer interface (BCI) development, temporal/spectral/ spatial/statistical features can be extracted from multiple electroencephalography (EEG) signals and the number of features available could be up to thousands. Therefore, feature subset selection is an important and challenging problem in BCI design. Sequential forward floating search (SFFS) has been well recognized as one of the best feature selection methods. This paper proposes a filter-dominating hybrid SFFS method, aiming at high efficiency and insignificant accuracy sacrifice for high-dimensional feature subset selection. Experiments with this new hybrid approach have been conducted on BCI feature data, in which both linear and nonlinear classifiers as wrappers and Davies-Bouldin index and mutual information based index as filters are alternatively used to evaluate potential feature subsets. Experimental results have demonstrated the advantages and usefulness of the proposed method in high-dimensional feature subset selection for BCI design.

Keywords: Feature selection, high-dimensional data analysis, data classification, brain computer interface.

1 Introduction

Feature dimensionality reduction, by feature subset selection or feature space projection, is a very important preprocessing step in pattern recognition in high-dimensional feature spaces, with feature selection being advantageous in terms of interpretability [1][2][3]. This is an important and challenging problem in brain-computer interface development, because temporal/spectral/ spatial/statistical features can be extracted from multiple electroencephalography (EEG) signals and the number of features available could be up to thousands [2][4]. Feature selection aims to obtain a subset of features so as to achieve good classification performance and high computational efficiency. There are two main components in a feature selection algorithm: search methods to obtain potential subsets of features, and criteria to evaluate the performance of potential feature subsets.

An intuitive search method is to examine the classification performance of all the possible combinations of features, *i.e.*, $\sum_{m=1}^{N} N!/m!/(N-m)!$ possible feature subsets, where N is the number of available features. This type of search is complete,

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 279–286, 2011.

but inefficient or even impractical when *N* is too large. Alternatively, heuristic search methods can be used, such as sequential search and genetic algorithm. They are more efficient than the exhaustive search, but they may not search the whole feature space.

There are two approaches to performance evaluation of potential feature subsets: filter approach and wrapper approach. Filter approach evaluates potential feature subsets by measuring how relevant they are to class labels or how much separability they provide, independent of the classifier to be used. Wrapper approach, on the other hand, compares cross-validation classification accuracies obtained using a specific classifier with the potential feature subsets as inputs. Wrapper approach can be unrealistic when the number of features available for selection and the number of sample points are too large, especially when the classifier training is computationally expensive. Although filter approach can be much faster than wrapper approach, it often happens that the selection criteria commonly used by filter approach are not in line with classification accuracy. For instance, Davies-Bouldin index (DBI) [5] and mutual information (MI) based indexes [6] are biased to lower dimensions or not comparable when feature subsets are of different cardinalities, which is known as the feature cardinality bias problem [7]. Hence, hybrid approaches combining the advantages of filter approach and wrapper approach have been explored in recent years with different motivations and different search methods [8][9][10]. For instance, the method in [10] aimed at improving classification accuracy by running filters and wrappers in parallel and fusing their results, but not addressing the efficiency issue in high-dimensional feature subset selection.

Sequential forward floating search (SFFS) [11] is a commonly used feature selection method [12]. To our best knowledge, the flexible-hybrid SFFS in [8] was the only published hybrid approach using SFFS, thus SFFS had limitations in high-dimensional (*e.g.*, hundreds) feature subset selection. In the hybrid SFFS in [8], wrapper approach is much more dominant than filter approach. Although efficiency has been improved, it is still too computationally expensive for high-dimensional feature selection.

As an alternative to the wrapper-dominating hybrid SFFS (WDHSFFS) [8], this paper proposes a filter-dominating hybrid SFFS (FDHSFFS) method, aiming at high efficiency and insignificant accuracy sacrifice for high-dimensional feature selection applications, which is presented in the next section. FDHSFFS is evaluated in comparison with WDHSFFS and pure wrapper approach in Section 3, with experimental results of feature subset selection on high-dimensional BCI feature data. Section 4 ends the paper with discussions and conclusions.

2 Filter-Dominating Hybrid Sequential Forward Floating Search for Feature Subset Selection

There are two phases in an iteration of SFFS: growing phase and pruning phase. Starting from an empty feature subset, in the growing phase SFFS selects one feature from the remaining available features and adds it to the feature subset selected in the previous iteration so as to maximally improve the performance. If the currently selected feature subset contains more than 2 features, a pruning phase is carried out, in which a feature is dropped out of the currently selected feature subset if this improves the performance. The feature selection process stops when a preset number of features have been selected or when the performance is not improved any more after a preset number of iterations. Like other feature selection methods, filter approach or wrapper approach can be used in SFFS for evaluating the performance of potential feature subsets. As mentioned in the previous section, each approach has its pros and cons.

Hybrid SFFS combines filter and wrapper approaches in evaluating potential feature subsets. WDHSFFS combines filter and wrapper in the manner illustrated in Fig. 1, where λ_1 and λ_2 are hyper-parameters for controlling the proportion of features that are pre-selected by a filter and to be passed to a wrapper in growing phase and pruning phase respectively. In practice, it is difficult to choose appropriate values for λ_1 and λ_2. Too small values may lead to poor performance, whilst too large values may result in unnecessary high computational load.

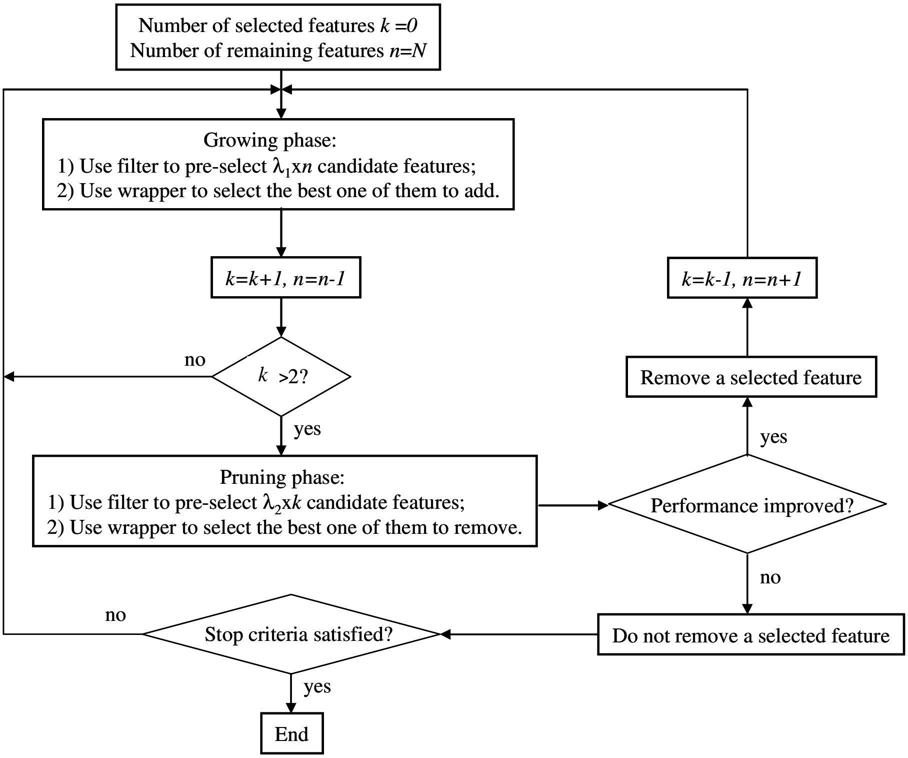

Number of selected features $k=0$
Number of remaining features $n=N$
Growing phase:
1) Use filter to pre-select λ_1xn candidate features;
2) Use wrapper to select the best one of them to add.
$k=k+1$, $n=n-1$
no
k >2?
yes
Pruning phase:
1) Use filter to pre-select λ_2xk candidate features;
2) Use wrapper to select the best one of them to remove.
Performance improved?
yes
Remove a selected feature
$k=k-1$, $n=n+1$
no
Do not remove a selected feature
Stop criteria satisfied?
no
yes
End

Fig. 1. The WDHSFFS algorithm

Aiming to avoid the difficulty of WDHSFFS in selecting values for the hyper-parameters (*i.e.,* λ_1 and λ_2) and to further improve the efficiency for high-dimensional feature selection applications, this paper proposes a filter-dominating hybrid SFFS (FDHSFFS) method, as illustrated in Fig. 2. It can be seen that the main modification is in the strategies for the growing and pruning phases, where the selection of a new feature to add or an existing one to remove is conducted purely by a filter, and a wrapper is used to compare the selected best feature subsets, which are of different cardinalities, before and after adding or removing a selected feature. On the contrary, in WDHSFFS a wrapper is used to compare several feature subsets preselected by a filter.

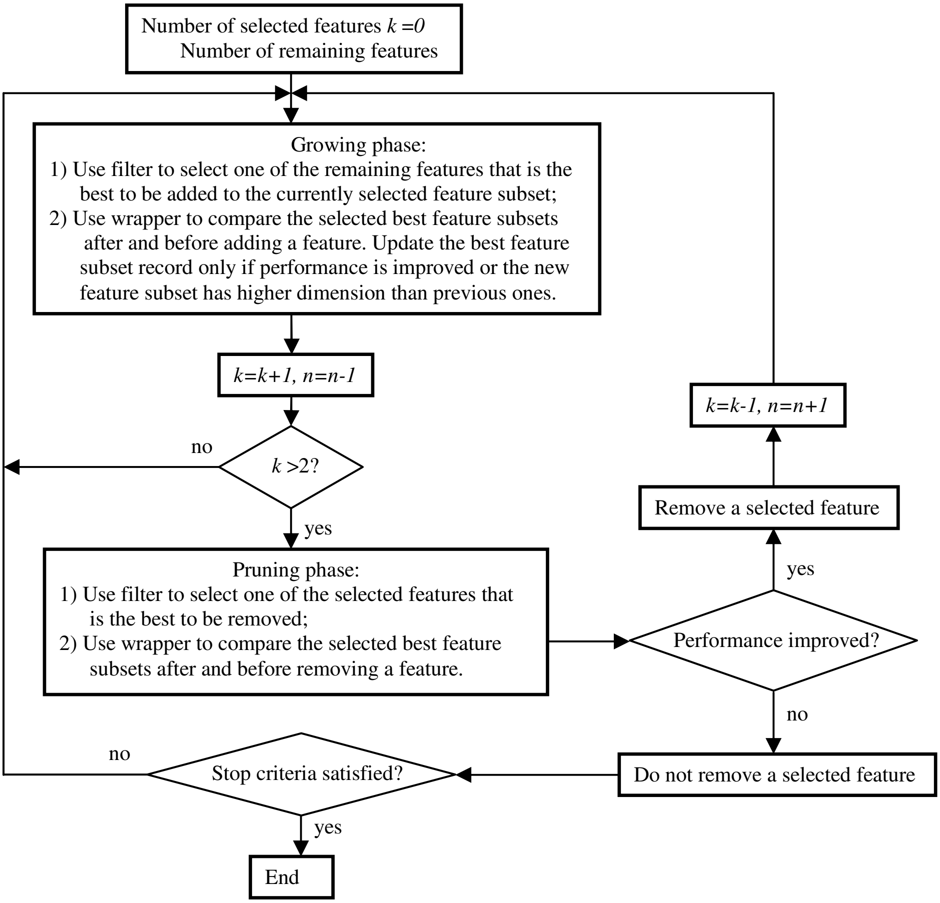

Number of selected features k =0
Number of remaining features
Growing phase:
1) Use filter to select one of the remaining features that is the best to be added to the currently selected feature subset;
2) Use wrapper to compare the selected best feature subsets after and before adding a feature. Update the best feature subset record only if performance is improved or the new feature subset has higher dimension than previous ones.
k=k+1, n=n-1
no
k >2?
yes
Pruning phase:
1) Use filter to select one of the selected features that is the best to be removed;
2) Use wrapper to compare the selected best feature subsets after and before removing a feature.
Performance improved?
yes
Remove a selected feature
k=k-1, n=n+1
no
Do not remove a selected feature
no
Stop criteria satisfied?
yes
End

Fig. 2. The FDHSFFS algorithm

Although pure filter approach is problematic in evaluating the separability of feature subsets of different cardinalities, it is found from our observations that filter approach can evaluate feature subsets of the same cardinality very well. Therefore, wrappers can be replaced by filters in evaluating the separability of feature subsets of the same cardinality for higher efficiency without significant loss of effectiveness. In the proposed FDHSFFS, a filter is only used to compare feature subsets of the same cardinality in selecting a new feature or removing an existing selected feature, and the comparison of feature subsets of different cardinalities is done by a wrapper. Therefore, it is expected that FDHSFFS is much more efficient than WDHSFFS without significant accuracy sacrifice, thus provides a promising alternative to the existing SFFS based feature subset selection methods.

3 Experimental Results

In this section FDHSFFS is tested on high-dimensional BCI feature data sets, in comparison with WDHSFFS and pure wrapper approach. The evaluation criterion used in the filter approach here is either DBI [5] or MI-based maximum relevance and minimum redundancy (MRMR) [6], and the wrapper used is either linear discriminate analysis (LDA) classifier or nonlinear support vector machine (SVM) classifier. In order to compare the efficiency of different methods in terms of the time spent by feature subset selection, the methods were tested on the same machine under similar conditions.

Table 1. Performance comparison

Methods	Average Number of Selected Features	Average Separability (Training Accuracy) (%)	Average Time Spent (Seconds)	Average 4-Fold Cross-Validation Accuracy (%)
FDHSFFS-DBI-LDA	8.67±2.31	74.84±1.41	908.33±141.65	72.49±2.61
WDHSFFS-DBI-LDA	9±1.73	75.45±1.7	11490±4.35e+3	73.05±1.96
FDHSFFS-DBI-SVM	8.33±2.89	79.22±7.67	3.57e+4±2.78e+3	76.49±9.30
WDHSFFS-DBI-SVM	8±1.73	79.74±7.32	4.07e+5±1.97e+5	77.47±8.33
FDHSFFS-MRMR-LDA	6.33±1.15	76.69±1.49	1503.33±321.46	75.95±2.61
WDHSFFS-MRMR-LDA	7.67±1.53	75.17±3.88	3.95e+4±3.35e+3	74.47±1.91
FDHSFFS-MRMR-SVM	10±0	95.95±1.42	5.61e+4±2.72e+4	77.98±7.80
WDHSFFS-MRMR-SVM	9.67±0.58	97.05±0.4	8.11e+5±5.02e+5	76.25±7.29
SFFS-LDA	10±0	79.58±2.04	2.44e+5±2.39e+4	77.55±2.98

The BCI data contains approximate entropy (ApEn) features extracted from EEG signals recorded from three subjects, each containing 76800 samples from 2 classes, with 420 features in each sample [13]. Such a large amount of high-dimensional features forms a real challenge to feature subset selection methods. Based on the suggestions from [8], the values of λ_1 and λ_2 were chosen so as to make the number of selected features by the filter sensible (sufficient but not redundant) and the classification performance of the selected feature subset satisfactory (close to the potential best). In this experiment λ_1 was set to 0.02 and λ_2 was set to 0.5. The maximum number of selected features was set to 10. The feature subset selection results on the ApEn data from the 3 subjects were obtained separately. However, due to the space limit, only the results averaged over the 3 subjects are presented in Table 1. The first column lists various combinations of methods, where SFFS-LDA represents pure wrapper approach with LDA classifier. Due to high-dimensionality and large number of samples, results from pure wrapper with SVM classifier were not obtained (It was actually impractical with the computer facility in our laboratory). The second column shows the average number of selected features, the third column shows the average training accuracy and standard deviation, the forth column presents the average time spent, and the last one contains the average 4-fold cross-validation accuracy and standard deviation. It is noted that the feature subsets selected by different methods are quite different. This is because the ApEn features are known to be redundant. However, to some extent all the methods were able to select reasonable feature subsets in terms of achieving good classification performance.

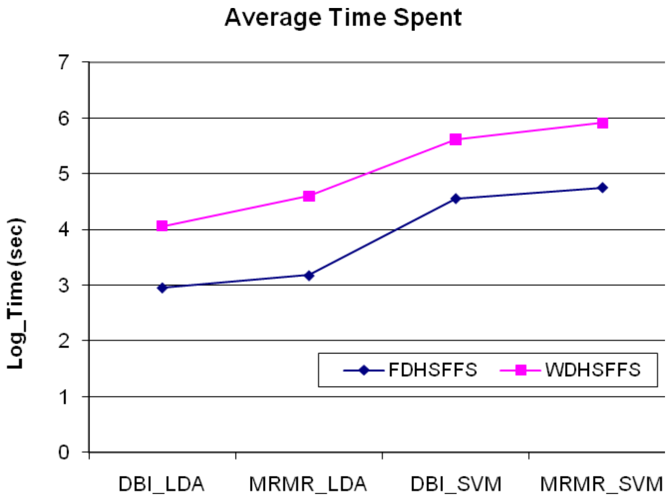


Fig. 3. Comparison of time spent by FDHSFFS and WDHSFFS

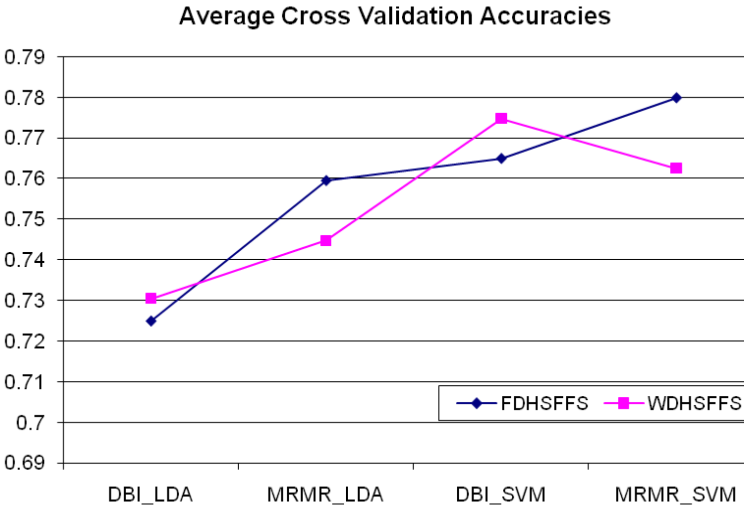


Fig. 4. Comparison of cross-validation classification accuracies of the feature subsets selected by FDHSFFS and those by WDHSFFS

A more intuitive performance comparison is illustrated in Figures 3 ~ 4. Fig. 3 illustrates the log-arithmetic average time spent by FDHSFFS against the time spent by WDHSFFS with different filter-wrapper pairs (DBI-LDA, MRMR-LDA, DBI-SVM, and MRMR-SVM), averaged over the data subsets from the 3 subjects. In a similar way, Fig. 4 gives the average cross-validation classification accuracies of the feature subsets selected by FDHSFFS against those selected by WDHSFFS. It is clear that with the same filter and wrapper, FDHSFFS was much faster than WDHSFFS, as shown in Fig. 3, but the feature subsets selected by them achieved similar classification performance, as shown in Fig. 4.

It is interesting to note that besides its obvious advantage in saving time, FDHSFFS also outperformed WDHSFFS in terms of cross-validation accuracy when using MRMR as filter. A sensible explanation for this is that using a wrapper to select features often results in over-fitting. From this perspective, FDHSFFS has an extra advantage over WDHSFFS due to its better capability to avoid over-fitting by using a filter dominant approach.

The methods were tested on more BCI data and similar results were obtained. However, those results are not included in this paper due to the paper length limit.

4 Discussions and Conclusions

This paper has proposed FDHSFFS as an alternative to the seminal SFFS and WDHSFFS for high-dimensional feature subset selection. Different from WDHSFFS, FDHSFFS uses filter as a dominant evaluation method since the filter approach is capable of evaluating potential feature subsets of the same cardinality, and it only uses wrapper to compare the selected best feature subsets of different cardinalities.

Comparing WDHSFFS and FDHSFFS algorithms as illustrated in Fig. 1 and Fig. 2 respectively, it is clear that a wrapper is used in WDHSFFS to evaluate $\lambda_1 n$ candidate feature subsets of the same cardinality whilst in FDHSFFS the wrapper is only used to evaluate one pair of feature subsets of different cardinalities. Because a wrapper is much more computationally expensive than a filter, replacing wrappers by filters in evaluating candidate feature subsets of the same cardinality in FDHSFFS makes it much faster than WDHSFFS. Furthermore, since filter approach is capable of evaluating potential feature subsets of the same cardinality, the feature subset selected by FDHSFFS can achieve classification performance comparable to the feature subset selected by WDHSFFS.

SFFS (pure wrapper approach) may be impractical for feature subset selection from a large number of samples of high-dimensional features. WDHSFFS can be much faster than SFFS, and FDHSFFS could be over 10 times faster than WDHSFFS with close classification performance when the feature dimension is very large, as demonstrated by the experimental results in the previous section. It is worthwhile to note that FDHSFFS outperformed WDHSFFS in terms of cross-validation classification accuracy when MRMR is used as a filter. Based on the experience in feature selection for BCI applications we believe that FDHSFFS will find widespread applications in feature subset selection from large data sets with high-dimensional features.

The two filters used here (DBI and MRMR) selected different features when applied in the FDHSFFS framework. This can be explained by the fact that the hybrid approach is sensitive to the filter used. DBI gives close values to features containing redundant information, resulting in the selection of redundant features. In contrast, MRMR selects features that are not redundant, which allows exploring more regions in the search space. When to use a filter to replace a wrapper and what type of filter to be used for specific applications should be investigated further.

One possible problem with the hybrid approach can be in handling the combination of several feature types, as the index value ranges differ from one feature type to another. This may make the feature selection biased towards features from a specific type. Normalization may provide an answer to this issue.

References

1. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Trans. on Knowledge and Data Engineering 17(4), 491–502 (2005)
2. Gan, J.Q.: Feature Dimensionality Reduction by Manifold Learning in Brain-Computer Interface Design. In: 3rd International Workshop on Brain-Computer Interfaces, Graz, Austria, pp. 28–29 (2006)
3. Gheyas, I.A., Smith, L.S.: Feature Subset Selection in Large Dimensionality Domains. Pattern Recognition 43(1), 5–13 (2010)
4. Awwad Shiekh Hasan, B., Gan, J.Q., Zhang, Q.: Multi-objective Evolutionary Methods for Channel Selection in Brain-Computer Interfaces: Some Preliminary Experimental Results. In: IEEE Congress on Evolutionary Computation, Barcelona, Spain, pp. 3339–3344 (2010)
5. Davies, J.L., Bouldin, D.W.: A Cluster Separation Measure. IEEE Trans. on Pattern Analysis and Machine Intelligence 1, 224–227 (1979)
6. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-dependency, Max-relevance, and Min-redundancy. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
7. Handl, J., Knowles, J.: Feature Subset Selection in Unsupervised Learning via Multiobjective Optimization. Int. Journal of Computational Intelligence Research 2(3), 217–238 (2006)
8. Somol, P., Novovičová, J., Pudil, P.: Flexible-Hybrid Sequential Floating Search in Statistical Feature Selection. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR 2006 and SPR 2006. LNCS, vol. 4109, pp. 632–639. Springer, Heidelberg (2006)
9. Huang, J., Cai, Y., Xu, X.: A Hybrid Genetic Algorithm for Feature Selection Wrapper Based on Mutual Information. Pattern Recognition Letters 28(13), 1825–1844 (2007)
10. Uncu, O., Turksen, I.B.: A Novel Feature Selection Approach: Combining Feature Wrappers and Filters. Information Sciences 177, 449–466 (2007)
11. Pudil, P., Novovicova, J., Kittler, J.: Floating Search Methods in Feature Selection. Pattern Recognition Letters 15, 1119–1125 (1994)
12. Jain, A., Zongker, D.: Feature Selection: Evaluation, Application, and Small Sample Performance. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(2), 153–158 (1997)
13. Dyson, M., Balli, T., Gan, J.Q., Sepulveda, F., Palaniappan, R.: Approximate Entropy for EEG-based Movement Detection. In: 4th International Workshop on Brain-Computer Interfaces, Graz, Austria, pp. 110–115 (2008)

Non-metric Multidimensional Scaling for Privacy-Preserving Data Clustering

Khaled Alotaibi, Victor J. Rayward-Smith, and Beatriz de la Iglesia

School of Computing Sciences,
University of East Anglia, Norwich, NR4 7TJ, UK
{K.Alotaibi,vjrs,B.Iglesia}@uea.ac.uk
http://www.springer.com/lncs

Abstract. Outsourcing data to external parties for analysis is risky as the privacy of confidential variables can be easily violated. To eliminate this threat, the data values of these variables should be perturbed before releasing the data. However, the perturbation itself may significantly change the underlying properties of the data, affecting the analysis results. What is required is a subtle transformation to generate perturbed data that maintains, as much as possible, the statistical properties and effectiveness (i.e. the *utility*) of the original data whilst preserving the privacy. We examine privacy-preserving transformations in the context of data clustering. In particular, this paper demonstrates how non-metric multidimensional scaling (MDS) can be profitably used as a perturbation tool and how the perturbed data can be effectively used in clustering analysis without compromising privacy or utility. We apply the proposed technique to real datasets and compare the results, which were, in some circumstances, exactly the same as those obtained from the original data.

Keywords: Privacy-preserving Data Mining, Non-metric Multidimensional Scaling, Random Perturbation, Clustering Analysis.

1 Introduction

Recent advances in the field of data mining have led to increased concerns about individual privacy. Clifton et al. [5] argue that data mining techniques are considered a challenge to privacy preservation, as often obtaining highly accurate results depends on the use of sensitive information about individuals.

Privacy concerns were firstly addressed in the statistics community with respect to protecting the individual's identity within a statistical database (i.e. microdata) using methods known as inference control in statistical databases or Statistical Disclosure Control (SDC) [6]. However, these methods were proposed to maintain only some properties in the data, such as marginal distributions, means and covariances whilst some other important properties such as distance between data points and correlations between variables were not adequately considered. In practice, most data mining algorithms also require the latter properties to ensure accurate results. In this context, many Privacy-Preserving Data

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 287–298, 2011.

Mining (PPDM) approaches have been proposed. They all share the same generic goal: to produce accurate mining results (i.e. to preserve the utility in the data) without disclosing "private" information.

We examine privacy-preservation in the context of cluster analysis. Partitional clustering aims to group a set of objects into homogenous non-overlapping subsets or clusters according to some notion of similarity. We use the k-means algorithm in this study as it is the most popular partitional algorithm.

The concept of "data perturbation" refers to transforming the data, and therefore hiding any private details whilst preserving the underlying probabilistic properties, so that the inherent patterns can still be accurately extracted. The probability of estimating the original data is one of several threats that can affect perturbation techniques.

In this paper, we propose using the non-metric MDS technique to preserve data utility for clustering whilst maintaining privacy. We use the perturbed data to carry out the clustering analysis using the k-means algorithm and show that the results are similar to those obtained from the original data.

The rest of this paper is organised as follows: Section 2 offers a general overview of related literature. Section 3 describes some basic concepts of MDS. Section 4 presents how the pertubed data is obtained with non-metric MDS. Section 5 discusses some potential attacks that could breach privacy and how our method should be resilient to them. Section 6 presents our experiments and discusses the results. Finally, Section 7 presents our conclusions and ideas for future work.

2 Related Work

The main body of literature on PPDM has appeared quite recently. The concept was first introduced in [2,14] and many techniques have emerged since then. Most works on PPDM are based on linear transformations using additive or multiplicative noise. Compared with other data perturbation methods, the randomization methods are relatively simple since the transformation process for the data values is performed in a data-independent way.

The additive perturbation technique originated within the SDC community [7] and was designed for microdata protection. In this method, a random number (noise) $r_{ij} \in R$ is added to the value of the attribute $x_{ij} \in X$ to produce a new value $y_{ij} \in Y$ where R is the random matrix, X is the original data matrix and Y is the perturbed data matrix. This random number is generally drawn from a normal distribution with mean $\mu = 0$ and standard deviation σ. The drawback of this method is that the added noise will distort the distances between data points and therefore poor results will be obtained when applying clustering algorithms on the perturbed data. Another drawback is that the additive noise can be filtered out and the privacy can then be compromised [1,10].

Multiplicative perturbation can provide, to some extent, a more powerful solution. However, the basic form [11], $y_{ij} = x_{ij}.r_{ij}$, does not preserve the distance between data points. Many researchers [3,4,16,18] have proposed variations to tackle this shortcoming using matrix multiplication. Suppose that n is the number of data records and m is the number of data attributes. The matrix multiplicative perturbation can be described simply by multiplying $X_{n\times m}$ by $R_{m\times m}$ in order to generate the new matrix $Y_{n\times m}$; this can then be released to the data miner. The random matrix is either a rotation matrix or a projection matrix. Although this kind of perturbation can provide a good data utility for data mining algorithms, the privacy model is not secure enough. The attacker can exploit some theoretical properties of the random matrices (they usually have a predictable structure) to disclose the original data values [8,9,15].

There are significant differences between the above techniques and the non-metric MDS introduced in this paper. Importantly, in non-metric MDS, the Euclidean distances between the points in the perturbed data are approximately preserved. This is achieved by placing the points in a lower dimensional space, within approximated distances, which reflect the distance between the original data points. Privacy attacks are more difficult since it is the rank-order of the dissimilarities between the original data points that is used instead of the real values, leading to greater uncertainty.

3 Basics of Multidimensional Scaling

Generally, MDS is a technique used to reduce the dimensionality of data. It attempts to find a set of data points (configuration) in some lower dimensional space where the distances between these points match to some degree the original dissimilarities of the n objects. The technique is also applicable when the experimental measurements are similarities, confusion probabilities, correlation coefficients or other diverse measures of proximity or dissociation between objects [12]. In this work we are concerned with application to clustering and we use the Euclidean metric, δ_{ij}, to describe the dissimilarities between two data points x_i and x_j:

$$\delta(x_i, x_j) = \delta_{ij} = \sqrt{\sum_{k=1}^{m}(x_{ik} - x_{jk})^2} \tag{1}$$

where m is the number of dimensions, and x_{ik} and x_{jk} are the k^{th} attributes of x_i and x_j, respectively.

MDS can be described mathematically as follows: given a set of objects $x_1, x_2, \cdots, x_n \in \mathbb{R}^m$ with dissimilarities $\delta_{ij}, 1 \leq i \leq j \leq n$, MDS aims to map these objects to a configuration or set of points $Y_1, Y_2, \cdots, Y_n \in \mathbb{R}^p, p < m$, where each point represents one of the objects and the distance between points d_{ij} are such that

$$d_{ij} = f(\delta_{ij}) \tag{2}$$

In the case of non-metric MDS, as we will see, f is a function that maintains a monotone relationship between the experimental dissimilarities and the distances in the configuration. Monotonicity is a very important property which will be discussed further as it is central to the non-metric MDS approach.

MDS can be viewed as a problem of statistical fitting - the dissimilarities are given and it is necessary to find the configuration whose distance fits them best. MSD must find a configuration $Y_1, Y_2, \cdots, Y_n \in \mathbb{R}^p$ where p is the dimensionality in which the distances are calculated. In practice, a perfect fit is not possible hence a"badness-of-fit" measure or loss function, e, over all configurations $Y_1, Y_2, \cdots, Y_n$ must be defined. Thus, with the lowest possible value of e, the best MDS is achieved.

The classical MDS usually choses an initial configuration in $\mathbb{R}^p$ (for a fixed p) and moves its points around, in iterative steps, to approximate the best relation $d_{ij} \approx f(\delta_{ij})$. In other words, the coordinates of each point in $\mathbb{R}^p$ are adjusted in the direction that maximally reduces the error. There are a variety of ways to formulate the approximation but all share only one objective, which is how well the interpoint distances, d_{ij}, approximate the original data dissimilarities, δ_{ij}.

In simple metric MDS, the relative error is a residual sum of squares, and can be defined by

$$e = \sum_{i,j}^{n} (f(\delta_{ij}) - d_{ij})^2. \tag{3}$$

Similarly, Sammon [19] suggests another metric approach to minimize the loss function. The data in m-dimensional space are projected into a lower p-dimensional space so the underlying structure is still preserved. The initial configuration of the data in the new space is usually sought by choosing Principle Components. A particular configuration of points with interpoint distances d_{ij}, representing the dissimilarities δ_{ij}, has the loss function

$$L = \frac{1}{\sum_{i<j}^{n} \delta_{ij}} \sum_{i<j}^{n} (d_{ij} - \delta_{ij})^2 / \delta_{ij}. \tag{4}$$

Both error measures defined above are suitable for *metric* MDS which uses the actual values of the dissimilarities, δ_{ij}. *Non-metric* MDS, in contrast, only uses the rank-order of the dissimilarities, so it is often called *ordinal* MDS. For two pairs of points (x_i, x_j) and (x_k, x_l), it is the rank-order $\delta_{ij} < \delta_{kl}$ that is sufficient to find an approximation of the distance values d_{ij}. Non-Metric MDS is important in the context of PPDM because it avoids the assumption made by other techniques that dissimilarities and distances are related by some fixed fomula. Furthermore, it does not use the variability of the data as a critical element in forming the distances in the configuration, and therefore it avoids some of the strong distributional assumptions that are necessary in variability-dependent techniques.

4 Non-metric MDS Data Perturbation

In non-metric MDS, only the rank-order of the dissimilarities, δ_{ij} (not the actual dissimilarities) is assumed to be important for generating the perturbed data.

Hence, the distances between the points in the perturbed data should, as far as possible, be in the same rank-order as the dissimilarities δ_{ij}. Given two pairs of points (x_i, x_j) and (x_k, x_l), whose rank-order is $\delta_{ij} < \delta_{kl}$, the corresponding distances in the perturbed data must ideally satisfy $d_{ij} \leq d_{kl}$. In practice, this is not achievable for all pairs of points and we seek a set of points Y that achieves this as best as possible.

In X, consider the set $\{\delta_{ij} \mid i < j\}$. This comprises $M = n(n-1)/2$ values that can be ordered

$$\delta_{i_1 j_1} < \delta_{i_2 j_2} < \ldots < \delta_{i_M j_M}.$$

Note, we are assuming here that there are no ties, i.e. that none of these M numbers are equal but, should this not be the case, the approach can be easily modified. In Y, we have corresponding distances

$$d_{i_1 j_1}, d_{i_2 j_2}, \ldots, di_M j_M$$

and, ideally, would like these to be in ascending order. Unfortunately, this is not usually the case. In [12,13], a monotone regression algorithm is described to compute from $d_{i_1 j_1}, d_{i_2 j_2}, \ldots, di_M j_M$ a nondecreasing sequence

$$\hat{d}_{i_1 j_1} \leq \hat{d}_{i_2 j_2} \leq \ldots \leq \hat{d}_{i_M j_M}$$

such that the *raw stress*

$$S^* = \sum_{i<j} (d_{ij} - \hat{d}_{ij})^2 \tag{5}$$

is minimised. This raw stress is normalised to produce a measure of the suitability of Y, called the (normalised) *stress*,

$$S = \sqrt{\sum_{i,j}^{n} (d_{ij} - \hat{d_{ij}})^2 / \sum_{i,j}^{n} d_{ij}^2}. \tag{6}$$

This stress is invariant under uniform stretching and shrinking of the configuration. As S is a residual sum of squares, it is positive, and the smaller the better. It can be expressed as a percentage, with 0% stress being equivalent to a perfect configuration, i.e. one that presents a perfect monotone relationship between dissimilarities and distances.

The non-metric MDS algorithm [12,13] then seeks $Y \subset \mathbb{R}^p$ such that the normalised stress is the minimum over all possible such Y. In other words $g(X)$ delivers the points $y_1, \cdots, y_n$ over p dimensions that minimises S. The minimisation can be accomplished by a numerical method described in [12,13]. The method proposed uses a steepest descent algorithm that starts with an arbitrary configuration and then improves it by moving it around to decreases the stress.

In the above, we have assumed the number of dimensions p is fixed and known. In practice this is not often the case and it is necessary to experiment to find an acceptable value of p by repeating the analysis using different values of p.

5 Privacy Breach Evaluation

Privacy breach can be described in terms of how well the original data values can be estimated from the perturbed data. Unlike other techniques, in which the transformation matrix is orthogonal (i.e. rotation) or projection, our method generates a new data configuration, whose interpoint distances approximate a nonlinear monotonic transformation of the original dissimilarities. Therefore, it seems, theoretically, to be more resilient to some potential attacks [4,8,15,20], particularly those that exploit the properties of the transformation matrix. All these attacks are based on how much information about the original data is available to the attacker (i.e. prior knowledge). This knowledge is obtained through one of two assumptions: *known input-output* and *known sample*. In the former, the attacker knows a collection of linearly dependent data points in the original data and the points to which they map in the perturbed data. The latter assumes that the attacker knows that the original data arose as independent samples of some m-dimensional random vector, $\mathbf{X}$, with an unknown pdf, and that (s)he has access to a collection of independent samples from $\mathbf{X}$.

Consider that the attacker knows some points in the original data, $X_{n_1 \times m}$, and their mapping in the perturbed data, $Y_{n_1 \times m}$, where $X_{n_1 \times m}$ and $Y_{n_1 \times m}$ are the known subsets from $X_{n \times m}$ and $Y_{n \times m}$, respectively. The attacker also knows that the data was transformed by an orthogonal transformation, $Y = XR$ where R is a random $m \times m$ rotation matrix. Liu et al. [15] proposed the *known input-output attack* to produce $\hat{R}$ as an estimation of R using both $X_{n_1 \times m}$ and $Y_{n_1 \times m}$. Then, the attacker can produce $\hat{x}_i$ as an estimation of x_i for all $i > n_1$ using the formula $\hat{x}_i = y_i \hat{R}^{-1}$.

In the known sample scenario, the attacker knows that some k samples arose independently from the same random variable $\mathbf{X}$, and that they are represented as columns in matrix S. Now, the attacker can use both matrix S and Y to estimate the random projection matrix R. This is known as the *PCA-based attack* [15]. The attack occurs when the eigenvectors of matrix Σ_S can be matched with their corresponding eigenvectors in matrix Σ_Y, where Σ_S and Σ_Y are the covariance matrix of S and Y respectively. Hence, the larger the differences between the eigenvalues of the covariance matrices, the more effective the attack. More precisely, the random matrix R can be estimated as $\hat{R} = Q_Y \Lambda Q_S^T$, where Q_Y and Q_S are the orthogonal matrices of the eigenvectors of Σ_Y and Σ_S, respectively, and Λ is a diagonal matrix $\Lambda \in I_n$, chosen so as to maximize the likelihood that $Q_Y \Lambda Q_S^T$ and Y have the same distribution.

Other attacks such as the *ICA-based attack* [4,8] have been defined but are based on assumptions (e.g. independent non-Gaussian distributed variables) that do not always hold in practice. Therefore, this kind of attack is not so effective in breaching privacy.

In non-metric MDS, the transformation is performed based on a monotone regression function, which monotonically relates the rank-order of the disparities $\hat{d}_{ij}$ resulting from $g : \delta_{ij} \rightarrow d_{ij}$ in the perturbed data to the given rank-order of the dissimilarities δ_{ij} obtained from the original data. The actual form of this function is unknown, and the only known is the final configuration (i.e. perturbed

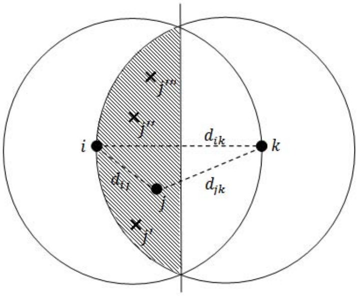


Fig. 1. Representation of all possible positions (j, j', j'', j''') to place the point j, without violating the constraint $d_{ij} \leq d_{jk} \leq d_{ik}$

data). As a result, the statistical information from the perturbed data Y are inconsistent with that from the original data X and, therefore, attacks such as those described above would be inefficient in breaching privacy.

In addition, the placement of points in the final configuration, is performed based on approximate distances that follow a predefined rank-order. Therefore, there is uncertainty about the exact distance between data points. Thus, this will make any distance-based attack, as described above, ineffective. Let i, j, k be three data points in the perturbed data Y; their interpoint distances are d_{ij}, d_{jk}, d_{ik} conforming to the rank-order $d_{ij} \leq d_{jk} \leq d_{ik}$. Thus, these points form a triangle, as illustrated in Fig 1. Assume that the points i and k have been placed and that the distance between them is d_{ik}. Without loss of generality, all possible positions for placing a point j, without violating the constraint $d_{ij} \leq d_{jk} \leq d_{ik}$, are bounded by the shaded area. The shaded area represents the uncertainty in the placement which helps to preserve privacy.

6 Experiments and Results

To evaluate the effectiveness of the non-metric MDS privacy model, we compared the quality of the generated clusters on both the original data X and the perturbed data Y. Ideally, the clustering results on Y should be the same, or very nearly the same, as those obtained from X. Four real numeric datasets were taken from the UCI machine learning repository (see Table 1).

The variation of information (VI) [17] was used to compare clusterings. A low value of VI infers that the two clusterings, $C = \{c_1, c_2, \cdots, c_k\}$, from X, and $C' = \{c'_1, c'_2, \cdots, c'_k\}$, from Y, are quite similar, while a high value infers the opposite. To compare the results and show how C and C' are related, we first construct a contingency table that tabulates the results of C against the results of C'. Then we calculate VI using

$$VI = E_C + E_{C'} - 2\,MI, \tag{7}$$

Table 1. Benchmark datasets used in our experiments

Dataset	# Records	# Attributes	# Classes
Iris	150	4	3
Wine	178	13	3
Breast Cancer Wisconsin (BSW)	699	9	2
Handwritten Digits (Optdigits)	3823	64	10

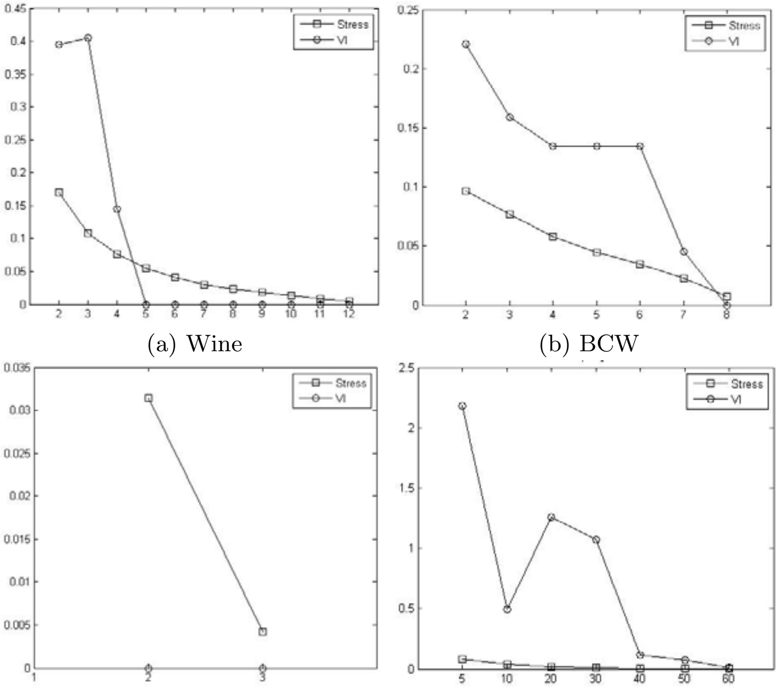
where E_C and $E_{C'}$ are the entropy of clustering C and C', respectively, and MI is the mutual information between C and C'. Equation 7 can be rewritten as follows:

$$VI = -\sum_{i=1}^{k} \frac{n_{i.}}{n} \log_2 \frac{n_{i.}}{n} - \sum_{j=1}^{k} \frac{n_{.j}}{n} \log_2 \frac{n_{.j}}{n} - 2\sum_{i=1}^{k}\sum_{j=1}^{k} \frac{n_{ij}}{n} \log_2 \frac{(n_{ij}/n)}{(\frac{n_{i.}}{n})(\frac{n_{.j}}{n})} \tag{8}$$

where n is the total number of records, k is the number of clusters, $n_{i.}/n$ and $n_{.j}/n$ are the marginal probabilities of clustering C_i and clustering C'_j, respectively, and n_{ij}/n is the joint probability that a record belongs to both C_i and C'_j. Note that the VI metric is bounded by $2\,log\,k$.

In our experiments, we used the implementation of k-means in Matlab to cluster the data. In each dataset, the number of clusters k was set as the number of classes. In k-means, the initial seeds (centroids) are chosen randomly, and thus, the final clustering can vary with each run. In other words, the k-means may assign the same data point x_i to a different clusters at every single execution. To guarantee stable clustering results, we determined the mean of the true classes as initial centroids for the k-means algorithm both for the original and perturbed data. This allows us to measure how the clusters obtained from both data (i.e. X and Y) compare without having to account for the randomness of the k-means algorithm.

The perturbation processes were carried out as follows: we normalized the original data, X, so all variables had zero mean and $\sigma = 1$. This helped in preventing one variable dominating the others in terms of Euclidean distance. Then, the dissimilarities $\mathcal{D}$ between the records in X were calculated using (1). To transform the dissimilarities $\mathcal{D}$ and generate the perturbed data Y, we used the Matlab non-metric MDS function (mdscale). The stress S (as defined in 6) was calculated to show how much information is lost as a result of the transformation. The results from experiments on the four datasets are depicted in Fig. 2, which indicate that the value of S increases whenever the number of dimensions p decreases. Meanwhile, the values of VI decrease as the number of dimensions increases. Indeed, the optimal performance of our perturbation method in terms of clustering validation (i.e. $VI = 0$) can be observed when the data are transformed into the lower dimensional space p by no more than 50% of their original dimensions, m. For instance, for the Wine dataset, we obtained a $VI = 0$, for all Y with $p = 5, \cdots, 12$ but the value of V increased for $p < 5$. For the Iris dataset, $VI = 0$ for all the lower dimensions. However, for the BCW dataset, the value of VI was zero only for $p = 8$ and, for other data with $p < 8$, the VI is slightly increased.



(a) Wine

(b) BCW

(c) Iris

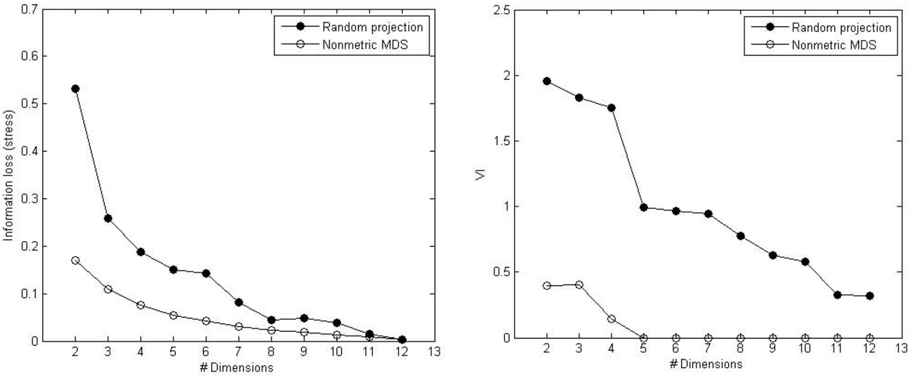
(d) Optdigits

Fig. 2. The VI and the relative error (stress) for the four datasets.

We plotted the clustering results obtained from original data X with those obtained by the perturbed data Y for some of the datasets and found that the clusterings were identical. The plots are not presented here due to space limitations.

In our experiments, we compared the performance of our method with the random projection method proposed by [18]. The results show that the linear transformation using random projection matrix causes the lossw of much information and it heavily distorts the distances between the data points. As a result, it leads to poor clustering results; the values of VI were high compared with our method. The result of this comparison is depicted in Fig. 3.

One limitation of non-metric MDS is its computational complexity, $\mathcal{O}(n^2)$, as fitting the monotone regression adds a considerable computational burden.



(a)

(b)

Fig. 3. Wine dataset, a comparison between Non-metric MDS and Random projection in terms of (a) information loss, and (b) clustering accuracy

7 Conclusions and Future Work

In this paper, we proposed using a non-metric MDS method to perturb data that is intended to be outsourced for data mining analysis, and to preserve as much as possible the underlying properties of the data. The results are promising and the prospects of more successful analyses are good. From the results of our experiments, it can be seen that non-metric MDS is a useful technique for privacy-preserving data clustering and may also be employed in other data mining tasks. It has also been shown that by projecting the data into a predetermined subspace, we can dramatically change their original form while preserving much of their underlying distance-related statistical characteristics. The most interesting feature of our method is that it is independent of the clustering algorithm (i.e. k-means), which is based heavily on the distance between the data points in order to partition the data.

In our experiments, we observe that when the data in a high dimension space m are transformed into a lower dimension space p, $p > m/2$, the generated clustering results obtained from the perturbed data are almost the same as those obtained from the original data, with $0 \leq VI < (2\, log\, k)/2$.

Since non-metric MDS requires the whole dataset to carry out the analysis of projecting the data into a lower dimension and processing new instances separately may be meaningless with respect to the transformed data in the new space, our model would not be suitable for incremental learning environment especially if we consider the computational efforts. However, it would be appropriate to investigate this in further work.

It would be interesting future work to address problems such as other potential attacks to our privacy model and the computational complexity of transforming large datasets. Moreover, we will investigate the ability of determining the number of dimensions for the scaling solution beforehand, so that we can build a trade-off between information loss and privacy. We will also apply our proposed method to other data mining tasks such as classification.

References

1. Agrawal, D., Aggarwal, C.C.: On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of the Twentieth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, p. 255. ACM, New York (2001)
2. Agrawal, R., Srikant, R.: Privacy-preserving data mining. ACM Sigmod Record 29(2), 439–450 (2000)
3. Chen, K., Liu, L.: Privacy preserving data classification with rotation perturbation. In: Proceedings of the Fifth IEEE International Conference on Data Mining, p. 4 (2005)
4. Chen, K., Sun, G., Liu, L.: Towards attack-resilient geometric data perturbation. In: Proceedings of the 2007 SIAM Data Mining Conference. SDM 2007 (2007)
5. Clifton, C., Kantarcioğlu, M., Vaidya, J.: Defining privacy for data mining. In: National Science Foundation Workshop on Next Generation Data Mining, pp. 126–133 (2002)
6. Domingo-Ferrer, J.: A survey of inference control methods for privacy-preserving data mining. In: Aggarwal, C., Yu, P. (eds.) Privacy-Preserving Data Mining: Models and Algorithms, ch. 3, pp. 53–80. Springer, Heidelberg (2008)
7. Domingo-Ferrer, J., Torra, V.: Disclosure control methods and information loss for microdata. In: Confidentiality, Disclosure and Data Access: Theory and Practical Applications for Statistical Agencies, pp. 93–112 (2002)
8. Guo, S., Wu, X.: Deriving private information from arbitrarily projected data. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 84–95. Springer, Heidelberg (2007)
9. Kargupta, H., Datta, S., Wang, Q., Sivakumar, K.: On the Privacy Preserving Properties of Random Data Perturbation Techniques. In: Proceedings of the Third IEEE International Conference on Data Mining, pp. 99–106. IEEE Computer Society, Los Alamitos (2003)
10. Kargupta, H., Datta, S., Wang, Q., Sivakumar, K.: Random-data perturbation techniques and privacy-preserving data mining. Knowledge and Information Systems 7(4), 387–414 (2005)
11. Kim, J.J., Winkler, W.E.: Multiplicative Noise for Masking Continuous Data. Technical report, Research Report Series - statistics 2003-01, Statistical Research Division. US Bureau of the Census, Washington, DC (2003)
12. Kruskal, J.B.: Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29(1), 1–27 (1964)
13. Kruskal, J.B.: Nonmetric multidimensional scaling: a numerical method. Psychometrika 29(2), 115–129 (1964)
14. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 36–54. Springer, Heidelberg (2000)
15. Liu, K., Giannella, C., Kargupta, H.: An attacker's view of distance preserving maps for privacy preserving data mining. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 297–308. Springer, Heidelberg (2006)

16. Liu, K., Kargupta, H., Ryan, J.: Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. IEEE Transactions on Knowledge and Data Engineering 18(1), 92–106 (2006)
17. Meila, M.: Comparing clusterings–an information based distance. Journal of Multivariate Analysis 98(5), 873–895 (2007)
18. Oliveira, S., Zaïane, O.R.: Privacy-preserving clustering to uphold business collaboration: A dimensionality reduction based transformation approach. International Journal of Information Security and Privacy 1(2), 13 (2007)
19. Sammon Jr., J.W.: A nonlinear mapping for data structure analysis. IEEE Transactions on Computers 100(5), 401–409 (1969)
20. Turgay, E., Pedersen, T., Saygın, Y., Savaş, E., Levi, A.: Disclosure risks of distance preserving data transformations. In: Ludäscher, B., Mamoulis, N. (eds.) SSDBM 2008. LNCS, vol. 5069, pp. 79–94. Springer, Heidelberg (2008)

Iranian Cancer Patient Detection Using a New Method for Learning at Imbalanced Datasets

Hamid Parvin, Behrouz Minaei-Bidgoli, and Hosein Alizadeh

Islamic Azad University Nourabad Mamasani Branch, Nourabad Mamasani, Iran
{parvin,b_minaei,halizadeh}@iust.ac.ir

Abstract. Most of standard learning algorithms presume or at least expect that distributions governed on the different classes of dataset are balanced. Also they presume that the misclassification cost of each data point is equal without considering its class. These algorithms fail to learn at the imbalanced datasets. Cancer detection is a well-known domain in which it is very common to face imbalanced class distributions. This paper presents an algorithm which is suit to this field, in both speed and efficacy. The experimental results show that the performance of the proposed algorithm outperforms some of the best methods in the field.

Keywords: Imbalanced Learning, Decision Tree, Artificial Neural Networks, Cancer Detection.

1 Introduction

Standard learning algorithms often assume or at least expect the distributions of different classes to be balanced and the misclassification cost of each data point to be equal without considering its class. When datasets are imbalanced these algorithms faces some problems to predict based on the correct distributions governed on the classes of dataset; in other words these algorithms are inclined to assign each data point to the most frequent class or the dominant class, so while they reach an acceptable precisions they have not an acceptable performances.

In fact, each dataset that represents an unbalanced distribution among its classes can be considered as an unbalanced dataset. However, datasets are generally considered to be unbalanced only if they have a very high rate of unbalanced distribution. We call this type of imbalances, the imbalance between classes (e.g. the distribution of 1000:1 of two classes, one class completely overshadows the other). Of course the imbalance concept is not special to the distributions between two classes. It is likely that one faces an imbalance dataset with number of classes more than two.

To clarify the importance of learning in the real world unbalanced dataset, we express an example. Let us consider a dataset of different clients. The clients are shown with either positive class or negative class based on being healthy or cancer patients. As it is expected, the number of healthy clients is much more than the normal number of cancer patients. The considered dataset includes 369 negative

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 299–306, 2011.

samples (majority class) and 17 positive samples (minority class). Here we need to have a classifier that well performs for both minority and majority classes. An unbalanced standard classifier generally gives accurate results so that the healthy class can be learned about 100 percent, while patient class can be learned about 0-10 percent. Suppose that the classifier accuracy is 10 percent for the patient class in this example. Thus 15-16 as patient clients are considered as healthy clients. The result is that 15-16 people who have cancer are diagnosed healthy. In medical recognition, an incorrect diagnosis of a cancer patient as a healthy one is more unacceptable than an incorrect diagnosis of a healthy one as a cancer patient. Thus in these cases it is required to use a classifier with high accuracy in such a way that the minority class is not affected by the majority class. It is obvious that the individual evaluation criteria such as overall accuracy or error rate do not provide sufficient information about the imbalanced learning. Unbalanced shape of a dataset is called intrinsic where the nature of data space results in its being unbalanced. It should be noted that the imbalance can be relative where the mean number of samples is high in the minority class, but it is very less than the number of samples of the majority class.

An artificial neural network is a model which is to be configured to be able to produce the desired set of outputs, given an arbitrary set of inputs. One of the most representatives of artificial neural networks is multilayer perceptron [7]. In this paper the multilayer perceptron is used as one of the base classifiers. Decision tree is considered as one of the most versatile classifiers in the machine learning field. Decision tree is considered as one of unstable classifiers. It means that it can converge to different solutions in successive trainings on same dataset with same initializations. It uses a tree-like graph or model of decisions. The kind of its knowledge representation is appropriate for experts to understand what it does [8].

2 Related Work

As it was expressed when the standard learning algorithm in the field of unbalanced dataset are used the description rules that express the concept of minority class are often poorer than the description rules expressing the concept of majority class and as a consequence the minorities class is not therefore often well learned. Here to show the effect on the issue of imbalanced learning algorithm, consider the standard decision tree learning algorithm. Decision tree is produced based on a recursively top-down search method. A decision tree is used a feature selection method to select the best feature in each node of the tree as the separation criteria. Next nodes are created based on property values for the separator. Thus in each level training set is divided into smaller subsets that can totally provide some rules for the concept of the class. Finally, the obtained rules jointly make a description that delivers the lowest error rates in the different classes. The problem comes when the feature space of the dataset is spitted more and more. During the splitting the feature space the data points of the minority class is being less and less. So in the leaf nodes the minority class is not well defined. To sum up, the provided solutions to learn in the imbalanced problems are generally in two approaches. A class of solutions is to apply some changes in dataset to be balanced. Other class generally focuses on learning algorithms to adapt them to be suited to learn an imbalanced data [5].

In the first approach, there are two common ways: over-sampling and under-sampling. Random over-sampling method takes a set of samples from minority class and then it is added to dataset. In fact, the number of samples in the minority class is enlarged in such a way that the number data points in both classes, either minority or majority classes, get balanced. Alternatively there is another way to balance the dataset. Unlike over-sampling, under-sampling reduces a set of samples from majority class in such a way that the number data points in both classes, either minority or majority classes, get balanced. The over-fitting is the problem that is challenged by over-sampling. The concept losing is the main problem of the under-sampling. An alternative to overcome the challenges is to turn to informed under-sampling. Two of the most well-known methods based on informed under-sampling are *EasyEnsemble* [2] and *BalanceCascade* [3]. Another example of the informed under-sampling methods is based on k-nearest neighbor [4]. In *EasyEnsemble* method it is tried to first produce many classifiers based on different runnings of the under-sampling method. Then use them as a classifier ensemble. It is worthy to note that each mentioned classifier is an AdaBoost classifier. *EasyEnsemble* is an unsupervised strategy since it uses independent random sampling with replacement. *BalanceCascade* method is very similar to *EasyEnsemble* method. *BalanceCascade* explores in a supervised manner. In *BalanceCascade* method it is tried to iteratively produce a classifier so as to improve the false positive rate of previously produced classifiers.

3 Evaluation Criteria at Learning Imbalanced

According to research in the field of imbalanced learning, it is needed to discuss criteria to assess how effective a model has learned an imbalanced dataset. In this part of the evaluation criteria to learn imbalanced datasets are outlined. The traditional conventional measures are the accuracy and error rate. The criteria are for a simple description of the performance of a learner on a dataset but are not suitable for unbalanced datasets. The performance criteria defined on imbalanced datasets is based on the confusion matrix. For example, consider a dataset includes a minority class containing 5% of total dataset and a majority class containing the rest of 95%. A classifier that always vote for the majority class, no matter what the sample is, hit the accuracy of 95%, in spite of its lowest performance for recognition of the minority class. Studying the confusion matrix makes it clear that the first column shows the number of positive samples and second column shows the number of negative samples. It is also clear that the first row shows the number of the samples that classifier recognizes them as the minority class and the second row shows the number of the samples that classifier recognizes them as the majority class. Columns show the distribution of class samples. Indeed each metric using them simultaneously can not be free from sensitivity to class unbalanceness. For example accuracy uses both columns and is sensitive to imbalances, i.e. changing the class distributions it can be changed while the performance may not change. Some measures which are adjusted for learning at the imbalanced dataset are: accuracy, precision, recall, and F-measure and G-mean [1]. The accuracy is obtained by equation 1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

The precision is obtained by equation 2.

$$precision = \frac{TP}{TP + FP} \tag{2}$$

The recall is obtained by equation 3.

$$recall = \frac{TP}{TP + FN} \tag{3}$$

The F-measure is obtained by equation 4.

$$F - measure = 2 * \frac{precision * recall}{precision + recall} \tag{4}$$

Evaluation based on receiver operating characteristic (ROC) curves, uses two criteria, True Positive (TP) rate and False Positive (FP) rate, of the confusion matrix and draws a graph depicting the TP rate in terms of the FP rate. ROC curve is a powerful method to evaluate the performance of a learner visually. In precision-recall chart, one could get more information on the performance assessment of a learner [1]. These charts can be considered as the best way to represent performance in an unbalanced application.

4 Proposed Method

Structure of the proposed algorithm is similar to *EasyEnsemble* algorithm is. The proposed algorithm initially takes a number of sub-sampling from the majority class with the size of the minority class. Regarding each of these sub-sampled data in addition to the data of the minority class as a temporal dataset, we train a decision tree or a multilayer perceptron. Finally, all classifiers jointly work as an ensemble. Pseudo code of the proposed algorithm is presented in the Fig. 1. Although as was said previously there are many algorithm to deal with learning at imbalanced datasets, this paper only focuses to handle under-sampling approaches. In this group of algorithms, the two of the best algorithms are considered as *BalanceCascade* and *EasyEnsemble*. It is worthy to mention that the second is one of the samples of informed sampling methods [2] and [3]. As it has shown [3], these two algorithms absolutely dominate other methods. Their superiority is in terms of efficiency and training speed.

On the other hand the algorithms of *BalanceCascade* and *EasyEnsemble* very similar to the proposed algorithm. Therefore, since the two algorithms in terms of the structure are very similar to the proposed algorithm, and they dominate other methods this paper compares the proposed method to only these methods.

Question is what the difference between proposed algorithm and *EasyEnsemble* algorithm is. Difference is in section 6 of the code. Instead of *EasyEnsemble* algorithm uses an Adaboost ensemble as classifier [5]. Using a complex classification system similar to Adaboost, not only comes with a lot of overhead time, but actually it is without justification, because after producing classifiers, C_i, voting mechanism is used. It is highly likely that the classifiers do not train properly in the AdaBoost algorithm due to the small number of samples minority class.

```
The ModifiedBagging algorithm pseudo code.
    Input: A set of minority class examples $S_{min}$, a set of majority class examples $S_{max}$, $/S_{min}/ < /S_{max}/$, the number of subsets, $T$ to sample from $S_{max}$
    $i \Rightarrow 0$
    repeat
    $i \Rightarrow i+1$
    Randomly sample a subset $E_i$ from $S_{max}$, $/E_i/ = / S_{min}/$, $S_i = S_{min} \cup E_i$
    Learn $C_i$ on $S_i$. $C_i$ is a simple classifier
    until $i = T$
    Output: An ensemble $\{H_i|1 \leq i \leq T\}$
```

Fig. 1. Pseudo code of the proposed algorithm

The difference between proposed algorithm and *BalanceCascade* algorithm is more obvious. Differences in the previous section are all valid here. Besides *BalanceCascade* algorithm tries to iteratively produce an AdaBoost so as to improve the false positive rate of previously produced classifiers. It is again highly likely that the classifiers do not train properly in the AdaBoost algorithm due to the small number of samples minority class.

5 Experimental Results

This section evaluates the results of applying the proposed framework on a real dataset of breast cancer. This paper explores a model to help medical jobs be done by a machine learning system for cancer breast detection. Dataset has been collected from some real clients of Bidgol-Aran city's hospital [6]. Dataset includes 369 clients. While 17 cases had breast cancer, the rest 352 cases had been healthy. Maximum of 26 features extracted that the most of them almost belong to the nominal features. The nominal features are first converted to numerical features. After the coding phase, each feature is normalized into interval [0-1]. The normalizing relations can be calculated by equation 5.

$$nf_{x,i} = \frac{f_{x,i}}{\max_{y}(f_{y,i}) - \min_{y}(f_{y,i})} \tag{5}$$

Where $f_{x,i}$ stands for ith feature of xth data point and $nf_{x,i}$ stands for ith normalized feature of xth data point. In this paper, multilayer perceptron and decision tree are used as base classifier. We use multilayer perceptrons with 2 hidden layers including respectively 10 and 5 neurons in the hidden layer 1 and 2, as the base simple classifier. All of decision trees used in this research employ Gini criterion as decision tree evaluation metric. Parameter Gini criterion for decision tree was set by two.

Table 1. Performances of different methods obtained by leave-one-out method

ModifiedBagging of 1 MLPs	ModifiedBagging of 1 Decision Trees	MLP	Decision Trees	Evaluation Criterion
4/17=23.53	10/17=58.82	0/17=0	1/17=3.70	TP
116/352=32.95	82/352=23.30	0/352=0	0/352=0	FP
236/352=67.05	270/352=76.70	352/352=1	352/352=1	TN
13/17=66.47	7/17=41.18	17/17=1	16/17=94.12	FN
240/369=65.04	280/369=75.88	352/369=95.39	353/369=95.66	Accuracy
3.33	10.87	∝ (50)	100	Precision
23.53	58.82	0	3.70	Recall
5.84	18.35	∝ (50)	7.14	F-Measure

The classifiers' parameters are kept fixed during all of their experiments. It is important to take a note that all classifiers in the algorithm are kept fixed to only either decision tree or multilayer perceptron. It means that all classifiers are considered as multilayer perceptron in the first experiments. After that the same experiments are taken by substituting all multilayer perceptrons with decision trees.

As it is inferred from Table 1, although the accuracies of simple decision tree classifier and multilayer perceptron neural network are very high, they do not have good performance at all. This is not something unexpected, because these classifiers assign each queried sample to the majority class. Consequently they hit a very high accuracy. While their accuracies are good, they are unable to recognize patients. If one looks at Table 1, it will be clearly identified that the performances of the same classifiers enclosed in the proposed framework are significantly increased; while they have also satisfactory accuracies. As expected, using the decision tree as the base classifier can improve considerably the performance of using the multilayer perceptron neural network as the base classifier.

Table 2. Performances of different methods obtained by leave-one-out method

ModifiedBagging of 25 MLPs Mean Cut of ROC curve	ModifiedBagging of 25 Decision Trees and Mean Cut of ROC curve	Bagging of 25 MLPs with Best Cut of ROC curve	Bagging of 25 Decision Trees with Best Cut of ROC curve	Evaluation Criterion
11/17=64.71	13/17=76.47	0/17=0	1/17=3.70	TP
116/352=32.95	71/352=20.17	0/352=0	0/352=0	FP
236/352=67.05	281/352=79.83	352/352=1	352/352=1	TN
6/17=35.29	4/17=23.53	17/17=1	16/17=94.12	FN
247/369=66.94	294/369=79.67	352/369=95.39	353/369=95.66	Accuracy
8.66	15.48	∝ (50)	100	Precision
64.71	76.47	0	3.70	Recall
15.28	25.75	∝ (0)	7.14	F-Measure

Now another comparison between the performances of using these two classifiers as the base classifier is done. These experiments show that the accuracy of the proposed method is acceptable when we use the ensemble. This will also show if the whole datapoints of dataset be used to construct the classifiers of the final ensemble, performance of the final ensemble is still poor to identify examples of the minority class. Table 2 depicts this important fact. As it is raised from Table 2, use of the ensemble without applying the proposed method to balance the training data, does not solve the problem. However, applying the proposed method along to the use of ensemble significantly increased the efficiency. Consider the comparison between the performance of using the multilayer perceptron neural network as the base classifier and the performance of using the decision tree as the base classifier in the Fig. 2. ROC curve of the proposed methods using decision tree classifier as the base classifier is depicted in the left curve of Fig. 2. ROC curve of the proposed methods using multilayer perceptron classifier as the base classifier is depicted in the right curve of Fig. 2. Readers shall be found with regard to Fig. 2 that if a better cut choice is taken over ROC the results can even be improved. However, this is not stable because after a while it causes TP to be reduced. The above tests indicate that the accuracy of the proposed method outperforms the simple classifiers and some ensemble methods. The other conclusion is the superiority of using decision tree as the base classifier over using multilayer perceptron neural network as base classifier.

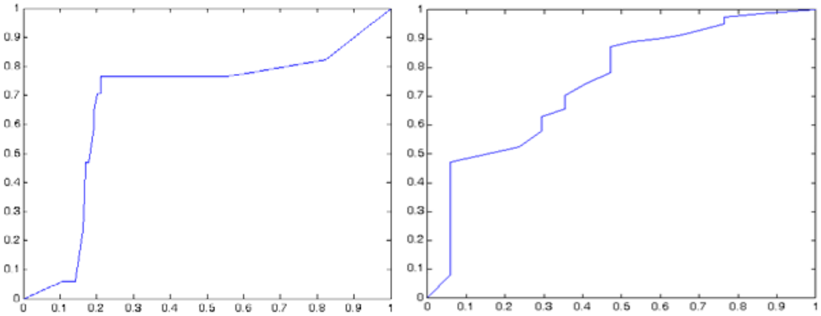


Fig. 2. ROC curve of the proposed methods with DT (left) and MLP (right) as base classifier

Now it is time to compare the proposed method with *EasyEnsemble* and *BalanceCascade* methods. By applying the algorithms, and simple linear classifiers used in reference [3], the methods were not again obtained acceptable results according to Table 3. Comparing the proposed algorithm with algorithms in the table below *EasyEnsemble*, we will reach the conclusion that the performances of the

Table 3. Comparison of proposed method with *EasyEnsemble* and *BalanceCascade* methods

Evaluation Criterion	*EasyEnsemble* of 25 classifiers Mean Cut of ROC curve	*BalanceCascade* 25 classifiers and Mean Cut of ROC curve	*ModifiedBagging*25 decision trees and Mean Cut of ROC curve
TP	3/17=17.65	5/17=17.65	13/17=76.47
FP	31/352=8.81	43/352=8.81	71/352=20.17
TN	321/352=91.19	309/352=91.19	281/352=79.83
FN	14/17=82.35	12/17=82.35	4/17=23.53
Accuracy	324/369=87.80	324/369=87.80	294/369=79.67
Precision	8.82	10.42	15.48
Recall	17.65	29.41	76.47
F-Measure	11.76	15.39	25.18

mentioned methods are weak. So that is needed not to go for reinforcement methods in such a dataset. Considering the higher time orders of the mentioned algorithms to learn in severely imbalanced datasets, we can claim that the proposed method in terms of both efficiency and speed of learning is superior. In addition, we have generally proposed a framework to achieve a similar learning model.

Perhaps the most important reason for failures of *EasyEnsemble* and *BalanceCascade* methods is in severely imbalanced nature of the dataset.

6 Conclusion

In this paper a new method to learn in imbalanced dataset in which data points of the minority class are much less than data points of the majority class was presented. This method was applied to a breast cancer dataset. Inability of basic methods to learn in imbalanced spaces was also shown. Also due to the rare number of data points of the minority class, the special-purpose methods are not suitable to learn the minority class in severely imbalanced datasets.

The main result of this research is in the field of medical research to be used as a medical assistant. According to the profile and history of clients in the health centers, the proposed model can identify high risk clients in an automated manner. It can detect and treat early cancer to cause significant savings to be in the country.

References

1. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowledge And Data Engineering 21(9), 1263–1284 (2009)
2. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory Under Sampling for Class Imbalance Learning. In: Proc. Int'l Conf. Data Mining, pp. 965–969 (2006)
3. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory Under sampling for Class-Imbalance Learning. IEEE Transactions on Systems, Man and Cybernetics—Part B: Cybernetics (2009)
4. Zhang, J., Mani, I.: KNN Approach to Unbalanced Data Distributions: A Case Study Involving Information Extraction. In: Proc. Int'l Conf. Machine Learning (ICML 2003), Workshop Learning from Imbalanced Data Sets (2003)
5. Hamzei, M., Kangavari, M.R.: Learning from imbalanced data. Technical Report. Iran University of Sci. & Tech., Iran (2010)
6. Minaei, F., Soleimanian, M., Kheirkhah, D.: Investigation the relationship between risk factors of occurrence of breast tumor in women, Aranobidgol, Iran (2009)
7. Haykin, S.: Neural Networks, a comprehensive foundation, 2nd edn. Prentice Hall International, Inc., Englewood Cliffs (1999) ISBN: 0-13-908385-5
8. Yang, T.: Computational Verb Decision Trees. International Journal of Computational Cognition, 34–46 (2006)

Automated Image Annotation System Based on an Open Source Object Database

Gabriel Mihai, Liana Stanescu, Dumitru Dan Burdescu, and Cosmin Stoica Spahiu

University of Craiova, Faculty of Automation, Computers and Electronics,
Bvd. Decebal, No.107, Craiova, Romania
{mihai_gabriel,stanescu,burdescu,stoica.cosmin}@software.ucv.ro

Abstract. Automated annotation of digital images is a challenging task being used for indexing, retrieving, and understanding of large collections of image data. Several machine-learning approached have been proposed to model the existing associations between words and images. Each approach is trying to assign to a test image some meaningful words taking into account a set of feature vectors extracted from that image. This paper presents an original image annotation system based on an open source object database called db4o. An object oriented model offers suport for storing complex objects as sets, lists, trees or other advanced data structures. The information needed for the annotation process is retrieved from the SAIAPR TC-12 Dataset – a set of annotated images having a vocabulary with a hierarchical structure. The annotation system is using an efficient annotation model called Cross Media Relevance Model.

Keywords: image annotation, image segmentation, ontology.

1 Introduction

Automatic image annotation is a task that assigns words to images taking into account their semantic content. There are two reasons that are making the image annotation a difficult task: *the semantic gap*, being hard to extract semantically meaningful entities using just low level image features and *the lack of correspondence* between the keywords and image regions in the training data.

There are many annotation models proposed and each model has tried to improve a previous one. These models were splitted in two categories:

a) Parametric models: Co-occurrence Model [1], Translation Model [2], Correlation Latent Dirichlet Allocation [3]

b) Non-parametric models: Cross Media Relevance Model (CMRM) [4], Continuous Cross-Media Relevance Model (CRM) [8], Multiple Bernoulli Relevance Model (MBRM) [9], Coherent Language Model (CLM) [10]

The annotation process implemented in our system is based on CMRM. Using a set of annotated images [12] the system extracts the information needed by the annotation process and then learns the joint distribution of blobs and words. Each new image is segmented using an efficient segmentation algorithm [11] that was used for an image annotation system presented in [15].

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 307–316, 2011.

The remainder of the paper is organized as follows: related work is discussed in Section 2, Section 3 contains a description of the annotation model and presents the structure of the SAIAPRTC-12 Dataset, Section 4 provides some details about db4o - an open source object database, Section 5 describes the automatic image annotation process based on an object oriented approach, Section 6 provides a description of the modules included in system's architecture and some details about the evaluation of the annotation system and Section 7 concludes the paper.

2 Related Work

Object recognition and image annotation are very challenging tasks. For this reason a number of models using a discrete image vocabulary have been proposed for the image annotation task. Mori et al. [1] used a Co-occurrence Model in which they looked at the co-occurrence of words with image regions created using a regular grid.

Duygulu et al [2] described images using a vocabulary of blobs. For each image region 33 features such as color, texture, position and shape information were computed. The regions were clustered using the K-means clustering algorithm into 500 clusters called “blobs".

Jeon et al. [3] viewed the annotation process as analogous to the cross-lingual retrieval problem and used a Cross Media Relevance Model to perform both image annotation and ranked retrieval. This model was used in [21] to annotate images from the medical domain. There are other models like Correlation LDA proposed by Blei and Jordan [4] that extends the Latent Dirichlet Allocation model to words and images. In [5] it is proposed the use of the Maximum Entropy approach for the task of automatic image annotation.

In [6][7] it is described a real-time ALIPR image search engine which uses multi resolution 2D Hidden Markov Models to model concepts determined by a training set. A computational efficiency is obtained in [19] due to a fundamental change in the modeling approach. In [6] every image was characterized by a set of feature vectors residing on grids at several resolutions.

An improved model of CMRM is proposed in [8], the Continuous Cross-Media Relevance Model (CRM) which preserves the continuous feature vector of each region and this offers more discriminative power. A further extension of the CRM model called the Multiple Bernoulli Relevance Model (MBRM) is presented in [9].

In [16] is described Oxalis, a distributed image annotation architecture allowing the annotation of an image with diagnoses and pathologies. In [18] it is described the SENTIENT-MD (Semantic Annotation and Inference for Medical Knowledge Discovery) a new generation medical knowledge annotation and acquisition system. In [14] it is presented a hierarchical medical image annotation system using Support Vector Machines (SVM) - based approaches.

3 The Annotation Model and SAIAPRTC-12 Dataset

The Cross Media Relevance Model is a non-parametric model for image annotation and assigns words to the entire image and not to specific blobs. A test image *I* is annotated by estimating the joint probability of a keyword *w* and a set of blobs:

$$P(w, b_1, \dots, b_m) = \sum_{J \in T} P(J) P(w, b_1, \dots, b_m | J). \quad (1)$$

$P(w, b_1, \dots, b_m | J)$ represents the joint probability of keyword *w* and the set of blobs $(b_1, \dots, b_m)$ conditioned on training image *J*. In CMRM it is assumed that, given image *J*, the events of observing a particular keyword *w* and any of the blobs $(b_1, \dots, b_m)$ are mutually independent. This means that $P(b_1, \dots, b_m | J)$ can be written as:

$$P(w, b_1, \dots, b_m | J) = P(w|J) \prod_{i=1}^{m} P(b_i|J) \,. \quad (2)$$

$$P(w|J) = (1 - \alpha_J) \frac{\#(w,J)}{|J|} + \alpha_J \frac{\#(w,T)}{|T|} \quad . \quad (3)$$

$$P(b|J) = (1 - \beta_J) \frac{\#(b,J)}{|J|} + \beta_J \frac{\#(b,T)}{|T|} \; . \quad (4)$$

where:

a) *P(w|J)* , *P(b|J)* denote the probabilities of selecting the word *w*, the blob *b* from the model of the image *J*.

b) *#(w, J)* denotes the actual number of times the word *w* occurs in the caption of image *J*, *#(w, T)* is the total number of times *w* occurs in all captions in the training set *T* .

c) *#(b, J)* reflects the actual number of times some region of the image *J* is labeled with blob *b*, *#(b, T)* is the cumulative number of occurrences of blob *b* in the training set.

d) |*J*| stands for the count of all words and blobs occurring in image *J*, |*T*| denote the total size of the training set.

e) The prior probabilities *P(J)* can be kept uniform over all images in *T*. The smoothing parameters α and β were used as: α = 0.1 and β = 0.9.

We have used for our experiments made on natural images the segmented and annotated SAIAPR TC-12 [12] benchmark which is an extension of the IAPR TC-12 [13] collection for the evaluation of automatic image annotation methods.. Each image was manually segmented using a Matlab tool named Interactive Segmentation and Annotation Tool (ISATOOL). Each region has associated a segmentation mask and a label from a predefined vocabulary of 275 labels. For each pair of regions the following relationships have been calculated: adjacent, disjoint, beside, X-aligned, above, below and Y-aligned. The following features have been extracted from each region: area, boundary/area, width and height of the region, average and standard deviation in x and y, convexity, average, standard deviation and skewness. The dataset contains several folders of images, each folder having the following structure: *images* folder contains the initial images that were manually segmented, *segmentation_masks* folder contains for each image region a file having the extension .mat (Matlab files) that is representing a segmentation mask, *single_mask folder* contains a single .mat file per image, representing the mask of the entire image, *spatial_relationships* contains a file per image with information about the spatial relationships detected between each pair of regions, *segmented_images folder* contains manually segmented images, *features.txt* contains the values of the extracted features from each region, *labels.txt* file contains the information needed to identify the words assigned to each image region, *ontology_path.txt* file contains the path in the ontology for each word associated to a region.

4 db4o (Database for Objects)

db4o [17] is an open-source object-oriented database having bindings to both the .NET and Java platforms and allowing the data objects to be stored exactly in the way they are defined by the application. Unlike string-based query languages db4o offers truly native and object-oriented data access APIs like language integrated queries for querying the database, query by example, retrieval by object graph. The elimination of data transformation in db4o leads to less demand on CPU or persistence operations, which shifts critical resources to the application logic and query processing [20].

For db4o there are available the following methods for querying objects:

a) *Query by Example (QBE)* - a query expression is based on template objects beeing fast and simple to implement. This method is an optimal solution for simple queries that are not using logical operators.

b) *Simple Object Data Access (SODA)* – a query expression is based on query graphs. This method builds a query graph by navigating references in classes and imposing constraints. SODA has several disadvantages [19] because a query is expressed as a set of method calls that explicitly define the graph and it is not similar in any way to traditional querying techniques.

c) *Native Queries (NQ)* – this querying approach express the query in a .NET or Java – compliant language by writting a method that returns a boolean value. The method is applied to all objects stored and the list of matching object instances is returned.

d) LINQ (Language Integrated Query) - is the recommended db4o querying interface for .NET platforms. LINQ allows you to write compile checked db4o queries, which can be refactored automatically when a field name changes and which are supported by code auto-completion tools.

db4o offers support for client/server interactions, each interaction beeing one of the following three types:

a) Networking – is the traditional way of operating in most database solutions. Remote clients open a TCP/IP connection to send/retrieve data to/from the db4o server.

b) Embedded – the client and the server are run on the same machine. The communication between the server and the client is the same as in networking mode but the work is entirely made within one process.

c) Out-of-band signalling - the information sent does not belong to the db4o protocol and does not consist of data objects, but instead is completely user-defined. This mode uses a message passing communication interface.

5 Automatic Image Annotation Based on an Object Oriented Approach

Our system uses the following classes presented in Table 1.

Table 1. The classes used by the system

Classes	*Members*	*Member's Type*
Image	PictureName	String
	Regions	List<Region>
Region	Index	int
	AssignedBlob	Blob
	AssignedWord	Word
	FeaturesVectorItem	FeaturesVector
	MatrixFilePath	String
Blob	Index	int
	AverageFeaturesVector	FeaturesVector
FeaturesVector	Features	List<double>
Word	Name	String
	OriginalIndex	int
RegionsRelationship	RegionA	Region
	RegionB	Region
	RelationshipMode	String
HierarchicalRelationship	ParentWord	Word
	ChildWord	Word

The annotation process contains the following steps for each new image:

1) *Image's segmentation and features' extraction*– using the segmentation algorithm it is obtained a list of image's regions. From each region it is extracted a feature vector. At the end of this process it is obtained a list of Region objects, List<Region>. This list is representing the input for the annotation module and a list of Word objects that describe the semantic content of the image will represent the output. This can be represented as:

List<Region> regions = SegmentImage(imagePath);

2) *Identifying the Blob object corresponding to a Region object and the list of distinct Blob objects assigned to the current image* – in our implementation a Blob object contains an AverageFeaturesVector object obtained by making an average of all FeaturesVector objects belonging to the regions assigned to that blob. In order to identify the Blob object that should be assigned to a new Region object it is computed the Euclidian distance between the AverageFeaturesVector object belonging to a Blob object and the FeaturesVectorItem belonging to that region. The Blob object for which it is obtained the minimum distance will be assigned to the Region object. For the annotation process we need also the list of distinct Blob objects assigned to the current image. For the object oriented approach this list can be computed using the following method based on LINQ:

```
Public List<Blob> DetectBlobs (List<Region> regions){
IObjectContainer db = Db4oFactory.OpenFile("Database.yap");
List<Blob> distinctBlobs = new List<Blob>();
foreach (Region region in regions){
//using LINQ to obtain a list of ComputedDistance objects
IEnumerable<ComputedDistance> distances=from Blob blob in db
  select newComputedDistance {
```

```
  Distance = ComputeDistance(blob.AverageFeaturesVector,
           region.FeaturesVector),
  BlobItem = blob };
//sorting ascending the list and selecting the first value
ComputedDistance min = distances.OrderBy(pd =>
pd.Distance).First();
//assigning the corresponding Blob object to the Region
object
region.Blob = min.BlobItem ;
//add the Blob object in the list if not already added
if (!distinctBlobs.Contains(min.BlobItem)){
      distinctBlobs.Add(min.BlobItem);}}
 return distinctBlobs;}
```

3) Estimating the joint probability of each Word object w based on the set of Blob objects detected above – basically this is equivalent to the estimation of the joint probability of a keyword *w* and a set of blobs described by equation (1).

Table 2 presents the used mapping.

Table 2. The mapping used by the system

CMRM model	*Object oriented model*
$P(w\|J)$	*public double PWJ(Word w, Image J, IObjectContainer db, int cardT)*
$P(b\|J)$	*public double PBJ(Blob b, Image J, IObjectContainer db, int cardT)*
$P(w, b_1, \dots, b_m\|J)$	*public double PWBsJ(Word w, List<Blob> blobs, Image J, IObjectContainer db, int cardT)*
$P(w, b_1, \dots, b_m)$	*public double PWBs(Word w, List<Blob> blobs, List<Image> T, IObjectContainer db, int cardT)*

This estimation can be made using the following statements based on LINQ:

```
//Obtaining the list of all Word objects from the database
IEnumerable<Word> words = from Word w in db select w;
//Obtaining the list of all Blob objects from the database
 IEnumerable<Blob> allBlobs = from Blob b in db select b;
//Obtaining the list of all Image objects from the training
set T existing in  the database
 IEnumerable<Image> T  = from Image img in db select img;
//Equivalent to |T|
int cardT = words.Count() + allBlobs.Count();
//The list of Region objects detected at step 1
List<Region> regions = SegmentImage(imagePath);
//The list of distinct Blob objects detected at step 2
List<Blob> blobs = DetectBlobs(regions);
//This list will contain the probabilities computed for each
Word object
List<Probability> probabilities = new List<Probability>();
```

```
double value;
foreach (Word w in words)
{//Calculating the probability for the Word w
   value = PWBs(w, blobs, T,db, cardT);
  //Creating a new Probability object
   Probability p = new Probability();
   p.Word = w; p.Value = value;
   probabilities.Add(p);}
//The list of probabilities is sorted in a descending order
based on the Value field
probabilities = probabilities.OrderByDescending(pd =>
pd.Value).ToList();
```

Let us suppose that the number of words that should be assigned to the image is *n*. In this case the first *n* elements from the probabilities list will be selected.

6 System's Architecture and Evaluation of the Annotation System

System's architecture is presented in Figure 3 and it is based on client server interactions. There are two main componenents: a client component and a server component that are communicating based on the principles defined for the *Networking* and *Out-of-band signalling* interaction types provided by d4bo. The Client component is used to perform the following operations for each new image that need to be annotated: image segmentation - at the end of this process it is obtained a list of Region objects, features extraction – from each region it is extracted a feature vector and this is used to create a new FeaturesVector object; at the end of this process each Region object will have associated the corresponding FeaturesVector object, sending the list of Region objects using the *dbo Client Module* to the *db4o Server Module* to perform the annotation process, using the *dbo Client Module* to retrieve the list of *n* Word objects detected by the annotation process; this list is provided by the db4o *Server Module.* The Server component can be used to perform the following operations: obtaining the list of Blob objects using the Clustering

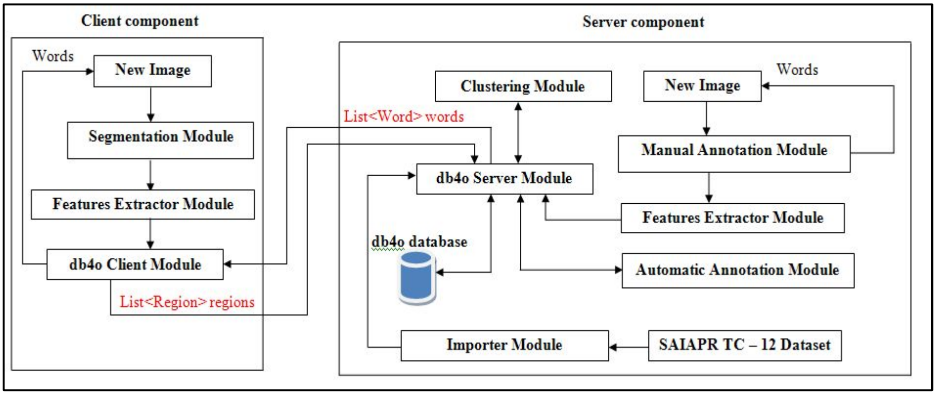


Fig. 1. System's architecture

module, importing the content provided by the SAIAPRTC-12 Dataset, performing a manually annotation of images, performing an automatic image annotation. The total number of distinct modules contained in this architecture is 8.

Each module is described below:

a) *Importer Module* – this module is used to extract the existing information from the SAIAPRTC-12 Dataset. The import process stores the created objects in the database and contains the following steps:

1) *Importing images and image's regions* – using the content of two folders (images, segmentation_masks) it is determined the content of each region. At the end of this process it will be obtained a list of Image and Region objects.
2) *Importing features* – the content of the *features.txt* file is analyzed. Each line in this file contains the features extracted from a specific region and it is used to create a new FeaturesVector object. At the end of this process it will be obtain a list of FaturesVector objects.
3) *Detecting the spatial relationships between regions* - the content of the *spatial_rels* folder is processed and the spatial relationships are detected and represented as RegionsRelationship objects.
4) *Importing ontology's paths* – the content of the *ontology_path.txt* file is analyzed line by line. Each hierarchical relationship will represent as a HierarchicalRelationship object.
5) *Importing the list of all words* – the content of the words' file is processed line by line and each word will be represented as a Word object.
6) *Importing the words assigned to regions* – the content of the *labels.txt* files is processed and it is detected the word assigned to each region.

b) *Segmentation Module* – this module is using the segmentation algorithm described in [13] to obtain a list of regions from each new image. For each region detected it is created a Region object.

c) *Features extractor Module* - for each segmented region it is computed a feature vector that is represented as a FeaturesVector object.

d) *Clustering Module* - we have used K-means algorithm to quantize the FeaturesVector objects obtained from the training set and to generate a list of Blob objects.

e) Automatic Annotation Module – this module is using the algorithm described earlier to annotate a new image.

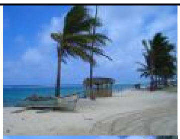
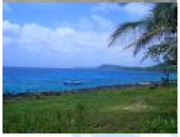
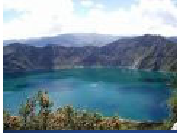
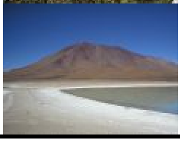
f) *Manual Annotation Module* – this module can be used to manually annotate images.

g) *db4o Server Module* – this module is the central piece used by the modules to communicate with the database. This component is listening for connections on a specific port and only the clients having appropriate credentials will be allowed to connect and to perform operations.

h) *db4o Client Module* **–** this module is used to send the list of Region objects to the *db4o Server Module* for the automatic image annotation process and to retrieve the list of Word objects obtained using this process.

In order to evaluate the annotation system we have used a testing set of 400 images that were manually annotated and not included in the training set used for the CMRM model. This set was segmented using the segmentation algorithm mentioned above and a list of words having the joint probability greater than a threshold value was assigned to each image. Then the number of relevant words automatically assigned by the annotation system was compared against the number of relevant words manually assigned by computing a recall value. Using this approach for each image we have obtained a statistic evaluation having the structure presented in Table 3.

Table 3. The statistic evaluation

Index	*Image*	*Relevant words automatically assigned (RWAA)*	*Words manually assigned (WMA)*	*Recall = RWAA/ WMA*
0		sky-blue, sand-beach, ocean	sand-beach, ocean, boat, palm, hut, sky-blue	3/6 = 0.50
1		sky-blue, grass, ocean, cloud	grass, ocean, boat, cloud, sky-blue, branch	4/6 = 0.66
2		sky, mountain, lake	lake, vegetation, mountain, cloud, sky	3/5 = 0.60
3		mountain, sky-blue, sand-dessert	mountain, lake, sand-dessert, sky-blue	3/4 = 0.75

After computing the recall value for each image it was obtained a medium recall value equal to 0.73.

7 Conclusions and Future Work

In this paper we described an original image annotation system based on an open source object database. TC-12 benchmark contains a large-size image collection comprising diverse and realistic images, including an annotation vocabulary having a hierarchical organization. The CMRM annotation model implemented by the system was proven to be very efficient by several studies. Object oriented databases expose means through which objects can be queried and stored using the same model that it employed by the application's programming language. The experimental results have shown that the annotation model in combination with db4o can produce good results. Further extensions of the system will include the two models of image retrieval provided by CMRM: Annotation-based Retrieval Model and Direct Retrieval Model.

References

1. Mori, Y., Takahashi, H., Oka, R.: Image-to-word transformation based on dividing and vector quantizing images with words. In: MISRM 1999 First Intl. Workshop on Multimedia Intelligent Storage and Retrieval Management (1999)
2. Duygulu, P., Barnard, K., de Freitas, J.F.G., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
3. Michael, D.B., Jordan, M.I.: Modeling annotated data. In: To Appear in the Proceedings of the 26th Annual International ACM SIGIR Conference
4. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic Image Annotation and Retrieval using Cross-Media Relevance Models. In: Proceedings of the 26th Intl. ACM SIGIR Conf., pp. 119–126 (2003)
5. Jeon, J., Manmatha, R.: Using maximum entropy for automatic image annotation. In: CIVR, pp. 24–32 (2004)
6. Li, J., Wang, J.: Automatic linguistic indexing of pictures by a statistical modeling approach. IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003)
7. Li, J., Wang, J.: Real-time computerized annotation of pictures. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(6), 985–1002 (2008)
8. Lavrenko, V., Manmatha, R., Jeon. J.: A model for learning the semantics of pictures. In: Proceedings of Advances in Neural Information Processing Systems, NIPS (2004)
9. Feng, S.L., et al.: Multiple bernoulli relevance models for image and video annotation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1242–1245 (2004)
10. Rong, J., Joyce, Y.C., Luo, S.: Effective automatic image annotation via a coherent language model and active learning. In: Proceedings of ACM International Conference on Multimedia (ACM MULTIMEDIA), pp. 892–899 (2004)
11. Burdescu, D., Brezovan, M., Ganea, E., Stanescu, L.: A New Method for Segmentation of Images Represented in a HSV Color Space. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2009. LNCS, vol. 5807, pp. 606–617. Springer, Heidelberg (2009)
12. Segmented and Annotated IAPR TC-12 dataset, `http://imageclef.org/SIAPRdata`
13. IAPR TC-12 Benchmark, `http://imageclef.org/photodata`
14. Igor, F.A., Filipe, C., Joaquim, F., da Pinto, C., Jaime, S.C.: Hierarchical Medical Image Annotation Using SVM-based Approaches. In: Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine (2010)
15. Ganea, E., Brezovan, M.: An Hypergraph Object Oriented Model for Image Segmentation and Annotation. In: Proceedings of the International Multiconference on Computer Science and Information Technology, pp. 695–701 (2010)
16. Daniel, E.: OXALIS: A Distributed, extensible ophthalmic image annotation system, Master of Science Thesis (2003)
17. db4objects, `http://www.db4o.com/`
18. Baoli, L., Ernest, V.G., Ashwin, R.: Semantic Annotation and Inference for Medical Knowledge Discovery. In: NSF Symposium on Next Generation of Data Mining (NGDM-2007), Baltimore, MD (2007)
19. Paterson, J., Edlich, S., Hoerning, H., Hoerning, R.: The Definitive Guide to db4o. Apress (2006)
20. Db4o Developer Community, `http://developer.db4o.com/`
21. Mihai, C.G., Stanescu, L., Burdescu, D.D., Brezovan, M., Spahiu, C.S., Ganea, E.: Annotation system for medical domain. In: Corchado, E., Snášel, V., Sedano, J., Hassanien, A.E., Calvo, J.L., Ślęzak, D. (eds.) SOCO 2011. AISC, vol. 87, pp. 579–587. Springer, Heidelberg (2011)

A Novel Multi-Objective Genetic Algorithm for Clustering

Oliver Kirkland, Victor J. Rayward-Smith, and Beatriz de la Iglesia

School of Computing Sciences
University of East Anglia
Norwich
UK
{O.Kirkland,B.Iglesia,Vjrs}@uea.ac.uk
http://www.uea.ac.uk/cmp

Abstract. In this paper, we introduce a new Multi-Objective Clustering algorithm (MOCA). The use of Multi-Objective optimisation in clustering is desirable because it permits the incorporation of different criteria for cluster quality. Since the criteria to establish what constitutes a good clustering is far from clear, it is beneficial to develop algorithms that allow for multiple criteria to be accommodated.

The algorithm proposes a new implementation of multi-objective clustering by using a centroid based technique. We explain the implementation details and perform experimental work to establish its worth. We construct a robust experimental set up with a large number of synthetic databases, each with a pre-defined optimal clustering solution. We measure the success of the new MOCA by investigating how often it is capable of finding the optimal solution. We compare MOCA with k-means and find some promising results. MOCA can generate a pool of clustering solutions that is more likely to contain the optimal clustering solution than the pool of solutions generated by *k*-means.

1 Introduction

Multi-Objective Evolutionary Algorithms (MOEA) have some unexplored potential for cluster analysis. Clustering algorithms optimise a specific measure of cluster quality, such as compactness and separation. Many clustering algorithms have been defined in the literature [7] and they generally aim to optimise a single objective. Unfortunately, defining what constitutes a good clustering solution remains a difficult problem and no individual measure of clustering quality has emerged as the overall winner. In this context, MOEAs give us the opportunity to optimise several of these quality measures at once. Furthermore, they will then deliver a number of clustering solutions representing trade-offs between the different quality measures.

Previous research into evolutionary algorithms for clustering has been conducted by Cole [3] who explored various techniques for representing clustering solutions and various objectives to be optimised. Handl and Knowles [5] and

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 317–326, 2011.

Chen and Wang [1] have developed their own Multi-Objective Clustering Algorithms (MOCA) that operate with new cluster quality measures. These previous works have used different methods such as graph based technique to assign objects to clusters. Here we will use a new centroid based technique to establish membership.

In this paper, we propose a new MOCA and evaluate its performance against the well known k-means algorithm. In section 2, we define some of the notation used; in section 3 we propose a new Multi-Objective Cluster Algorithm; in section 4, we propose a method of assessing its quality; finally, we report out results in section 5 and give our conclusions in section 6.

2 Notation Definition

We define an object, x, as a d dimensional feature vector, $x = \left(x^1, \ldots, x^d\right)$. Each element, x^e, of the vector is a number from the real domain, $x^e \in \mathbb{R}$. A data set, $\mathcal{D}$, is a set of n of these objects $\mathcal{D} = \{\mathcal{D}(1), \ldots, \mathcal{D}(n)\}$. For any two objects in the data set, $\mathcal{D}(i)$ and $\mathcal{D}(j)$, we can compute the distance between them using the Euclidian distance metric:

$$\delta\left(\mathcal{D}(i), \mathcal{D}(j)\right) = \sqrt{\sum_{e=1}^{d} \left(\mathcal{D}(i)^e - \mathcal{D}(j)^e\right)^2} \ . \tag{1}$$

The data set, $\mathcal{D}$, can be partitioned to form a set of k subsets, $\mathcal{P}$, representing a clustering solution $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_k\}$ where each subset, $\mathcal{P}_g$, represents a cluster. In this work, we are concerned with complete, non-overlapping, clustering solutions. That is to say,

- all objects must belong to at most one cluster: $\mathcal{P}_g \cap \mathcal{P}_h = \emptyset, \forall \mathcal{P}_g, \mathcal{P}_h \in \mathcal{P}$ where $\mathcal{P}_g \neq \mathcal{P}_h$;
- all objects must belong to a cluster so no objects are classified as outliers or noise: $\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k = \mathcal{D}$; and,
- no cluster is allowed to be empty: $\mathcal{P}_g \neq \emptyset$.

The centroid, $\mathcal{V}_g$, of a cluster, $\mathcal{P}_g$, is a d dimensional feature vector representing the centre point of the cluster. The centroid may, or may not, correspond to a member of the cluster. To compute the centroid the value of each dimension must first be calculated. The e^{th} dimension is calculated as $\mathcal{V}_g e = \left(\sum_{i=1}^{|\mathcal{P}_g|} \mathcal{P}_g(i)^e\right) / |\mathcal{P}_g|$ where $|\mathcal{P}_g|$ is the number of objects in cluster $\mathcal{P}_g$. Thus, the centroid of $\mathcal{P}_g$ is given as $\mathcal{V}_g = \left\{\mathcal{V}_g^1, \ldots, \mathcal{V}_g^d\right\}$.

The centroid, $\mathcal{V}$, of the data set, $\mathcal{D}$, is calculated in a similar fashion; each of its dimensions are calculated as the mean of the values of all the objects in the data set, $\mathcal{V}_e = \left(\sum_{i=1}^{n} \mathcal{D}(i)^e\right) / n$. The vector representing the centroid is then $\mathcal{V} = \{\mathcal{V}_1, \ldots, \mathcal{V}_d\}$.

3 Multi-Objective Clustering Algorithm (MOCA)

Evolutionary algorithms, used in single objective optimisation problems, start with an initial pool of possible solutions for a specified problem. These solutions are evaluated according to some objective function and solutions are then selected for reproduction. In reproduction, solutions are mutated and combined to create new solutions. Finally, new solution pools are created from the original and the newly created solutions. During this process, the principle of "survival of the fittest" is used to select solutions for reproduction and for selecting solutions to keep for the next iteration. This process is continued until some stopping criteria is met. An optimal or semi-optimal solution generally emerges from the process.

Multi-objective optimisation problems aim to find the vector of solutions $\overrightarrow{x}^* = [x_1^*, x_2^*, \cdots x_n^*]^T$ which will satisfy m inequality constraints $g_i(\overrightarrow{x}) \geq 0, \quad i = 1, 2, \cdots, m$ and will optimise the vector function $\overrightarrow{f}(\overrightarrow{x}) = [f_1(\overrightarrow{x}), f_2(\overrightarrow{x}), \cdots, f_r(\overrightarrow{x})]^T$. Hence in MOEAs, the solutions are evaluated according to a number, r, of sometimes conflicting objective functions. $\mathcal{F}$ denotes the feasible region of the problem (i.e., where the constraints are satisfied).

The result of an MOEA is typically a Pareto front, which is a set of solutions representing compromises or "trade offs" between the objectives. The Pareto front may contain solutions that are very good in respect to one objective function but are very bad with respect to another objective. More formally, assuming we are minimising, we say that a vector of decision variables $\overrightarrow{x}^* \in \mathcal{F}$ is *Pareto optimal* if there does not exist another $\overrightarrow{x} \in \mathcal{F}$ such that $f_i(\overrightarrow{x}) \leq f_i(\overrightarrow{x^*})$ for all $i = 1, ..., k$ and $f_j(\overrightarrow{x}) < f_j(\overrightarrow{x^*})$ for at least one j. The Pareto front obtained by a MOEA should have both good coverage (all areas of the Pareto space should be represented) and good convergence (solutions should be Pareto optimal). Poor coverage may be obtained if diversity in the solutions has not been maintained from one generation to the next.

We have chosen to use NSGA-II [4], one of the best known MO algorithm, as the underlying implementation for our MOCA. NSGA-II introduced techniques for producing a set of solutions that provide good coverage and convergence. To adapt NSGA-II for clustering requires the following:

- an appropriate representation of a clustering solution,
- a set of evaluation functions for a clustering solution,
- an initialisation operator that creates valid solutions,
- a mutation operator,
- a crossover operator.

Additional parameters are used to define a minimum and maximum number of clusters allowed, $k_{\min}$ and $k_{\max}$ respectively. Sensible values are $k_{\min} = 2$ and

$k_{\max} = n/2$ but the decision maker may use any values as long as $1 \leq k_{\min} \leq k_{\max} \leq n$ is true.

3.1 Solutions Representation and Initialisation

The solution representation consists of two sets of cluster prototypes: the set of selected prototypes $\mathcal{A} = \{\mathcal{A}(1), \ldots, \mathcal{A}(a)\}$ and the set of potential prototypes, not in use, $\mathcal{B} = \{\mathcal{B}(1), \ldots, \mathcal{B}(b)\}$. Therefore, each cluster $\mathcal{P}_g$ in the represented clustering solution is associated with a cluster prototype, $\mathcal{A}(g)$, from set $\mathcal{A}$.

To generate initial valid clustering solutions, the values of the cluster prototypes are drawn from $\mathcal{D}$, hence the initial prototypes are medoids. The lengths of $\mathcal{A}$ and $\mathcal{B}$ are required to create the initial solutions. The value of a is set to $k_{\min} + (k_{\max} - k_{\min})/2$ and b is set to $n - a$. Each object from the data set is then randomly added to $\mathcal{A}$ or $\mathcal{B}$.

Once a set of selected prototypes has been defined, the distance between every object in the data set, $x \in \mathcal{D}$, and every cluster prototype, $\mathcal{A}(g) \in \mathcal{A}$, is calculated. x is added to the cluster that minimises $\delta(x, \mathcal{A}(g))$.

3.2 Mutation Operator

The mutation operator encompasses three techniques for altering a solution. Each alteration is applied with a probability: a 50% probability of decreasing the number of prototypes in $\mathcal{A}$, a 25% probability of increasing the number of prototypes in $\mathcal{A}$, or a 25% probability of recomputing the cluster prototypes. The techniques are defined as follows:

Decrease. A cluster prototype is moved from $\mathcal{A}$ to $\mathcal{B}$. To determine the prototype to remove, we first identify the nearest neighbour, $\mathcal{A}(g)_{\mathcal{A}NN}$, of every cluster prototype $\mathcal{A}(g)$ in $\mathcal{A}$. We then move the prototype that minimises $\delta(\mathcal{A}(g), \mathcal{A}(g)_{\mathcal{A}NN})$, $\mathcal{A}(g) \in \mathcal{A}$. The objects associated with the removed prototype, $\mathcal{A}(g)$, are likely to be associated with $\mathcal{A}(g)_{\mathcal{A}NN}$ after the removal.

Increase. Similarly, a cluster prototype is moved from $\mathcal{B}$ to $\mathcal{A}$. The prototype drawn from $\mathcal{B}$ is the cluster prototype that is furthest away from any cluster prototype in $\mathcal{A}$. That is, for each cluster prototype, $\mathcal{A}(g) \in \mathcal{A}$, its furthest neighbour in $\mathcal{B}$, $\mathcal{A}(g)_{\mathcal{B}FN}$, is computed. The cluster prototype in $\mathcal{B}$ that maximises $\delta(\mathcal{A}(g), \mathcal{A}(g)_{\mathcal{B}FN})$ is moved to $\mathcal{A}$. This ensures that new cluster prototypes are not near pre-existing cluster prototypes so they should produce new and interesting clusters.

Recompute Prototypes. The values of the cluster prototypes are recomputed as the value of the centroids of the clusters with which they are associated. For example, the value of a cluster prototype, $\mathcal{A}(g)$, will be replaced with the value of $\mathcal{V}_g$ where $\mathcal{V}_g$ is the centroid of $\mathcal{P}_g$. This process is similar to a single iteration of the clustering algorithm, k-means.

3.3 Crossover Operator

Given two clustering solutions, we identify the largest, $\mathcal{P}^l$, and smallest, $\mathcal{P}^s$, according to the number of clusters, k. If there is a tie the designation is random.

For the smaller solution, $\mathcal{P}^s$, we then identify the largest cluster, $\mathcal{P}_g^s \in \mathcal{P}^s$, and its prototype, $\mathcal{A}(g)^s$.

For each object $x \in \mathcal{P}_g^s$ we determine the cluster in $\mathcal{P}^l$ in which is lies and the associated prototype. Let $\left\{\mathcal{A}(1)^l, \ldots, \mathcal{A}(o)^l\right\}$ denote the set of prototypes obtained in this was. The crossover operation then exchanges prototype $\mathcal{A}(g)^s$ in the small solution with all the prototypes $\left\{\mathcal{A}(1)^l, \ldots, \mathcal{A}(o)^l\right\}$ associated with it in the larger solution.

To ensure that $a + b = n$, we randomly remove the required number of prototypes from the set $\mathcal{B}$ in the smaller solution and add them to the set $\mathcal{B}$ of the larger solution.

The resulting crossover is therefore an exchange of one cluster in one solution with the corresponding smaller clusters in the other solution.

3.4 Fitness Measures for MOCA

Density Based Fitness Measure. A common measure of the quality of a clustering solution is the density of the clusters. A clustering solution is dense if the distances between the objects in each cluster are low. This can be measured by taking the average of the distance between each object in the cluster and its centroid. The density of each cluster can then be summed to give the Average Within Group Sum of Squares for a given clustering solution:

$$\mathrm{awgss}(\mathcal{P}) = \sum_{g=1}^{k} \frac{\sum_{i=1}^{|\mathcal{P}_g|} \delta\left(\mathcal{P}_g(i), \mathcal{V}_g\right)^2}{|\mathcal{P}_g|} \;. \tag{2}$$

This measure does not have a bias for large clusters. Values of this measure are high when the clusters are not very dense so this measure should be minimised.

Separation Based Fitness Measure. Another method of assessing the quality of a clustering solution is how well separated are the clusters. A clustering solution is considered good if the clusters are well separated.

We define the Average Between Group Sum of Squares of a clustering solution, $\mathrm{abgss}(\mathcal{P})$, as the average distance between the centroids of the clusters and the centroid of the data set:

$$\mathrm{abgss} = \frac{\sum_{g=1}^{k} |\mathcal{P}_g| \, \delta\left(\mathcal{V}_g, \mathcal{V}\right)^2}{k} \;. \tag{3}$$

A low value of $\mathrm{abgss}(\mathcal{P})$ would indicate that all of the cluster centroids are near the centroid of the data set and therefore also near each other, so the value of this measure should be maximised. The value is normalised by k to avoid bias.

Connectivity Based Fitness Measure. The concept of nearest neighbour consistency has been extended to clustering by Ding [2]. Under this, a clustering is considered good if each object is contained within the same cluster as its nearest neighbours.

Handl and Knowles [6] have proposed the measure Connectivity which takes into account the violation of nearest neighbour consistency for a given clustering solution. Connectivity calculates the sum of the values of a penalty function for each object in the data set and its l nearest neighbours. A penalty for an object, x, and its m^{th} nearest neighbour, x_{mNN}, is 0 if they are contained in the same cluster and $\frac{1}{m}$ if they are not members of the same cluster. The quality measure is defined as follows:

$$\text{connectivity}\left(\mathcal{P}\right) = \sum_{i=1}^{n} \sum_{m=1}^{l} \text{penalty}\left(\mathcal{D}\left(i\right), \mathcal{D}\left(i\right)_m\right) \tag{4}$$

$$\text{where} \qquad \text{penalty}\left(x_{mNN}\right) = \begin{cases} \frac{1}{m} & \text{if } \nexists\, \mathcal{P}_g : x \in \mathcal{P}_g \wedge x_{mNN} \in \mathcal{P}_g, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The value of this measure increases as nearest neighbour consistency is violated so this measure should be minimised.

4 Experimentation

To test our MOCA we shall perform a comparison between its performance and the performance of k-means when clustering a large number of prefabricated data sets where a desired clustering solution exists.

4.1 Construction of Synthetic Data Sets

For this study, we constructed a large number of synthetic data sets building upon a technique used by Milligan and Cooper [8] who performed an evaluation of internal cluster quality measures. For the study, they produced one hundred and eight synthetic data sets by identifying three factors and crossing them to produce data sets. We extended their method with an additional parameter and larger ranges of possible values. Milligan has explained in detail the method for generating the data sets [9]; it is briefly summarised here and then expanded upon.

To generate data objects we must first identify the boundaries of each cluster where points are to be generated. The boundaries of the clusters may not overlap in the first dimension. The length of the boundaries is selected from a uniform distribution running from ten to forty. The centroid of each cluster is then determined. The value of the centroid for a given dimension is the midpoint of its boundary for that dimension. The standard deviation of a cluster for a given dimension is defined as a third of the length of its boundary for that dimension. Points are generated using a multivariate normal distribution with the centroid of the distribution defined as the centroid of the cluster to be generated. The diagonal entries of the variance-covariances matrix are set to the standard deviations of each dimension of the cluster. Each point that is generated must be within 1.5 standard deviations of the centroid. The process is repeated for each cluster that is to be generated.

In our experimentation we consider four factors. The first factor is the number of clusters in a data set: values between two and forty are used. The second factor is the number of dimensions: values range from two to twenty dimensions within Euclidean space so that no one dimension dominates the other dimensions. The third factor is the proportion of objects that are members of each cluster. For this, there are three possible designs: in the first design the objects are evenly distributed between all of the clusters; in the second design a cluster consists of 10% of the objects and the rest are as evenly distributed as possible; in the third design, a cluster consists of 60% of the objects and the rest are as evenly distributed as possible. Finally, the fourth factor is the proportion of objects that shall be generated as outliers. Outliers will be within 9 standard deviations of the centroid of each cluster instead of 1.5 standard deviations. The proportion shall be either: 0, 20% or 40% percent of the objects which will be generated as outliers.

Each of these four factors are varied to obtain six thousand six hundred and sixty nine different data set designs. Each design was generated three times resulting in a final set of twenty thousand and seven data sets. Each data set consists of five hundred objects.

4.2 Comparison of Clustering Solutions

An external cluster quality measure is a method that evaluates the quality of a clustering solution, $\mathcal{P}$, against a known optimal clustering solution, $\mathcal{P}'$. The optimal solution is designated as such because it has been labelled by a human or has been specifically generated for the purpose. As our data sets have been specifically generated, we have designated for each the "optimal" clustering. Here we have chosen to use the Rand Statistic [10] which measures the similarity of any two clustering solutions.

A pair of objects, $\mathcal{D}(i)$ and $\mathcal{D}(j)$, are classified as follows:

SS. If $\mathcal{D}(i) \in \mathcal{P}_g$, $\mathcal{D}(j) \in \mathcal{P}_g$, $\mathcal{D}(i) \in \mathcal{P}_h$ and $\mathcal{D}(j) \in \mathcal{P}_h$
SD. If $\mathcal{D}(i) \in \mathcal{P}_g$, $\mathcal{D}(j) \in \mathcal{P}_g$, $\mathcal{D}(i) \in \mathcal{P}_h$ and $\mathcal{D}(j) \notin \mathcal{P}_h$
DS. If $\mathcal{D}(i) \in \mathcal{P}_g$, $\mathcal{D}(j) \notin \mathcal{P}_g$, $\mathcal{D}(i) \in \mathcal{P}_h$ and $\mathcal{D}(j) \in \mathcal{P}_h$
DD. If $\mathcal{D}(i) \in \mathcal{P}_g$, $\mathcal{D}(j) \notin \mathcal{P}_g$, $\mathcal{D}(i) \in \mathcal{P}_h$ and $\mathcal{D}(j) \notin \mathcal{P}_h$

where $\mathcal{P}_g \in \mathcal{P}$ and $\mathcal{P}_h \in \mathcal{P}'$ and **SD** stands for "same" and "different".

The values of a, b, c and d are the numbers of pairs of objects classified as **SS**, **SD**, **DS** and **DD** respectively. From this the Rand Statistic is defined as:

$$\mathcal{R} = \frac{a+d}{a+b+c+d} \ . \tag{6}$$

The value of R is be between 0 and 1. A value of 0 indicates that the solutions are totally dissimilar and a value of 1 indicates that they are identical.

4.3 Experimental Method

We set the population size for NSGA-II at 100; the number of generations was set at 1,000; the mutation probability and the crossover probability were both set to 0.5.

Our MOCA was executed upon each of the previously described synthetic data sets with these parameters. For each set of solutions given by the MOCA the optimal solution was evaluated against it using $\mathcal{R}$. We extracted the highest, lowest and average values of $\mathcal{R}$ recorded for each pool of solutions associated with a data set. The value of k associated with each value of $\mathcal{R}$ is also reported.

We also make a comparison of performance against the algorithm k-means. For each synthetic data set we execute the algorithm k-means for varying values of k ranging from 2 to 40. We report the highest value of $\mathcal{R}$ recorded for each pool of solutions associated with a data set and the associated k value.

5 Results

The results of our experiment are reported in table 1. Our results show that, when looking at the best solution reported by MOCA for each dataset (largest $\mathcal{R}$, reported as *MOCA Best* in table 1) the optimal clustering solution, equivalent to $\mathcal{R} = 1$, was contained in the pool of solutions generated by the MOCA at least once for 18.18% of the data sets. However the optimal solution was drawn from the pool of solutions generated by k-means in only 4.09% of cases (k-means Best column in table 1). Furthermore, when looking at solutions close to the optimal solution ($\mathcal{R} \geq 0.9$) they were found by k-means in 21.37% of cases but by MOCA in 100% of cases.

We also extracted the worst solution from each pool of solutions generated by the MOCA (the minimum value of $\mathcal{R}$, reported as *MOCA Worst*) and found that in 1.07% of cases this value was equal to 1. This shows that in these cases every solution offered by the MOCA was the optimal solution.

We extracted the average solutions reported by MOCA (average $\mathcal{R}$, reported as *MOCA Average*). In 30.11% of cases the average solution was close to the optimal solution. This shows that the average result in a pool of solutions generated by the MOCA is better than the best result drawn from a pool of solutions generated by k-means.

We average our results and found that the average value of $\mathcal{R}$ from the best solutions generated by MOCA was 0.98. This was higher than the average equivalent generated by k-means which was 0.88. Also we calculated the average $\mathcal{R}$ from the worst and average solutions generated by MOCA. They were 0.56 and 0.89 respectively. Hence the average solution generated by MOCA is close to the best solution generated by k-means.

We also extracted the value of k associated with the solutions with the highest, average and lowest values of $\mathcal{R}$ generated by MOCA. Similarly, we extracted k associated with the solutions generated by k-means. We found that MOCA found the correct value of k in 30.54% of cases whereas k-means had the correct value in 8.34% of cases. The best solution drawn from the solutions generated by MOCA

Table 1. Summary of Results

	MOCA Best	MOCA Average	MOCA Worst	k-means Best
$\mathcal{R} \geq 0.9$ total	100%	30.11%	3.79%	21.37%
$\mathcal{R} = 1$ total	18.18%	1.07%	1.07%	4.09%
Max $\mathcal{R}$	1	1	1	1
Min $\mathcal{R}$	0.92	0.38	0.11	0.53
Average $\mathcal{R}$	0.98	0.89	0.56	0.88
StDev $\mathcal{R}$	0.02	0.03	0.11	0.05
Correct value of k	30.54%	0%	1.41%	8.34%
Average difference of k	5.78	20.44	18.9	-2.42
StDev of difference of k	6.50	11.32	11.47	11.67

had 5.78 extra clusters on average and the worst solution had 18.9 extra clusters on average whereas the best solution generated by k-means had 2.42 less clusters than the optimal number of clusters on average.

6 Conclusion and Future Work

We have shown that MOCA can generate a pool of clustering solutions that is more likely to contain the optimal clustering solution than the pool of solutions generated by k-means. The solutions in this pool are generally more similar to the optimal solution than the solutions generated by k-means. The solutions generated by MOCA tend to have many extra clusters whereas k-means tends to have less clusters than the optimal clustering solution.

Future work on the MOCA should be focused on reducing the number of clusters in the solutions as too many clusters are introduced in this implementation. This may be achieved by tuning the parameters of NSGA-II by experimentation. Parameter experimentation of NSGA-II may also reduce the runtime of MOCA. Currently it is not practical for large n. Future work should also include comparing MOCA to other existing clustering algorithms using a wider range of data sets, including real world data sets. Investigation into selecting the "best" solution from the pool of solutions automatically should also be carried out. This may be done computationally or by producing aids for a human to select the "best" solution.

References

1. Chen, E., Wang, F.: Dynamic clustering using multi-objective evolutionary algorithm. Computational Intelligence and Security, 73–80 (2005)
2. Chris, X.H., Ding, H.Q.: K-nearest-neighbor consistency in data clustering: incorporating local information into global optimization. In: ACM Symposium on Applied Computing, pp. 584–589 (2004)
3. Cole, R.: Clustering with genetic algorithms. Master's thesis. University of Western Australia (1998)

4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation 6(2), 182 (2002)
5. Handl, J., Knowles, J.D.: Evolutionary multiobjective clustering. In: Yao, X., Burke, E.K., Lozano, J.A., Smith, J., Merelo-Guervós, J.J., Bullinaria, J.A., Rowe, J.E., Tiňo, P., Kabán, A., Schwefel, H.-P. (eds.) PPSN 2004. LNCS, vol. 3242, pp. 1081–1091. Springer, Heidelberg (2004)
6. Handl, J., Knowles, J.: An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation 11(1), 56–76 (2007)
7. Jain, A., Murty, M., Flynn, P.: Data clustering: a review. ACM Computing Surveys (CSUR) 31(3), 264–323 (1999)
8. Milligan, G.: A Monte Carlo study of thirty internal criterion measures for cluster analysis. Psychometrika 46(2), 187–199 (1981)
9. Milligan, G.: An algorithm for generating artificial test clusters. Psychometrika 50(1), 123–127 (1985)
10. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336), 846–850 (1971)

A Novel Ensemble of Distance Measures for Feature Evaluation: Application to Sonar Imagery

Richard Harrison[1,2,⋆], Roger Birchall[2], Dave Mann[2], and Wenjia Wang[1]

[1] School of Computing Sciences, University of East Anglia, Norwich, NR47TJ, UK
richard.harrison@uea.ac.uk, wenjia.wang@uea.ac.uk
[2] Gardline Geosurvey, Great Yarmouth, NR310NN, UK
roger.birchall@gardline.co.uk, dave.mann@gardline.co.uk

Abstract. Mapping interesting regions in qualitative sidescan sonar imagery predominantly relies on an expensive human interpretation process. It would therefore be useful to automate components of this task with a feature-based, Machine Learning system. We must first establish a framework for reliably and efficiently evaluating the features. A novel ensemble of probabilistic distance measures is proposed, as an objective function for this purpose. The idea is motivated by the fact that different distance measures yield conflicting feature ranking results. In the ensemble, distances can be combined to produce a consensus rank score. As a test case, we find a sub-optimal parameterisation of a Co-occurrence Matrix, for identifying textures peculiar to the tube-building worm, *Sabellaria spinulosa.* A strong correlation is found between ensemble scores and classification accuracies. The proposed methodology is applicable to any sonar imagery, classification task or feature groups.

Keywords: GLCM, saliency, rank disagreement, ensemble, sonar.

1 Introduction

Sidescan sonar imagery contains useful visual information indicating the nature and extent of different physical and morphological regimes on the seabed. Of particular interest are protected habitats, such as reefs formed by colonies of the tube-building worm, *Sabellaria Spinulosa* (Sabellaria) [1]. Mapping the colonies usually involves manual segmentation of Sabellaria textures in the imagery - a tedious and costly process. Automating the identification of Sabellaria would greatly assist in this mapping task.

Examples of features used in Machine Learning approaches to the textural analysis of sonar imagery include: Gabor filter banks [2]; Wavelets [3]; feature fusion [4] and the prevalent Grey Level Co-occurrence Matrices (GLCM) [5] [6]. Yet, there is no published research on the features which should be used for

[⋆] PhD research supported by the Grant (R17336) jointly funded by EPSRC, UEA and Gardline Geosurvey of Great Yarmouth, UK.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 327–336, 2011.

accurate machine identification of Sabellaria in qualitative imagery. Choice of features and their parameterisations depends on factors such as the degree of noise, geometric and radiometric distortion, image resolution and the classification task. To determine sub-optimal feature sets, we must first establish a framework for feature evaluation - the focus of our work herein.

Several methods are available for feature evaluation and selection, e.g. [7]. It is well-known that filtering approaches such as distance or (dis)similarity measures provide an efficient means of evaluating feature saliency. There are in fact, dozens of distance measures to choose from and many of these are described in [8]. However, using different measures on identical underlying distributions, leads to disagreements in the feature rankings. Applying a single measure in all situations is a feasible but myopic approach due to the numerous factors influencing the measured saliency. Consequently, there is no guarantee that saliency information generated by a single measure will be efficacious in all situations. Furthermore, given a choice of conflicting rankings from various distance measures, under identical conditions, it is not clear which one is most useful.

In this paper we propose a new approach to addressing these issues by making use of a fusion of information from multiple measures. We define an ensemble of probabilistic distance measures for measuring the saliency of features generated on sonar imagery. The parameter-free Kullback Leibler Divergence (KL), Chi-Squared (CS), Bhattacharyya (Bh), Euclidean L_2 (Eu), Harmonic Mean (HM) and Tanimoto (Ta) measures are deployed in the ensemble as a test case. We identify some of the factors influencing pairwise rank disagreements between the distance measures, and apply the ensemble to the parameterisation of a co-occurrence matrix for feature generation. For the parameter subspace and data considered, correlation between the ensemble estimated saliency and classification accuracy is very strong. This facilitates a calibration so that saliency information can be converted directly to an estimated classification accuracy.

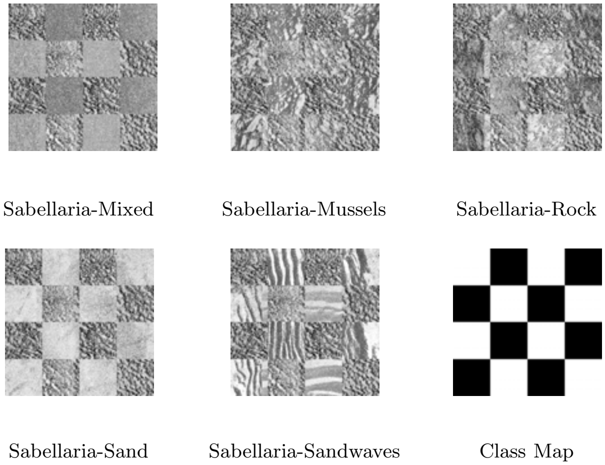
The paper is organised as follows: In section 2, image data and seabed classes are described. The conceptual framework for our ensemble is defined in section 3. GLCM parameterisation is outlined in section 4. A selection of results are presented and discussed in section 5. Finally, conclusions are drawn in section 6.

2 Experimental Data

Five test images (figure 1) with 5-bit radiometric range ($2^5 = 32$ quantised levels) and a resolution of 6 pixels/metre are synthesised by concatenating image regions sampled from multiple ground-truth locations in a sidescan mosaic. Corresponding seabed classes are listed in table 1, with specific, $c_s = \{c_1, c_2, ...c_n\}$ and generic, Sabellaria, S and not-Sabellaria, $\neg S$, class labels. A further five images are created by applying combined noise components; random, additive Gaussian noise $\mathcal{N} \sim (0,\ 0.1)$ and multiplicative speckle noise. Each image is scaled according to the kernel dimension, k, to ensure class homogeneity in the feature generation and down-sampling process.

Table 1. Ground-truthed textural classes used in this study (from [9])

Class	Description	Gen. class
c_1	Mixed sediments	$\neg S$
c_2	Mussels	$\neg S$
c_3	Rock	$\neg S$
c_4	Sabellaria	S
c_5	Sand	$\neg S$
c_6	Sandwaves	$\neg S$



Sabellaria-Mixed

Sabellaria-Mussels

Sabellaria-Rock

Sabellaria-Sand

Sabellaria-Sandwaves

Class Map

Fig. 1. Synthesised test image instances (no added noise, kernel dimension, $k = 11$). White squares in the class map correspond to the Sabellaria class.

3 Proposed Ensemble for Consensus Feature Ranking

The distance between the probability distributions of features generated on two different classes, provides an estimate of the binary classification error probability (see for instance, Webb [10]). Disjoint distributions are indicative of highly discriminative features, compared to those with class distributions which are closer together. Hence, the features can be ranked in terms of their relative saliency when the distances between the distributions for all features have been computed. However, as we mentioned earlier, there are numerous different distance measures and each may produce conflicting rankings, even though the underlying distributions are identical. It is not known which measures are appropriate for use with sonar data, so we propose a new approach, using an ensemble to compute a consensus ranking, by combining the output from individual distance measures. Six non-parametric probabilistic distance measures are considered in our experiments, although any types or numbers of individuals can easily be deployed in the ensemble. The conceptual framework set out in this section defines the most general case of consensus ranking with m independent distance measures.

Important stages in our process include; (1) Balanced sampling of the input space, (2) Scaling the individual feature values to [0, 1] across the classes and features (3) Estimating the normalised, non-parametric probability distributions for all features and classes (4) Computing all pair-wise distance measures for all distributions (5) Assigning an integral (or continuous) consensus rank to each feature using equally weighted rank scores from each distance measurement.

3.1 Non-parametric Distribution Estimation

For a set of n specific classes $c_s = \{c_1, c_2, ...c_n\}$ non-parametric class conditional probability distributions $\mathrm{P}(\mathbf{x} \mid c_s)$ for each component, x_f of the feature vector, $\mathbf{x} \in \mathbb{R}^D$ (where D is the dimension of the feature space) are estimated using histograms. The scaled feature domain, x_f = [0, 1] is divided into equal width intervals, i. Pair-wise comparisons for the individual features are computed if the distances between the generic S and $\neg S$ distributions are required. These comparisons can be made for every specific class in c_s. Probability distribution estimates for the generic classes are then, $p = p(x_f \mid S)$ and $q = q(x_f \mid \neg S)$ respectively. Once computed the set of $\frac{1}{2}nD(n-1)$ histograms is re-used by each measure in the ensemble.

3.2 Individual Distance Measures

Six distance measures are considered in this test case as summarised in table 2.

Table 2. Summary of the six distance measured used

Distance measure $\mathbf{d}_m$	Abbreviation	Range
Bhattacharyya distance measure [11]	Bh	[0, $\pi/2$]
Chi-Squared distance measure	CS	[0, 1]
Euclidean L_2	Eu	[0, 1]
Harmonic mean	HM	[0, 1]
Kullback Leibler Divergence [12]	KL	[0, 2ln(2)]
Tanimoto distance measure	Ta	[0, 1]

3.3 Consensus Ranking of Features

Each distance measure, $\mathbf{d}_m$, $m \in$ [1,2,...$\mid \mathbf{d}_m \mid$] generates a list of values dependent on the individual feature, x_f, $f \in$ [1,2,...D] and specific class distribution comparison, $(p,q)_w$, $w \in$ [1,2,...η], where m, f, $w \in \mathbb{Z}^+$ and say, $\eta = \mid S \mid \mid \neg S \mid$. Thus, for every pair-wise class combination we have $\mid \mathbf{d}_m \mid$ (in this case up to six) lists of computed distance values for each feature, given by,

$$\mathbf{d}_m(\mathbf{x}, (p,q)_w) = (d_1(x_1), d_2(x_2), ...d_D(x_D)) \tag{1}$$

The output of the ranking function, $\mathbf{r}$ is the ordered set of values for the particular distance measure, d and specific class distribution comparison, $(p,q)_w$,

$$\mathbf{r}(\mathbf{d}_m, \mathbf{x}, (p,q)_w) = (r_1(x_f), r_2(x_f), ...r_{max}(x_f)) \tag{2}$$

In the event of a tie for the highest ranked position, all tied positions can be allocated a rank score equal to the cardinality of the set of values. Each feature, x_f now has a rank score, $x_f(r_j)$ in $\eta \cdot \mid \mathbf{d}_m \mid$ lists. Further, by summing the rank scores for individual features over the distance measures and all pair-wise combinations of specific classes, an integral, mean-distance consensus rank value for each feature, $\mathbf{r}_{con}(x_f)$ in terms of its capacity to discriminate between the generic classes is obtained, from,

$$\mathbf{r}_{con}(x_f) = \left\lfloor \frac{1}{\mid \mathbf{d}m \mid} \sum_{m=1}^{\mid \mathbf{d}m \mid} \sum_{w=1}^{\eta} x_f(r_j)(\mathbf{d}m, (p,q)_w) \right\rceil \tag{3}$$

The formula we have derived represents a general case for any number of features, distance measures and pair-wise class comparisons. Of course, it is not essential to assign an integral rank score. Since, the distance measures are continuous valued functions over defined ranges, clearly, our method can easily be modified to generate a normalised mean distance measure, $\bar{\mathbf{d}}_m$,

$$\bar{\mathbf{d}}_m = \frac{1}{\mid \mathbf{d}_m \mid} \sum_{m=1}^{\mid \mathbf{d}_m \mid} \frac{\mathbf{d}_m}{max(f_{\mathbf{d}_m})} \tag{4}$$

where, $max(f_{\mathbf{d}_m})$ is the functional maxima of the specific distance measure. The output from equation 4 can be ordered, to produce a continuous valued ranking of the features.

4 Co-occurrence Matrix Parameterisation

Five features, F_j ($\mid F \mid = 5$) are derived from co-occurrence matrices; (1) Angular Inverse difference Moment (AIDM), (2) Angular Second Moment (ASM), (3) Contrast (CON), (4) Correlation (COR) and (5) Entropy (ENT), as defined in Haralick *et al* [13]. Ranges of the co-occurrence matrix parameters and the applied directional configurations are summarised in table 3.

Table 3. Summary of co-occurrence matrix parameters and directional configurations

Parameter	**Range**
Computational kernel dimension, k (pixels)	{5, 7, 9, 11, 17, 23}
Quantisation level, Q (bits)	{1, 2, 3, 4}
Sampling vector orientation, θ ($\mid \theta \mid = 4$, due to symmetry)	$\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}, \pi, \frac{5\pi}{4}, \frac{3\pi}{2}, \frac{7\pi}{4}\}$
Inter-pixel distance, $\lVert \mathrm{d} \rVert$	$f(k, S_p)$
Sampling pattern, $S_p(\theta)$	(square, octagonal, uniform)
Directional configuration F_*	**Dimension, D**
All directions (Rotationally Variant), F_{RV}	$\lvert\theta\rvert\lvert F\rvert$
Maximum directional response, $F_{max} = max\{F_j(\theta_0, ...\theta_n)\}$	$\lvert F\rvert$
Rotationally invariant (mean), $F_\mu = \frac{1}{\lvert\theta\rvert} \sum_{k=1}^{\lvert\theta\rvert} F_\theta$	$\lvert F\rvert$
Rotationally invariant (DFT), $F_{(DFT)} : F_{j(\alpha)} = \frac{1}{\lvert\theta\rvert} \sum_{\theta=0}^{\lvert\theta\rvert-1} F_j(\theta) exp\{-2\pi i \alpha \theta / \mid \theta \mid\}$	$(\lvert\theta\rvert - 2)\lvert F\rvert$

5 Application to Sonar Data: Results and Discussion

Due to space constraints, results exposition is limited to the somewhat arbitrary choice of parameter subspace; Q $= 3$, $k = 11$, $S_p = octagonal$ (rotationally invariant ||d||), ||d||$\in \{1, 3, 5, 7, 9\}$ and discrimination between Sabellaria and the classes listed in table 1.

5.1 Factors Influencing Distance Measure Disagreements

We begin by demonstrating the influence of directional configuration and the classification task on pairwise rank disagreement. In all results, the total disagreements are factored for D and expressed as a percentage.

It is clear (figure 2) that F_{max} and F_{μ} generate far fewer rank disagreements across all classes, with and without added noise, compared to F_{DFT} and F_{RV}. In most cases, adding noise to the image classes reduces the number of disagreements. An obvious exception is F_{DFT} where the number of disagreements on each task and the variability over all classes increases when noise is added.

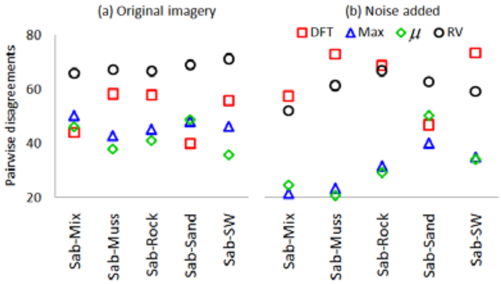


Fig. 2. Dependence of pairwise disagreements on directional configuration of the features, F_* and class discrimination task; (a) original imagery, (b) with added noise

The information in figure 2 does not tell us explicitly which directional configuration is preferable. The number of disagreements quantifies the certainty that different distance measures will agree on the feature rankings. If we were to evaluate and rank say, $\forall F_j \in F_{max}$ for a Sabellaria-Sandwaves discrimination task we would be more certain of obtaining consistent ranking results from different measures, compared to rankings of $\forall F_j \in F_{DFT}$ on the same task. Although as we shall see, consistency in ranking does not necessarily imply higher feature saliency or greater classification accuracy.

Trends in the disagreements are also evident for other parameters. Figure 3 shows the variability in rank disagreements between individual distance measures over F_{max}, as a function of the sampling vector magnitude, ||d|| on the Sabellaria-Sandwaves classes. More disagreements arise as the magnitude of the sampling vector decreases. Addition of noise reduces the disagreements in most cases (excluding ||d|| = 3). Eu and CS exhibit more disagreements and a greater variance in disagreements, compared to the other distance measures, probably due to the capture of diverse information from the feature distributions (clearly, if all measures captured the same information, they would all agree). This sensitivity of the measures to multiple parameters, coupled to the inherent spatial variability of visual concepts, noise and distortion in qualitative sonar imagery provides strong justification for fusing saliency information in an ensemble of distance measures.

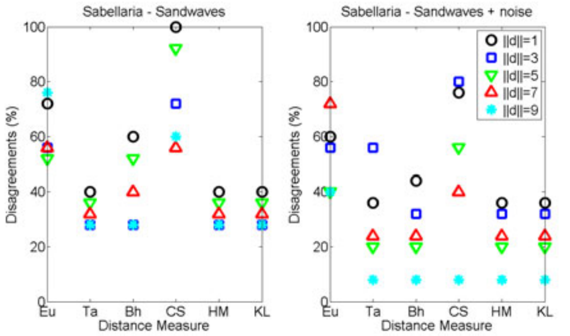


Fig. 3. Dependence of disagreements on the sampling vector magnitude ||d||

5.2 Using the Ensemble to Parameterise a Co-occurrence Matrix

Information from the ensemble can facilitate reliable decision making, concerning feature parameterisations and configurations. It can also provide an insight into the relative difficulty of the class discrimination tasks. For the parameter subspace defined earlier, figure 4(a) shows the mean ensemble saliency over $F_j \in F_{max}$ as a function of ||d|| and class discrimination task. As the magnitude of the sampling vector increases, with few exceptions, the mean saliency of the features decreases. The difference in saliency over the range of sampling vector magnitudes is greatest for the Sabellaria-Sandwaves classes, indicating a greater scale dependency, possibly due to strong textural anisotropies in the Sandwaves class. Based on the information in figure 4(a), we might expect the classification accuracy for all textural test cases to decrease as the scale of textural analysis increases. Further, the Sabellaria-Mixed sediment and Sabellaria-Sand discrimination tasks will be more accurate than for Sabellaria-Sandwaves. The lowest classification accuracies should occur on the Sabellaria-Rock and Sabellaria-Mussels tasks.

Correlation of Ensemble Output with Classification Accuracy. We now proceed to validate the reliability of the ensemble and establish the relationship between feature saliency estimated by the esemble and the classification accuracy. A linear-kernel Support Vector Machine (SVM) classifier [14] is used as the core learning algorithm in a wrapper. Multiple training sets are created from balanced, non-replacement, randomly sampled patterns. The mean and SD of the classification results are summarised in table 4 and mean results shown graphically in figure 4(b).

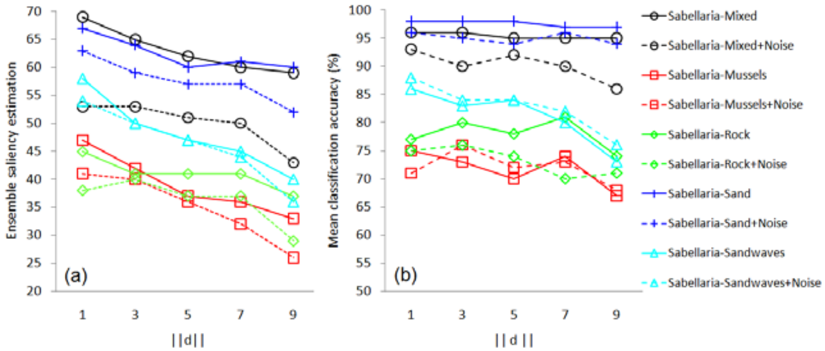


Fig. 4. (a) Mean ensemble estimated saliency as a function of ||d|| and class discrimination task. (b) Classification accuracy.

The trends in figures 4(a) and (b) are remarkably similar. Using least squares regression an empirical linear relationship for calibrating the ensemble saliency estimation, $\hat{E}$ against the SVM classifier accuracy, C_{acc}, is,

$$\hat{C}_{acc} = 0.87\hat{E} + 42 \tag{5}$$

So, presented with the task of separating Sabellaria from other textures, in this particular subspace; $\forall F_j \in F_{max}, Q = 3, k = 11, S_p = octagonal, ||\mathrm{d}|| = 1$, we can expect to discriminate from other textural types with the following estimated accuracies, $\hat{C}_{acc}(\mu)\%$: Mixed sediments(100), Sand(100), Sandwaves(92), Mussels(83), Rock(81). The estimated values from equation 5 are in very good agreement with (although 2 - 8% higher than) the SVM classification accuracies in table 4.

Table 4. Mean and standard deviation of SVM classification accuracies (%)

$F_{max}, Q = 3, k = 11, \|\|\mathrm{d}\|\| =$	1		3		5		7		9		μ_k	
Discrimination task	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ	μ	σ
Sabellaria-Mixed	96	1	96	1	95	2	95	3	95	2	95	2
Sabellaria-Mixed+Noise	93	2	90	3	92	3	90	2	86	6	90	3
Sabellaria-Mussels	75	17	73	14	70	11	74	7	67	14	72	13
Sabellaria-Mussels+Noise	71	17	76	7	72	13	73	6	68	9	72	10
Sabellaria-Rock	77	6	80	3	78	3	81	3	74	3	78	4
Sabellaria-Rock+Noise	75	5	76	4	74	8	70	10	71	5	73	6
Sabellaria-Sand	98	1	98	0	98	1	97	2	97	0	98	1
Sabellaria-Sand+Noise	96	1	95	2	94	4	96	1	94	3	95	2
Sabellaria-Sandwaves	86	10	83	7	84	6	80	5	73	4	81	6
Sabellaria-Sandwaves+Noise	88	5	84	9	84	6	82	6	76	3	83	6

The correlation between $\hat{E}$ and classification accuracy for all class discrimination tasks in the parameter subspace is shown by the scatter-plot in figure 5 (b). Error bars represent one standard deviation in the x and y directions.

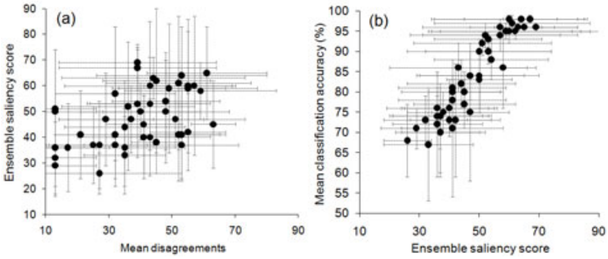


Fig. 5. (a) Correlation between mean disagreements and mean ensemble score, (b) correlation between mean ensemble score and mean classification accuracy

Figure 5 (a) shows a weaker linear correlation ($r = 0.47$) between the mean number of disagreements and $\hat{E}$. An interesting, albeit speculative implication is, the more useful the features, the less certain we can be that a single distance measure is capable of capturing the full spectrum of useful information about the features.

6 Conclusions and Scope for Future Work

A conceptual framework for an ensemble of probabilistic distance measures has been proposed. The idea is motivated by the fact that different distance measures disagree on feature rankings and each measure contains useful but diverse information about the distributions. Further, there is no single definitive distance measure for use with sonar imagery. The sample of results we have presented in this initial investigation, demonstrate the ensemble is capable of reliably evaluating features and estimating classification accuracies. Further work will involve analysing the results generated on the full parameter space, different feature types and on benchmark photographic textures as well as other sonar imagery.

References

1. Limpenny, D., Foster-Smith, R., Edwards, T., Hendrick, V., Diesing, M., Eggleton, J., Meadows, W., Crutchfield, Z., Pfeifer, S., Reach, I.: Best methods for identifying and evaluating Sabellaria spinulosa and cobble reef. Aggregate Levy Sustainability Fund Project MAL0008
2. Samiee, K., Rad, G.: Textural Segmentation of Sidescan Sonar Images Based on Gabor Filters Bank and Active Contours without Edges. In: IEEE Fifth International Conference on Advanced Video and Signal Based Surveillance. AVSS 2008, pp. 3–8 (2008)

3. Wang, Y., Liu, Z., Sang, E., Ma, H.: Sonar image classification based on directional wavelet and fuzzy fractal dimension. In: 2nd IEEE Conference on Industrial Electronics and Applications, ICIEA 2007, pp. 118–120. IEEE, Los Alamitos (2007)
4. Karoui, I., Fablet, R., Boucher, J., Pieczynski, W., Augustin, J.: Fusion of textural statistics using a similarity measure: application to texture recognition and segmentation. Pattern Anal. Appl. 11(3), 425–434 (2008)
5. Reed, T., Hussong, D.: Digital image processing techniques for enhancement and classification of SeaMARC II side scan sonar imagery. J. Geophys. Res. 94(B6), 7469–7490 (1989)
6. Blondel, P., Gómez Sichi, O.: Textural analyses of multibeam sonar imagery from Stanton Banks, Northern Ireland continental shelf. Appl. Acoust. 70(10), 1288–1297 (2009)
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The J. Mach. Learn. Res. 3, 1157–1182 (2003)
8. Cha, S.: Comprehensive survey on distance/similarity measures between probability density functions. Int. J. Math. Model. Method. Appl. Sci. 4(1), 300–307 (2007)
9. Pearce, B.: Thanet Offshore Wind Farm Benthic and Intertidal Resource Survey, Section III - Sabellaria Distribution (2005)
10. Webb, A.: Statistical pattern recognition. John Wiley and Sons Ltd., Chichester (1999)
11. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by their probability distributions. Bull. Calcutta Math. Soc. 35(99-109), 4 (1943)
12. Petrou, M., Sevilla, P.: Image processing: dealing with texture. John Wiley & Sons Inc., Chichester (2006)
13. Haralick, R., Shanmugam, K., Dinstein, I.: Textural features for image classification. IEEE T. Syst. Man Cy. 3(6), 610–621 (1973)
14. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1998)

Engineering Spatial Analysis in Real Estate Applications

Michele Argiolas[1] and Nicoletta Dessì[2]

[1] Università degli Studi di Cagliari, Dipartimento di Ingegneria del Territorio,
Via Marengo 3, 09123 Cagliari, Italy
[2] Università degli Studi di Cagliari, Dipartimento di Matematica e Informatica,
Via Ospedale 72, 09124 Cagliari, Italy
{michele.argiolas,dessi}@unica.it

Abstract. This paper considers the urban processes that real estate (RE) experts use in assessing the value of a certain property, based on characteristics of that property and its environment. The main objective is to illustrate the confluence of RE decisional processes and spatial analysis and to show how these techniques can be put to work together. This paper describes a software package specifically designed for supporting spatial analysis of urban data collections. This software can serve as a reference architecture for developing applications that support decisional processes in real estate. Based on geographical features, the computational environment supports appraisal of a wide range of real estate types and can also create analytical maps for use in developing plans and strategies. A case study demonstrates how the computational environment can improve the quality of the diagnosis of urban real estate in a region that has been selected for a prototype implementation.

Keywords: Data Engineering and Applications, Real Estate, Spatial Analysis.

1 Introduction

Originally based on the development of statistical methods for geography, spatial analysis can be defined as a technology that describes the spatial relationships and the spatial actions of an object in the geographical space [1]. Spatial analysis is extremely relevant to Geographic Information Systems (GIS) as a support technology providing a link between the cartographic domain and the key areas of statistical analysis and modeling [2]. This technology can be applied in three markedly different contexts: a) testing "a priori" hypotheses about relevant patterns in spatial data, b) describing spatial patterns and relationships and c) supporting decisions about spatial planning [2].

This paper considers the last context and leverages a substantial body of research [3-7] that explores the role and the potential of spatial analysis to fruitfully assist urban analysts and modelers. The motivation is that many urban processes are spatially conditioned, i.e. they are processes in which events at one location are partially affected by events at other locations. Specifically, this paper considers urban processes supporting real estate (RE) experts in assessing property value based on its characteristics. Some of these characteristics are intrinsic to the property, others are environmental or economic factors, such as the real estate market and the general

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 337–344, 2011.

trend of the local economy. In analyzing RE processes we study an individual city, conceptualized as a collection of various interrelated components. RE evaluation reflects economic and technical factors and is a process concerning different aspects of decision making such as urban planning, evaluating risk with investing, assessing market values etc. The main objective of this paper is to document the confluence between RE decisional processes and spatial analysis and to show how these techniques can be put to work together.

From a technical point a view, spatial analysis of a real estate domain can be carried out with off-the-shelf GIS software. However, some GIS software packages are too expensive for an expert and, more importantly, do not provide specific options for rapid and simple analysis of urban data. Within the real estate domain an additional problem is that it generates unmanageably huge datasets with diverse internal structures and no principles for integrating information. For example, data stored by municipalities becomes ineffective when each municipal unit begins to establish separate information systems, resulting in legal, administrative and economic problems. Real estate evaluators are aware of these problems and seek technical methods to solve them. Technical solutions must avoid significant re-engineering of existing datasets and applications. In this respect, the design of applications supporting real estate spatial analysis calls for a deep re-thinking of current design approaches that often result in monolithic systems.

This paper describes a software package specifically designed for supporting spatial analysis of urban data collections. It aims to be a reference architecture for developing applications that support real estate appraisers. By offering geographical references, the proposed software supports the appraisers in planning, designing and coordinating various real estate evaluation scenarios. It has been designed and tested using open source components. Base-map data are provided via the Internet and can be embedded in the local software. The software can import data about the characteristics of an urban land as well, and can automatically check the accuracy of this data. The computational environment shows maps of the locations where data was collected and, more interestingly, it can also create analytical maps indicating the number of observations, the number of distinct classes of observations and the values of diversity indices for real estate evaluations. For example, a map of land parcels describes the spatial position of the parcels with points placed where buildings or public offices are located.

The paper is organized as follows. Section 2 gives a short overview of the decision making issues in the real estate domain. Section 3 presents the proposed software, whose effectiveness is tested by a case study that is exposed in the section 4. Finally, section 5 presents conclusions.

2 Background

As each parcel of land has unique characteristics, specialists are often called on to valuate real estate and facilitate transactions in purchasing. There are multiple usage scenarios. Let us start with a simple example: the evaluation of a house for living or a place for building a hotel. A closer look at how such problems are usually solved indicates that the value of a property relies on two basic variables: the overall quality and the price of property. Assessing the overall quality is concerned with interpreting complex, heterogeneous information that is usually hard to express numerically. This requires information to be evaluated by a human analyst on the basis of his/her domain expertise and knowledge of the study area.

A large number of models have been built for selecting sites for storage, retail shops, parks, fire stations, hospitals and so forth, where the location choice is not only a spatial problem but also involves economical factors [8-10]. For example, in locating public facilities (e.g., police stations or hospitals) one has to consider how to offer acceptable services to residents with the minimal cost of locating the facilities, while in locating retail outlets one should focus on how to select a site with the maximal number of potential shoppers [11].

With the emergence of GIS, both spatial and non-spatial data can be handled simultaneously by thematic maps with a variety of demographic information relating to population, housing and economic activities. These maps help to solve location problems by visualizing both data and geographic information. Based on digitized remote sensing data, spatial statistics present within a GIS have made possible the development of accurate, consistent, and unbiased explanatory variables in a fast and efficient manner. These variables can then be used as indicators for measuring the environmental characteristics of properties and increasing the understanding of house pricing variations.

3 The Reference Architecture

A widely used approach for creating flexible and reusable applications is to structure the related modules into a three-layer architecture, with the user interface or browser as the first layer. The application tools, at the second layer, are the integration point, providing access to a wide range of capabilities. The third layer, the data layer, includes databases, computer and file resources. Breaking up applications into layers makes it possible to modify or add one specific layer, rather than having to rewrite the entire application over.

Here, we propose a flexible extension of the above mentioned architecture by considering mash-up applications. Generally speaking, a mash-up is a process that integrates data/content from different resources on the Internet in order to provide the user with a flexible and easy-of-use way for service composition on the web. Usually hosted on a Web server, a mash-up application is rendered in a Web browser, through which user interaction takes place. In this way, mash-ups combine third-party data and content from more than one source to create a completely new application. This process is made possible by providers that promote free tools on the Internet and release their API for free. This allows users to develop applications that integrate provider tools with specific user data. Google Maps and Google Earth are well known examples of this kind of tools.

We introduce the mash-up as a middleware layer, i.e. as an additional layer that interfaces both the application layer and the data layer. In more detail, the user interface is supported by a Web Browser capable of assembling and composing mashed-up content. Results can be shown graphically in separate layers and the user can select the best one from the pool of presented layers. The user is able to iteratively refine and modify the answer to a graphical query as well as formulate queries about the source of data. Using map browser tools, such as zoom-in and zoom-out, users can perform these tasks interactively. Most importantly, users can define new tasks properly and determine carefully what criteria should be employed in the evaluation and comparison of alternative locations.

The application layer contains the core business logic processing, i.e. the applications that can be invoked by the decision maker. It incorporates calculation models, statistical packages and simulation tools. Each application contributes a distinct activity for addressing the problems encountered in the various steps of the decision process.

The middleware level is enabled by mash-ups that we implemented based on open APIs. There are two styles of mash-ups: web-based and server-based. Web-based mash-ups typically use the user's web browser for combining and reformatting the data. Server-based mash-ups analyze and reformat the data on a remote server and transmit the data to the user's browser in its final form. The data communication format is XML. The middleware level supports data management by means of:

- customer mash-ups for personalization of data and viewing, allowing decision makers to combine and reformat the data according to their needs;
- data mash-ups allow to combine information from multiple sources into a single representation;
- business mash-ups, focusing on a single presentation of data and allowing users to access application packages on Internet.

Data are stored in a geo-coded database implemented in a DBMS (MySQL) that extends standard SQL with geometric attributes as defined by Open Geospatial Consortium. MySQL was chosen as it is stand-alone software and is freely available, thus serving as a good starting point for people who work on real estate environments but do not have access to commercial GIS software. However, those who do have access to commercial GIS software may still be interested in our tool, as it addresses specific real estate problems for which functionality is not available or difficult to use in existing GIS packages.

The computational environment we implemented provides access to Google Maps, Google Earth and EBay. We are also planning to extend the system by allowing it to access public data that comes from Craigslist and Trulia. We chose Google, since it is free technology that provides high-resolution satellite images for most urban areas and supports spatial data infrastructures. Specifically, the computational environment incorporates the two-dimensional features from Google Maps and the three-dimensional features from Google Earth.

To answer user queries, the computational environment gets the data from proper layers. Results are shown on a website by a mash-up application that was designed with embedded Google Maps and Google Earth APIs on it. In turn, when the user selects a geographical location, the computational environment retrieves related data from the database.

4 A Case Study

This section describes a case study that explores the potential of the proposed software. The city of Cagliari (Italy) was selected as a case study area. Collected data was subdivided into two categories:

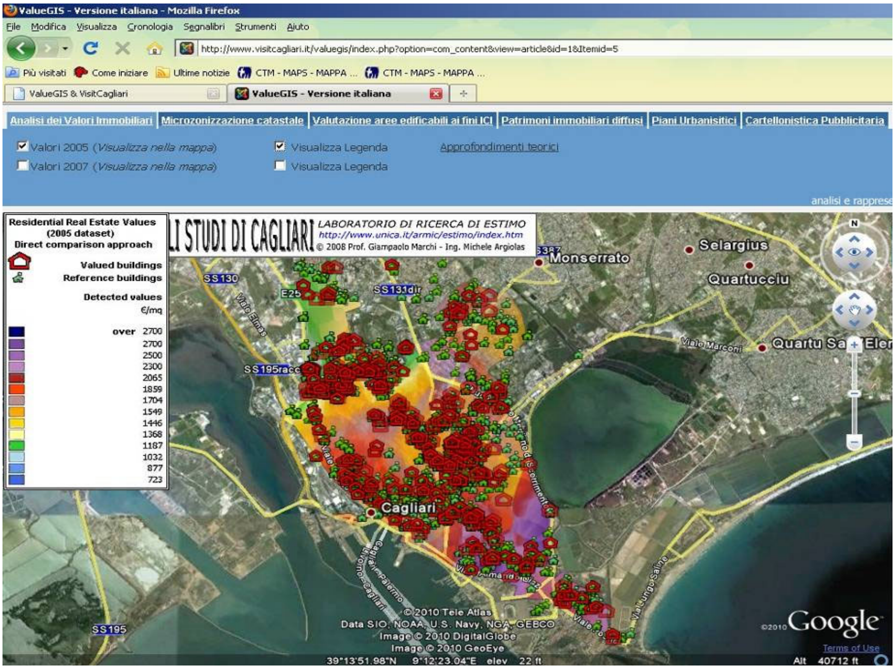
- marketing and economical data, accounting for temporal and economical aspects that are important for the evaluation of profit and risks of investments;
- urban quality data, considering presence of special sites and infrastructure such as parks, shops, railway stations and so on.

Data include qualitative judgments and values, as supplied by different real estate brokers, and data about buildings in the examined area that are offered for sale at Ebay. Additional data can be collected from other networked resources, including municipalities and private realtors.

Once the database has been geo-referenced, we can carry out various spatial analyses. Most of these are based on dividing the space into equal-sized cells, the size of which can be changed by the user. Compared to the use of geographical or administrative regions, these cells have the advantage of being all the same size, allowing them to be compared more objectively. In the following, we describe two scenarios of interest in RE evaluation and we show how these scenarios are supported by the environment we implemented.

Evaluating the Real Estate Market. The state and the evolution of the real estate market are shaped by various factors such as the urban quality, the market values or the number of properties available for sale. A computational environment that supports the spatial interaction between all these factors is a very useful instrument, not only for professionals (i.e. people involved in building appraisal, urban planning, RE taxation etc.) but also for citizens. The median sale price of a building is one of the most common measurements used to compare real estate prices in different markets, areas, and periods. This statistic can be very useful, but also misunderstood or even misleading. To obtain a better understanding, a good cross-reference is to look at the location of homes that have been used to generate the median sale price.

As a case study, we consider the medium sale values in 2009 in Cagliari and we apply some special functions we implemented to support exploratory spatial analysis. The first function is the zone-design procedure that calculates the optimal number of distinct classes of the variable "sale price". Fig. 1 shows the chromatic layer that visualizes these classes (i.e. the best way for clustering buildings according to their prices). A second specialized function we apply allows to determine the number of observations (i.e. buildings) in each class. As Fig. 1 shows, the map represents each single building by pinpointing them with a pushpin. If a user clicks on one of these elements, the system displays information related to the property as stored in the DBMS (Fig. 2).

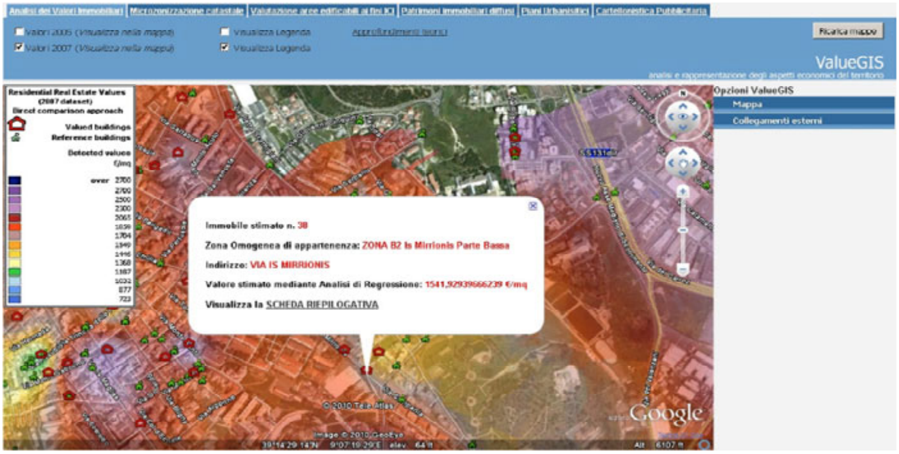

ValueGIS - Versione italiana - Mozilla Firefox
File Modifica Visualizza Cronologia Segnalibri Strumenti Aiuto
http://www.visitcagliari.it/valuegis/index.php?option=com_content&view=article&id=1&Itemid=5
Più visitati Come iniziare Ultime notizie CTM - MAPS - MAPPA ... CTM - MAPS - MAPPA ...
ValueGIS & VisitCagliari
ValueGIS - Versione italiana
Analisi dei Valori Immobiliari | Microzonizzazione catastale | Valutazione aree edificabili ai fini ICI | Patrimoni immobiliari diffusi | Piani Urbanisitici | Cartellonistica Pubblicitaria
Approfondimenti teorici
analisi e rappres
LI STUDI DI CAGLIARI
LABORATORIO DI RICERCA DI ESTIMO
http://www.unica.it/armic/estimo/index.htm
© 2008 Prof. Giampaolo Marchi - Ing. Michele Argiolas
Monserrato
Selargius
Quartucciu
Quartu Sa
Viale Marconi
SS130
SS131dir
E25
SS195racc
Viale Elmas
Cagliari
SS195
Viale
©2010 Tele Atlas
Data SIO, NOAA, U.S. Navy, NGA, GEBCO
Image © 2010 DigitalGlobe
Image © 2010 GeoEye
39°13'51.98"N 9°12'23.04"E elev 22 ft
©2010 Google
Terms of Use
Alt 40712 ft

☑ Valori 2005 (*Visualizza nella mappa*) ☑ Visualizza Legenda
□ Valori 2007 (*Visualizza nella mappa*) □ Visualizza Legenda

Residential Real Estate Values (2005 dataset)
Direct comparison approach

🏠	Valued buildings
🏠	Reference buildings
	Detected values €/mq
■	over 2700
■	2700
■	2500
■	2300
■	2065
■	1859
■	1704
■	1549
■	1446
■	1368
■	1187
■	1032
■	877
■	723

Fig. 1. The thematic map of residential buildings in Cagliari


Analisi dei Valori Immobiliari | Microzonizzazione catastale | Valutazione aree edificabili ai fini ICI | Patrimoni immobiliari diffusi | Piani Urbanisitici | Cartellonistica Pubblicitaria
Approfondimenti teorici
Ricarica mappe
ValueGIS
analisi e rappresentazione degli aspetti economici del territorio
Opzioni ValueGIS
Mappa
Collegamenti esterni
SS131dir
© 2010 Tele Atlas
Image © 2010 GeoEye
Google

□ Valori 2005 (*Visualizza nella mappa*) □ Visualizza Legenda
☑ Valori 2007 (*Visualizza nella mappa*) ☑ Visualizza Legenda

Residential Real Estate Values (2007 dataset)
Direct comparison approach

Valued buildings
Reference buildings
Detected values €/mq
over 2700, 2700, 2500, 2300, 2065, 1859, 1704, 1549, 1446, 1368, 1187, 1032, 877, 723

Immobile stimato n. 38

Zona Omogenea di appartenenza: ZONA B2 Is Mirrionis Parte Bassa

Indirizzo: VIA IS MIRRIONIS

Valore stimato mediante Analisi di Regressione: 1541,92939666239 €/mq

Visualizza la SCHEDA RIEPILOGATIVA

Fig. 2 . Detailing building information

Locating Historical and Natural Preserved Sites. Urban conservation seeks to preserve, conserve and protect buildings and landscapes of historical significance. In Italy, the presence of historical and natural preserved sites can strongly influence RE values. The possibility to inspect such sites can significantly influence the appraisal especially in towns attracting large tourist flows. Fig. 3 shows the map of the

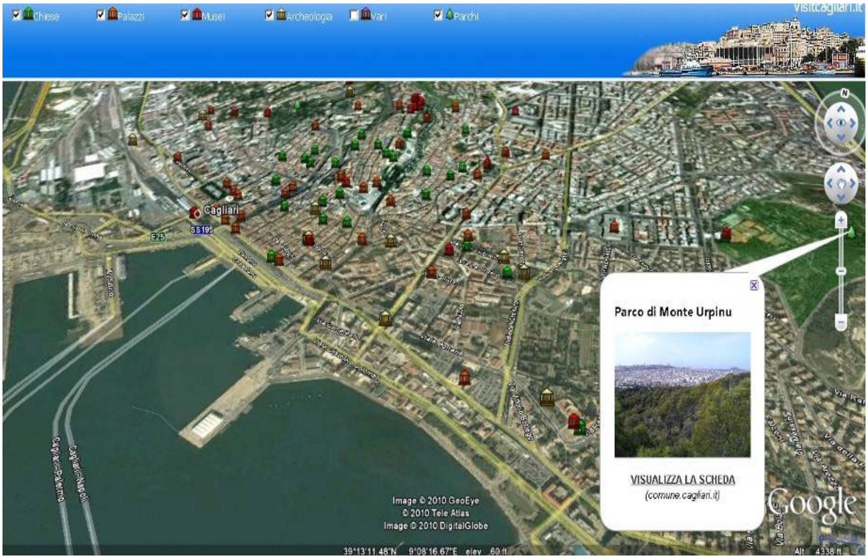


Fig. 3. The thematic map of historical sites in Cagliari

historical and natural preserved sites in Cagliari: with a click on the pushpin, an RE appraiser can access the description of the historical or natural preserved site and learn the medium price of nearby houses in order to evaluate the related influence on market values.

5 Conclusions

The presented software is likely to be most useful for the analysis of data covering very large areas. The problem of mining the emerging large amount of heterogeneous geographically referenced data is not addressed by conventional spatial analysis methods. In this paper, the challenge is viewed as the need of engineering the spatial analysis in order to change the "modus operandi" in real estate domain which is rich in data but poor in models. While some may argue that maps provide a very poor form of technology for supporting spatial analysis, we believe the problem lies not in the definition of what spatial analysis means, but in identifying the nature of the technology needed to provide the basic functionality relevant to the spatial analysis in real estate domain. We set out to build a software package based on these principles and to assemble a set of spatial functions which offer a prospect of new insights.

References

1. Xiang, N., Han, X.: Definition and contents of spatial analysis. Journal of Central South University of Technology 4(1), 23–31 (1997)
2. Openshaw, S.: Developing appropriate spatial analysis methods for GIS. In: Maguire, D., Goodchild, M., Rhind, D. (eds.) Geographical Information Systems, Principles and Applications, pp. 389–402. Longman, London (1991)

3. Paez, D., Scott, D.M.: Spatial statistics for urban analysis: a review of techniques with examples. GeoJournal 61, 53–57 (2004)
4. Landis, J., Zhang, M.: Using GIS to improve urban activity and forecasting models: three examples. In: Fotheringham, A.S., Wegener, M. (eds.) Spatial Models and GIS: New Potential and New Models, pp. 63–81. Taylor and Francis, London (2000)
5. Marchi, G., Argiolas, M.: A GIS based technology for representing and analyzing real estate values. In: Urban and Regional Data Management, UDMS Annual 2007, pp. 345–354. Taylor & Francis Group, London (2008)
6. Okunuki, K.: Urban analysis with GIS. GeoJournal 52, 181–188 (2001)
7. Sun, L., Zhu, H.: GIS-Based spatial decision support system for real estate appraisal. In: Fourth International Conference on Computer Sciences and Convergence Information Technology. IEEE, Los Alamitos (2009)
8. Abbasi, G.Y.: A decision support system for bank location selection. International Journal of Computer Applications in Technology 16, 202–210 (2003)
9. Arampatzis, G., Kiranoudis, C.T., Scaloubacas, P., Assimacopoulos, D.: A GIS-based decision support system for planning urban transportation policies. European Journal of Operational Research 152, 465–475 (2004)
10. Bechmann, M.J.: Lectures on location theory. Springer, New York (1999)
11. Cheng, E.W.L., Li, H.: Exploring quantitative methods for project location selection. Building and Environment 39, 1467–1476 (2004)

Infeasibility Driven Evolutionary Algorithm with ARIMA-Based Prediction Mechanism

Patryk Filipiak[1], Krzysztof Michalak[2], and Piotr Lipinski[1]

[1] Institute of Computer Science,
University of Wroclaw, Wroclaw, Poland
{patryk.filipiak,piotr.lipinski}@ii.uni.wroc.pl
[2] Institute of Business Informatics
Wroclaw University of Economics, Wroclaw, Poland
krzysztof.michalak@ue.wroc.pl

Abstract. This paper proposes an improvement of evolutionary algorithms for dynamic objective functions with a prediction mechanism based on the Autoregressive Integrated Moving Average (ARIMA) model. It extends the Infeasibility Driven Evolutionary Algorithm (IDEA) that maintains a population of feasible and infeasible solutions in order to react on changing objectives faster. Combining IDEA with ARIMA leads to a more efficient evolutionary algorithm that reacts faster to the changing objectives which profits from using information coming from the prediction mechanism and remains one time instant ahead of the original algorithm. Preliminary experiments performed on popular benchmark problems confirm that the IDEA-ARIMA outperforms the original IDEA algorithm in many cases.

1 Introduction

Dynamic single and multi objective optimization is a class of optimization problems, where the objective function as well as constraints change with time during computations [2]. Classic evolutionary algorithms are usually inefficient in such optimization problems, because they attempt to lead the population to the most promising region of the search space and focus on more and more close regions, which does not allow for a reaction to the changing objectives.

However, a number of modern evolutionary algorithms were developed for dynamic objective functions, [3], [5], [8]. Some of them, such as Infeasibility Driven Evolutionary Algorithm (IDEA) [8], maintains a population of feasible and infeasible solutions, which allows for a faster reaction to the changing objectives, when some solutions from previous populations, infeasible in previous time instants, become feasible in the future and outperforms the best solutions found in previous iterations. Some approaches also aim at combining standard evolutionary search with some prediction mechanisms [6], [9].

In this paper, we focus on the Infeasibility Driven Evolutionary Algorithm and combining it with an ARIMA-based prediction mechanism to increase its ability to react on changing objectives.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 345–352, 2011.

This paper is structured in the following manner: Section 2 defines the optimization problem with dynamic objective functions. Section 3 presents the evolutionary algorithm with a prediction mechanism. Section 4 reports on a number of experiments concerning popular benchmark problems. Finally, Section 5 concludes the paper.

2 Optimization Problem with Dynamic Objective Function

This paper refers to optimization problems with objective functions $F^{(\alpha)} : \mathbb{R}^d \rightarrow \mathbb{R}$ in a parametric form with parameters α, where such parameters change with time. Therefore, the objective function also changes with time and may be denoted as a dynamic objective function $F^{(t)} = F^{(\alpha_t)}$, where α_t stands for the values of the parameters at time t. Similarly, constraint functions $G_i^{(\alpha)} : \mathbb{R}^d \rightarrow \mathbb{R}$, $i = 1, 2, \ldots, m$, lead to dynamic constraint functions $G_i^{(t)} = G_i^{(\alpha_t)}$.

Therefore, the optimization problem with dynamic objective function considered in this paper is to find $x^{(t)} \in \mathbb{R}^d$ such that

$$x^{(t)} = \text{arg min}\{F^{(t)}(x) : x \in \mathbb{R}^d \ \wedge \ G_i^{(t)}(x) \geq 0, \text{for } i = 1, 2, \ldots, m\}, \qquad (1)$$

for all t over a specific time period.

Such an optimization problem occurs in many practical applications, such as time series analysis, where the parameters α come from historical time series instances and change each time a new instance is recorded. It is worth noting that in such an approach, at each time t_0, although the future objective functions $F^{(t)}$ for $t > t_0$ are unknown and cannot be computed, the past objective functions $F^{(t)}$ for $t < t_0$ can be computed, because the past parameters α_t are known.

3 Evolutionary Algorithm with a Prediction Mechanism

In this paper ARIMA prediction is used to improve performance of IDEA. Algorithm 1 presents a general overview of the IDEA-ARIMA approach.

3.1 Evolutionary Algorithm

The Infeasibility Driven Evolutionary Algorithm used in this paper is described in [8]. In this algorithm, a population consists in part of individuals that violate constraints. Even though these individuals do not represent valid solutions of the problem they help to find feasible solutions.

In order to detect changes in a dynamic multiobjective function F and in the constraints, an evaluation at random points is performed before each iteration of the evolutionary algorithm. If a change is detected, i.e. the value of the objective or a constraint function differs from the previously recorded value for at least one point, the current population P_{i-1} is re-evaluated according to the new conditions.

In the IDEA algorithm a sub-evolution (sub-EA) procedure is used which is invoked with an argument P_{i-1} as an initial population. After N'_{gen} iterations of crossover and mutation, the next population P_i is generated by unioning the resulting population of sub-evolution C_{i-1} with P_{i-1} and reducing the result to N individuals with the highest fitness value including a fixed amount of infeasible solutions.

3.2 ARIMA Prediction

In order to improve the robustness of the above algorithm, IDEA-ARIMA uses a prediction mechanism ARIMA(p, d, q) [1] to forecast changes of the object function.

For each generation $i = 1, \ldots, N_{Gen}$ of IDEA-ARIMA, S_i is a set of all previously obtained solutions each of which is assigned a time series $\mathbf{x}_t$ containing all values of the object function evaluated until now. Anytime a change of function F is detected, as described further, causing the current population to be re-evaluated, new values of the objective function are being calculated for all individuals in S_i and stored in corresponding time series. Similarly, in the sub-evolution procedure an S'_j set holds the union of all populations $P'_1, \ldots, P'_{N'_{Gen}}$ for all evolutionary steps $j = 1, \ldots, N'_{Gen}$. In the end, former values of the object function for solutions in S'_j are calculated and stored as a time series.

After $H_{Start} - 1$ iterations of regular IDEA algorithm in which historical data is collected, ARIMA prediction begins. In generations $H_{Start}, \ldots, N_{Gen}$ additional hidden population H_i is initiated randomly and evaluated according to predicted values of the object function. In order to compute the forecast for an arbitrary individual $p \in P_i$ a fixed amount of its closest (e.g. in Euclidean norm) neighbours $s_1, \ldots, s_k \in S_i$ is selected and then the weighted sum of predicted values of $F(s_1), \ldots, F(s_k)$ in the next time step is calculated as a result. In the case of the hidden population, no record of former individuals is stored anywhere and the evaluation step is based on the predicted (not actual) values of the object function.

Algorithm 1. Evolutionary Framework for Dynamic Optimization with ARIMA-prediction where N = population size, N_{Gen} = number of generations and H_{Start} is the number of iteration at which the ARIMA-prediction begins.

```
$S_1 = \emptyset$
$P_1 =$ InitPopulation()
Evaluate($P_1$)
for $i = 2 \to N_{Gen}$ do
    if the function $F$ has changed then
        Evaluate($P_{i-1}$)
        StoreEvaluation($S_{i-1}, i$)
        if $i - 1 \geq H_{Start}$ then
            Evaluate($H_i$)
            $P_{i-1} =$ Reduce($P_{i-1} + H_i$)
        end if
    end if
    Sub-evolve population $P_i$ (using actual object function) producing $P_{i+1}$
    if $i \geq H_{Start}$ then
        $H_i =$ InitPopulation()
        Sub-evolve hidden population $H_i$ (using ARIMA prediction on $S_{i-1}$) producing $H_{i+1}$
    end if
end for
```

4 Experiments

In order to validate the proposed algorithm, a number of experiments were performed on some popular benchmark problems:

Benchmark problem g24_1 [7] contains the following object function:

$$F^{(t)}(x) = -\left[\sin\left(k\pi t + \frac{\pi}{2}\right) \cdot x_1 + x_2\right], \quad x = (x_1, x_2) \in [0,3] \times [0,4] \tag{2}$$

minimized subject to the following constraints:

$$G_1(x) = 2x_1^4 - 8x_1^3 + 8x_1^2 - x_2 + 2 \geq 0, \tag{3}$$

$$G_2(x) = 4x_1^4 - 32x_1^3 + 88x_1^2 - 96x_1 - x_2 + 36 \geq 0. \tag{4}$$

Benchmark problem g24_2 [7] contains the following object function:

$$F^{(t)}(x) = -\left[p_1(t) \cdot x_1 + p_2(t) \cdot x_2\right], \quad x = (x_1, x_2) \in [0,3] \times [0,4] \tag{5}$$

$$p_1(t) = \begin{cases} \sin\left(\frac{k\pi t}{2} + \frac{\pi}{2}\right), & t \mid 2 \\ p_1(t-1), & t \nmid 2 \end{cases} \tag{6}$$

$$p_2(t) = \begin{cases} p_2(\max\{0, t-1\}), & t \mid 2 \\ \sin\left(\frac{k\pi(t-1)}{2} + \frac{\pi}{2}\right), & t \nmid 2 \end{cases} \tag{7}$$

minimized subject to the following constraints:

$$G_1(x) = 2x_1^4 - 8x_1^3 + 8x_1^2 - x_2 + 2 \geq 0, \tag{8}$$

$$G_2(x) = 4x_1^4 - 32x_1^3 + 88x_1^2 - 96x_1 - x_2 + 36 \geq 0. \tag{9}$$

In both benchmark problems defined above, severity regulator $k \in [0,2]$ was set to 0.25 which implies periodic repetitions of the object function formula every 16 iterations (i.e. $F^{(t)} \equiv F^{(t+16)}$).

Modified benchmark problem FDA1[4] contains the underlying object function:

$$F^{(t)}(x) = 1 - \sqrt{\frac{x_1}{1 + \sum_{i=2}^{n}\left(x_i - \sin\left(\frac{\pi t}{4}\right)\right)^2}},$$

$$x = (x_1, \ldots, x_n) \in [0,1] \times [-1,1]^{n-1} \tag{10}$$

minimized in the three variants:

(a) under no constraints,
(b) under static constraints defined as follows:

$$G_i(x) = \frac{3[x_2 - \frac{1}{2}(\alpha_i + \beta_i)]^2}{2(\alpha_i - \beta_i)^2} - x_1 + \frac{1}{4} \geq 0, \tag{11}$$

$$\alpha_i = \sin\left(\frac{\pi(i+1)}{4}\right), \quad \beta_i = \sin\left(\frac{\pi(i+1)}{4}\right), \quad i \in \{1,..,4\}. \tag{12}$$

(c) under dynamic constraints altering at each time t step as follows:

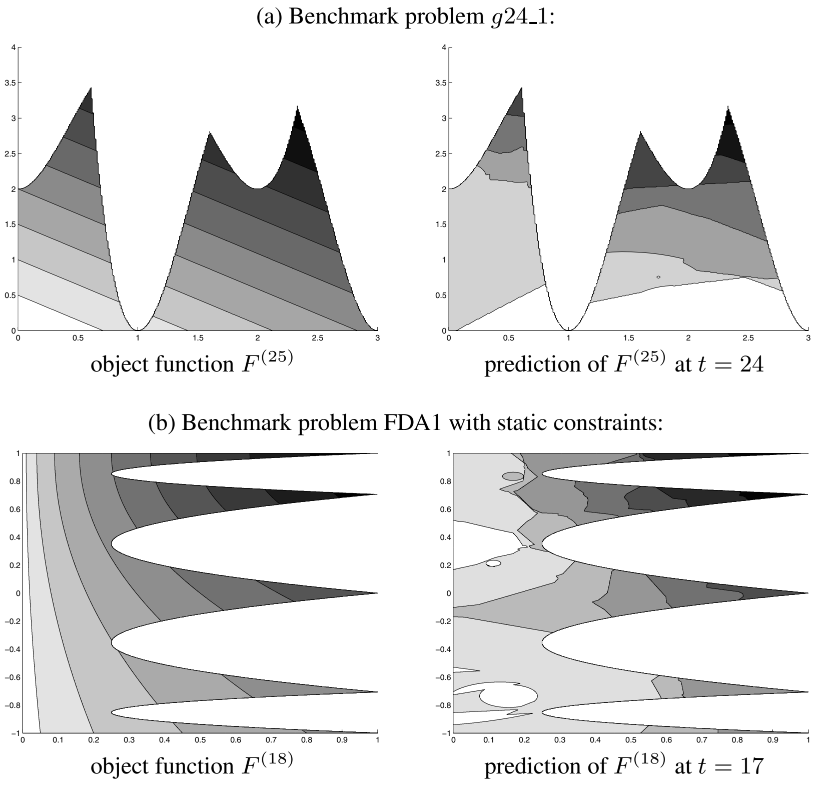
$$G_1^{(t)}(x) = \frac{3\left(x_2 - 1\right)^2}{4\left[1 - \sin\left(\frac{\pi t}{4}\right)\right]^2} - x_1 + \frac{1}{4} \geq 0, \tag{13}$$

$$G_2^{(t)}(x) = \frac{3\left(x_2 + 1\right)^2}{4\left[1 + \sin\left(\frac{\pi t}{4}\right)\right]^2} - x_1 + \frac{1}{4} \geq 0. \tag{14}$$

Note that due to the periodic characteristic of the above benchmark problem, for each time step t hold $F^{(t)} \equiv F^{(t+8)}$ and $G_i^{(t)} \equiv G_i^{(t+8)}$ $(i = 1, 2)$. Moreover, for all $t, n > 1$:

$$\forall_{x \in [0,1] \times [-1,1]^{n-1}} \quad 0 \leq F^{(t)}(x) \leq 1 \quad \text{and} \tag{15}$$

$$\exists_{x_0 \in [0,1] \times [-1,1]^{n-1}} \quad F^{(t)}(x_0) = 0 \;\wedge\; G_i^{(t)}(x_0) \geq 0 \quad \text{for } i = 1, 2. \tag{16}$$


(a) Benchmark problem $g24\_1$:

object function $F^{(25)}$

prediction of $F^{(25)}$ at $t = 24$

(b) Benchmark problem FDA1 with static constraints:

object function $F^{(18)}$

prediction of $F^{(18)}$ at $t = 17$

Fig. 1. 2D-plots of the object functions (left) and its predictive counterparts (right) at sample time steps in benchmark problems: (a) $g24\_1$ and (b) modified FDA1 with static constraints. Darker shades correspond to lower values of $F^{(t)}$

Parameters of ARIMA prediction model were set to (1, 0, 1) in all experiments. At each time step $t \geq H_{Start}$ a predicted value of the object function $\widehat{F}^{(t+1)}(x)$, $x \in \mathbb{R}^d$ was approximated with a weighted sum of $\widehat{F}^{(t+1)}(\cdot)$ calculated for the three closest neighbours of x stored in S_t, namely s_1, s_2, s_3, with weights $\|x - s_j\|^{-1} / \sum_{k=1}^{3} \|x - s_k\|^{-1}$, $j = 1, 2, 3$. A single run of IDEA-ARIMA on the presented benchmark problems took approximately 2-4 hours and did not consume a great deal of memory.

Table 1 summarizes the number and the percentage of matches (i.e. correct selections of global optima among local ones) at each time step $H_{Start} < t \leq N_{Gen}$ of $g24\_1$ and $g24\_2$ benchmark problems. A selection is considered correct if the Euclidean distance between the global optimum and at least one feasible solution is less then $\delta = 0.1$. First H_{Start} iterations were not taken into account in Table 1 since observable results of both algorithms do not differ within this time period. It is clear that IDEA-ARIMA outperforms IDEA in each run.

Table 1. Summary of matches in 10 runs of IDEA and IDEA-ARIMA for benchmarks $g24\_1$ and $g24\_2$ with $N = 20, N_{Gen} = 4 \cdot 16 = 64, N'_{Gen} = 10, H_{Start} = 16$. First 16 iterations were ommited as observable results of both algorithms do not differ within this time period.

	benchmark $g24\_1$		benchmark $g24\_2$	
run	IDEA	IDEA-ARIMA	IDEA	IDEA-ARIMA
1.	21 (43.8%)	28 (58.3%)	5 (10.4%)	29 (60.4%)
2.	23 (47.9%)	26 (54.2%)	9 (18.8%)	25 (52.1%)
3.	13 (27.1%)	29 (60.4%)	19 (39.6%)	21 (43.8%)
4.	19 (39.6%)	30 (62.5%)	10 (20.8%)	23 (47.9%)
5.	19 (39.6%)	28 (58.3%)	9 (18.8%)	26 (54.2%)
6.	15 (31.3%)	26 (54.2%)	14 (29.2%)	26 (54.2%)
7.	22 (45.8%)	29 (60.4%)	11 (22.9%)	23 (47.9%)
8.	16 (33.3%)	29 (60.4%)	12 (25.0%)	23 (47.9%)
9.	22 (45.9%)	29 (60.4%)	18 (37.5%)	23 (47.9%)
10.	21 (43.8%)	26 (54.2%)	14 (29.2%)	22 (45.8%)

Table 2 summarizes the results in all the three variants of the modified FDA1 problem. As an additional comparison for performance of IDEA and IDEA-ARIMA, two simple modifications of these algorithms were used. One of them (named *iterated IDEA*) is the original IDEA algorithm yet reinitialized with a new random population in each iteration. Another one (named *perfect prediction*) is similar to IDEA-ARIMA only the actual value of $F^{(t+1)}$ was used instead of the predicted one at each time step t. Each row of Table 2 contains mean values and variances of minimal values of $F^{(t)}$, $H_{Start} < t \leq N_{Gen}$, found by a given algorithm among feasible solutions after N'_{Gen} iterations in n-dimentional modified FDA1 benchmark problem. Since (15) and (16) hold, it is clear that the algorithm with the lowest mean value and variance is superior to the others. As it is presented in Table 2, IDEA-ARIMA outperforms IDEA in majority of the test cases. However, for $n = 20$ no algorithm reached mean value below 0.2.

Figure 1 depicts the object functions (left) and its predictive counterparts (right) at sample time steps in benchmark problems: (a) $g24\_1$ and (b) modified FDA1 with static constraints. Darker shades correspond to lower values of $F^{(t)}$.

Table 2. Summary of results in the three variants of modified FDA1 benchmark problem with $N = 100, N_{Gen} = 8 \cdot 8 = 64, H_{Start} = 16$. First 16 iterations were ommited as observable results of both algorithms do not differ within this time period.

(a) No constraints:

		iterated IDEA		IDEA		IDEA-ARIMA		perfect prediction	
n	N'_{Gen}	mean	variance	mean	variance	mean	variance	mean	variance
2	10	0.002160	0.000006	0.001493	0.000008	0.001043	0.000008	0.000335	$< 10^{-6}$
2	20	0.000256	$< 10^{-6}$	0.000024	$< 10^{-6}$	0.000147	$< 10^{-6}$	0.000076	$< 10^{-6}$
5	10	0.092458	0.004905	0.242495	0.028307	0.083589	0.006497	0.011662	0.000097
5	20	0.020289	0.000723	0.069089	0.004967	0.013313	0.000508	0.001052	0.000004
10	10	0.378588	0.015056	0.424032	0.038679	0.297579	0.023194	0.164623	0.013937
10	20	0.257860	0.020371	0.352496	0.041520	0.139074	0.020252	0.052366	0.00381
20	10	0.617583	0.010092	0.567337	0.049116	0.571919	0.027719	0.483748	0.017680
20	20	0.559521	0.014415	0.546209	0.031698	0.483610	0.035850	0.394778	0.025932

(b) Static constraints:

		iterated IDEA		IDEA		IDEA-ARIMA		perfect prediction	
n	N'_{Gen}	mean	variance	mean	variance	mean	variance	mean	variance
2	10	0.026037	0.000957	0.011909	0.000859	0.006696	0.000271	0.006580	0.000086
2	20	0.007736	0.000135	0.001421	0.000025	0.001108	0.000016	0.000953	0.000002
5	10	0.161832	0.006272	0.079280	0.009862	0.070884	0.008095	0.060071	0.004535
5	20	0.071655	0.004494	0.026093	0.003576	0.023647	0.000985	0.011907	0.000304
10	10	0.439336	0.015338	0.186657	0.036261	0.156280	0.021570	0.151008	0.013162
10	20	0.310564	0.016344	0.087379	0.020944	0.114495	0.014076	0.043490	0.001861
20	10	0.653699	0.010381	0.398609	0.043211	0.458713	0.035772	0.423361	0.033427
20	20	0.593292	0.012989	0.231508	0.036150	0.349036	0.047805	0.279418	0.034973

(c) Dynamic constraints:

		iterated IDEA		IDEA		IDEA-ARIMA		perfect prediction	
n	N'_{Gen}	mean	variance	mean	variance	mean	variance	mean	variance
2	10	0.010577	0.000121	0.004557	0.000205	0.001837	0.000013	0.001362	0.00005
2	20	0.002520	0.000007	0.000444	0.000005	0.000579	0.000001	0.000171	$< 10^{-6}$
5	10	0.142978	0.007100	0.086602	0.016235	0.079515	0.004154	0.021180	0.000513
5	20	0.033714	0.001274	0.082928	0.009907	0.015362	0.000335	0.003651	0.000014
10	10	0.440479	0.018068	0.209387	0.026222	0.305928	0.025859	0.212496	0.025188
10	20	0.282467	0.027742	0.186919	0.037881	0.158682	0.022923	0.057928	0.008040
20	10	0.669181	0.007789	0.387301	0.055594	0.611805	0.011683	0.580973	0.016480
20	20	0.581248	0.011613	0.399809	0.049096	0.405708	0.038885	0.381798	0.047503

5 Conclusions

This paper proposes an extension of the Infeasibility Driven Evolutionary Algorithm (IDEA) with a prediction mechanism based on ARIMA, where a population of feasible and infeasible solutions reacts much faster to the changing objectives due to information coming from the prediction mechanism and remains one time instant ahead of the original algorithm.

Although more detailed studies are necessary, preliminary experiments performed on popular benchmark problems confirm that IDEA-ARIMA algorithm outperforms IDEA in many cases.

There are some approaches that aim at combining standard evolutionary search with prediction mechanisms [6], [9]. This paper shows that ARIMA-based prediction gives promising results when applied to continuous object functions with low severity of changes. There exists a significant class of real world dynamic optimization problems with predictable objective functions where IDEA-ARIMA algorithm is applicable.

References

1. Box, G.E.P., Jenkins, G.M.: Time series analysis: Forecasting and control. Revised edition. Holden-Day, San Francisco (1976)
2. Bui, L.T., Abbass, H.A., Branke, J.: Multiobjective optimization for dynamic environments. In: Proceedings of Congress on Evolutionary Computation, pp. 2349–2356 (2005)
3. Coello Coello, C.A., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. IEEE Trans. Evolutionary Computation 8(3), 256–279 (2004)
4. Farina, M., Deb, K., Amato, P.: Dynamic Multiobjective Optimization Problems: Test Cases, Approximations and Applications. IEEE Trans. Evol. Comp. 8(5) (2004)
5. Greeff, M., Engelbrecht, A.P.: Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 2917–2924. IEEE, Los Alamitos (2008)
6. Hatzakis, I., Wallace, D.: Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In: Proceedings of the GECCO 2006, pp. 1201–1208. ACM, New York (2006)
7. Nguyen, T., Yao, X.: Benchmarking and solving dynamic constrained problems. In: IEEE Congress on Evolutionary Computation, CEC 2009 (2009)
8. Singh, H.K., Isaacs, A., Nguyen, T.T., Ray, T., Yao, X.: Performance of infeasibility driven evolutionary algorithm (IDEA) on constrained dynamic single objective optimization problems. In: Proceedings of IEEE Congress on Evolutionary Computation, CEC 2009, pp. 3127–3134 (2009)
9. Zhou, A., Jin, Y., Zhang, Q., Sendhoff, B., Tsang, E.: Prediction-Based Population Re-initialization for Evolutionary Dynamic Multi-objective Optimization. In: Obayashi, S., Deb, K., Poloni, C., Hiroyasu, T., Murata, T. (eds.) EMO 2007. LNCS, vol. 4403, pp. 832–846. Springer, Heidelberg (2007)
10. Zitzler, E., Thiele, L.: Multiobjective optimization using evolutionary algorithms - A comparative case study. In: Eiben, A.E., Bäck, T., Schoenauer, M., Schwefel, H.-P. (eds.) PPSN V 1998. LNCS, vol. 1498, pp. 292–301. Springer, Heidelberg (1998)

Multiobjective Optimization of Indexes Obtained by Clustering for Feature Selection Methods Evaluation in Genes Expression Microarrays

Rodolfo Garcia, Emerson Cabrera Paraiso, and Júlio Cesar Nievola

Post-Graduate Program in Informatics (PPGIa),
Pontifical Catholic University of Paraná (PUCPR),
Imaculada Conceição Street, 1155, Prado Velho, Curitiba, PR-Brazil
{rodolfobbgarcia,paraiso,nievola}@ppgia.pucpr.br

Abstract. The selection of relevant genes in microarray is an important task, since that in a single experiment expressions of thousands of genes are extracted. One way to evaluate feature selection methods in a dataset is by clustering the instances that have similar behaviors. The aim of this paper is to use a set of indexes that measure the quality of a clustering and, through the multiobjective optimization of this set, to show how it is possible to find the best feature selection methods in genes expression datasets obtained by microarray technique.

Keywords: clustering, multiobjective optimization, feature selection, gene expression, microarray.

1 Introduction

Microarray technique is responsible for extracting information about expression of large quantity of genes related to one characteristic [1]. The increased amount of gene information makes scientists manual analysis take years to be done, needing computational help to realize this work faster [2].

Due to the fact that only few genes are involved in the characteristic activation or inhibition process, many of them are considered redundant or irrelevant because they do not improve the results of analysis. Thus, it is extremely important to select the most relevant features and get results with better comprehensibility [3].

This task is performed by the data mining step called feature selection. Its goal is select the most representative subset of features from one database, conforming the evaluation criteria, in order to make the model easier to analysis [4]. This paper uses the C-FOCUS, *Relief*-F and the CFS, applied in genes expression datasets obtained by the microarray technique.

The C-FOCUS algorithm is a FOCUS extension to be used with nominal or discrete features [5]. It aims to return the minimal relevant features subset, called Min-Features [6].

Another method used in this paper is the Relief-F, a Relief extension for problems involving more than two classes and aims to estimate the best n features, where n is

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 353–360, 2011.

specified by the user. The more powerful is a feature to distinguish instances of different classes, the higher is its probability of being chosen [7].

The Correlation-based Feature Selection, or CFS, is a fast method which works in continuous features, using correlation measures and excluding the redundant features [8].

These methods belong to the filter approach, which can be used with any other data mining steps. Comparing them to the wrapper approach, they are simpler, faster and are useful in large dimensionality datasets, like the genes datasets are [4].

One way to evaluate the feature selection methods is by clustering instances that have similar behaviors. In the genes expression datasets, instances with similar behaviors can represent the same characteristic. Thus, a good clustering means that the features are relevant and the instances define well the classes that they belong to.

This paper uses the K-means algorithm for clustering due to its popularity and simplicity. The result is one partition with K clusters, where K is specified by the user, and each instance belong to only one cluster [9].

In the clustering, its quality is measured by the clusters evaluation indexes [10]. This paper will use the indexes C, Isolation and Jaccard, which belong to different approaches, to evaluate clusters obtained by K-means method, a well know clustering method.

The optimization of many indexes characterizes a multiobjective problem and, in this paper, the Multi-Objective Particle Swarm Optimization technique chooses the best features in the original genes expression datasets obtained by microarray technique.

This paper aims to use a set of cluster validation indexes and, through the multiobjective optimization of this set, to show that it is possible to choose the best feature selection methods in genes expression datasets.

Section 2 explains the basics concepts of clustering and the clusters evaluation indexes. Section 3 introduces the multiobjective technique and explain how the Multi-Objective Particle Swarm Optimization method works and defines the best solution for a multiobjective problem. In section 4 is presented the proposal of this paper, the steps realized and the datasets used. Then, in section 5, the experiments and the results are described. Section 6 draws some conclusions.

2 Clustering

The clustering step aims to organize a set of instances in a way that those with similar behaviors will be in the same cluster without knowing the existing classes [11]. A good clustering means that the features are relevant and the instances define well the classes that they belong to. For this reason, this step can be used to evaluate feature selection methods.

One of the most simple clustering method is the K-means. With the clustering realized, the partition needs to be evaluated with clusters validation indexes. There are three types of cluster validation index: relative, internal and external. The relative index is generally used to compare two different clustering schemes to determine the correct number of clusters in a dataset. Internal indexes measure the quality of a clustering using only information contained in the dataset. Finally, the external indexes evaluate a clustering based on another clustering already obtained [10].

In [12] and [10] are presented the following indexes that belong to different approaches and are used here: C, Isolation and Jaccard indexes.

The index called C is a relative index and is optimized by minimizing the Equation 1. The quantity of all pairs of samples that belong to the same cluster is called p, S_U is the sum of the distances of those p that belong to cluster U. S_{min} and S_{max} are, respectively, p smaller distances and p biggest distances beetwen all samples in the dataset [12].

$$C(U) = \frac{S_U - S_{min}}{S_{max} - S_{min}} . \tag{1}$$

One index widely used in partitional clustering is the Isolation, an internal index. It calculates the separation between the centroid c_i of the cluster C_i and the global centroid c using the Sum of Squared Errors (SSE). Its optimization is done minimizing the Equation 2.

$$isolation(C_i) = SSE(c_i, c) . \tag{2}$$

The Jaccard index, being an external index, compares the clustering obtained (A_o) to a reference clustering (A_r). To [10], two instances are said:

- SS, if they belong to the same cluster in A_o and to the same cluster in A_r;
- SD, if they belong to the same cluster in A_o but to different clusters in A_r;
- DS, if they belong to different clusters in A_o but to the same cluster in A_r;

Being a_1, a_2 and a_3, respectively the number of pairs SS, SD and DS, the value of the Jaccard index is calculated by Equation 3 and the maximum value represents identical clusterings [10]:

$$jaccard = \frac{a_1}{(a_1 + a_2 + a_3)} . \tag{3}$$

3 Multiobjective Optimization

An optimization problem that deals simultaneously with many objectives is called multiobjective. In [13] can be found a mathematical formulation to an multiobjective optimization problem formed by m objectives (Equation 4). Each objective is composed of n parameters, also called decision variables, which are the possible solutions for the problem.

$$Optimize\ \vec{Y} = \vec{f}(\vec{Y}) = \left(f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)\right). \tag{4}$$

In a multiobjective problem is possible to choose multiple decision variables as the best solutions, since that one decision variable is not necessary the best for all objectives. This is the aim of Multi-Objective Particle Swarm Optimization (MOPSO) based on the Pareto front. It is an evolucionary algorithm inspired by the social behavior of a bird population flocking looking for food [14]. Besides being simple, another factor that made it popular was the efficiency in several applications, producing good results with low computational cost [15].

Each solution in the problem is a particle. The particles that move to promising regions are considered the best solutions and are called leaders. Although the possibility of having more than one leader, each particle can be guided by only one leader [16]. This guide can be chosen using the sigma method ($\vec{\sigma}$), presented in [17]. Besides being very effective in multiobjective problems based in swarm techniques, this method furnishes diversified solutions in only few iterations.

The $\vec{\sigma}$ vector is composed by $\binom{m}{2}$ elements , where m is the dimension of objective space. For a problem of three objectives f_1, f_2, f_3, the sigma vector will be composed by three elements, which are calculated in Equation 5. The leader with sigma vector closer to the particle sigma vector is the guide of this particle [17].

$$\vec{\sigma} = \begin{pmatrix} f_1^2 - f_2^2 \\ f_2^2 - f_3^2 \\ f_3^2 - f_1^2 \end{pmatrix} / ({f_1}^2 + {f_2}^2 + {f_3}^2) \; . \tag{5}$$

To elect the leaders, this paper uses the method based on the Pareto front, which generates many leaders in only one iteration [18]. The Pareto dominance rule is defined to a minimizing problem with m objectives as [18]:

- Given two particles $\vec{x}$ e $\vec{y}$ that belong to the same solution space π, $\vec{x}$ dominates $\vec{y}$, or $\vec{x} \prec \vec{y}_{,}$ if $\vec{x} \leq \vec{y}$, for $i = 1, \ldots, m$ and $\vec{x} \neq \vec{y}$;

If there is no solution $\vec{y}$, where $f(\vec{y}) \prec f(\vec{x})$, $\vec{x}$ is said Pareto Optimal, is elected leader and can be defined as the optimal solution for the problem [13]. The Pareto Optimal set forms the Pareto Front and the best solution is selected by a decision maker, that can be the user [18].

4 Proposal

The aim of this paper is to use a set of cluster validation indexes and, through the multiobjective optimization of this set, to show that it is possible to choose the best feature selection methods in genes expression datasets.

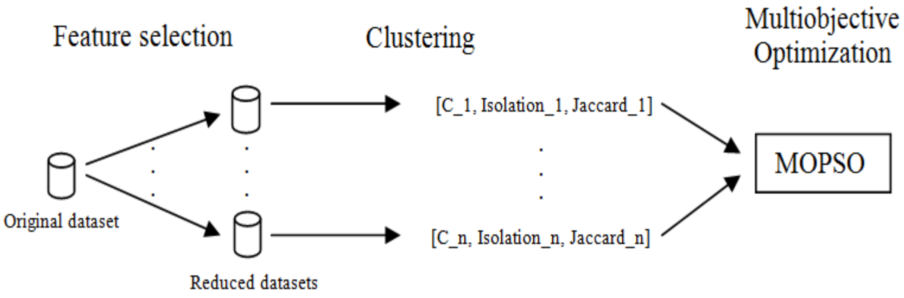

Feature selection
Clustering
Multiobjective Optimization
Original dataset
Reduced datasets
[C_1, Isolation_1, Jaccard_1]
[C_n, Isolation_n, Jaccard_n]
MOPSO

Fig. 1. Work data flow diagram

Figure 1 shows all the steps performed in this work using n feature selection algorithms. The datasets used here, presented in Table 1, are formed by instances with genes expressions from people with diseases related to cancer and healthy people. They were used in [19] and are available in "arff" format with its documentations in the Kent Ridge page [1].

Table 1. Datasets characteristics

Name	Number of instances	Number of clusters	Number of attributes
DLBLC-Stanford	47	2	4026
DLBLCL-NIH	80	2	7399
ALL-AML	34	2	6817

All datasets were subjected to the process of feature selection by three methods, C-FOCUS, CFS and *Relief*-F, this last reducing the datasets to 10%, 25%, 50% and 75% of the number of features in the original dataset.

Each dataset for one characteristic are evaluated by the cluster validation indexes C, Isolation and Jaccard through the K-means clustering, as can be seen in Figure 1. The Jaccard index is calculated by the comparison between the clusters obtained by the clustering method and the real clustering, since there is *a priori* knowledge about the composition of the genes expression datasets. The prior knowledge was also used to specify the number of clusters in the K-means clustering method.

In the end of the clustering step seven solutions were built, each one composed by three indexes. Multiple indexes makes the problem a multiobjective optimization one. This paper uses the evolutionary behavior of MOPSO based on the Pareto front to move the solutions, or particles, following its guides chosen by the three criteria sigma method to the promising regions. The leaders elected at the end of the execution, when no alterations occurs in the repository, are considered the best solutions, revealing the best feature selection methods for the dataset used.

Optimize the clusters validation indexes means the maximization of the Jaccard index and the minimization of the C and Isolation indexes. Solutions that satisfy the optimization are inside the promising region.

To calculate the velocity of a particle, $r_1, r_2 \in [0,1]$ were chosen randomly for each iteration. The inertia value was fixed in 1 to facilitate the global exploration [16]. The knowledge factors were fixed in 2 due to experience acquired by [19].

For the fact that few solutions are generated, all of them belong to the same neighborhood, forming a full connected graph. According to [16], this topology converge to the final result faster.

All the experiments were realized in a Intel ® Core™ I7 with 1.73GHz and 6GB RAM machine. The MOPSO based on Pareto front was developed in Java language, as well as the feature selection methods, the K-means and the C, Isolation and Jaccard indexes. They use the Weka library, which is compatible with "arff" format [20].

[1] http://datam.i2r.a-star.edu.sg/datasets/krbd/, acessado em 07 de Abril de 2011.

5 Experiments and Results

The experiments performed here aims to optimize a set of cluster validation indexes using the MOPSO to evaluate feature selection methods.

For each dataset are generated seven solutions. These solutions are formed by the results of the clusters validation indexes C, Isolation and Jaccard, on the K-means clustering method, using different feature selection methods.

In Tables 2, 3 and 4 are shown the number of each seven solutions and the respective C, Isolation and Jaccard indexes values. The enumeration of the solutions is organized as follow:

1- Solution obtained by C-FOCUS method;
2- Solution obtained by CFS method;
3- Solution obtained by Relief-F method with 10% of original size;
4- Solution obtained by Relief-F method with 25% of original size;
5- Solution obtained by Relief-F method with 50% of original size;
6- Solution obtained by Relief-F method with 75% of original size;
7- Solution obtained by original dataset;

Table 2. DLBCL-Stanford dataset solutions

Solution	C	Isolation	Jaccard
1	0.31	0.46	0.33
2	0.17	1.15	0.35
3	0.25	2.52	0.56
4	0.15	3.75	0.39
5	0.15	5.82	0.33
6	0.15	6.83	0.33
7	0.55	8.88	0.34

With the DLBCL-Stanford dataset, presented in Table 2, the MOPSO returned the solutions 1,2,3,4 as leaders, since the solutions 5 and 6 are dominated by solution 4 and the solution 7 is dominated by solutions 3 and 4. After the iteration, the leaders were the same of before the multiobjective optimization.

Table 3. DLBCL-NIH dataset solutions

Solution	C	Isolation	Jaccard
1	0.23	0.77	0.37
2	0.75	0.80	0.37
3	0.87	3.35	0.34
4	0.72	5.41	0.35
5	0.64	9.88	0.35
6	0.73	10.6	0.36
7	0.77	11.1	0.36

In the results presented in Table 3 from the DLBCL-NIH dataset, it is possible to observe that the solution 1 dominates the others solutions. When the MOPSO was performed, only the solution 1 was selected as leaders and considered the best feature selection method for this characteristc.

Table 4. ALL-AML dataset solutions

Solution	C	Isolation	Jaccard
1	0.51	0.30	0.32
2	0.51	0.30	0.32
3	0.34	4.43	0.44
4	0.32	7.24	0.44
5	0.45	10.6	0.45
6	0.54	14.34	0.47
7	0.49	15.2	0.45

Table 4 shows the results for the ALL-AML dataset. The only solution dominated by another solution was the number 7, since the solution 5 dominates it. Solution 1 do not dominate the solution 2 because they are identical. The MOPSO chose all the solution, except the number 7, as leaders and good solutions for this characteristic.

6 Conclusions

This paper evaluated, through MOPSO, feature selection methods in genes expression datasets obtained by microarray technique.

As can be seen from the results, feature selection methods can be evaluated by clusters validation indexes obtained by K-means since datasets with less features had better results than the original datasets. It is interesting to use more than one clustering method because different methods build different clusterings, which can be lead to better results.

The multiobjective method worked as expected when optimizing clusters validation indexes with different objectives, since C and Isolation indexes aim to minimize and the Jaccard index aims to maximize its values. Due to the few solutions, it was possible to evaluate each solution manually and confirm that MOPSO chose consistently the solutions that actually is considered the best one. Then, the clustering step can be used to evaluate feature selection methods.

In most of the datasets, it was obtained more than one solution as leader, which shows the importance of a decision maker. This decision maker can be a user with a deep knowledge about the datasets in order to choose the best solution.

Using feature selection methods, the results showed that reduced datasets were classified as best solutions in all experiments, since the original dataset was never selected as a leader. Besides that, the results of C-FOCUS method, which generates the smallest feature set, are always present in the final leaders set.

Acknowledgments. The authors of this paper would like to thanks CNPq (Brazilian Council for Scientific and Technological Development) for the financial support (project Feature Selection using Multiobjective Criteria to Microarrays Genes Selection, reference 555264/2009-2).

References

1. D'Haeseleer, P.: How Does Gene Expression Clustering Work?, vol. 23(12), pp. 1499–1501 (2005)
2. Dy, J.: Unsupervised Feature Selection. In: Liu, H., Motoda, H. (eds.) Computational Methods of Feature Selection, pp. 19–39. Chapman & Hall/CRC (2008)
3. Handl, J., Knowles, J.: An Evolutionary Approach to Multiobjective Clustering. IEEE/ACM Transactions on Evolutionary Computation 11(1), 56–76 (2007)
4. Yu, L., Liu, H.: Redundancy Based Feature Selection for Microarray Data. In: 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2004)
5. Azofra, A.A., Sanchez, J.M.B., Peña, J.L.C.: C-FOCUS: A continuous Extension of FOCUS. In: Advances in Soft Computing - Engineering, Design and Manufacturing, pp. 225–232 (2003)
6. Dash, M., Liu, H., Yao, J.: Dimensionality Reduction of Unsupervised Data. In: 9th International Conference on Tools with Artificial Intelligence (ICTAI 1997), p. 532 (1997)
7. Kononenko, I., Sikonja, M.R.: Non-Myopic Feature Quality Evaluation with (R)ReliefF. In: Liu, H., Motoda, H. (eds.) Computational Methods of Feature Selection, pp. 169–191. Chapman & Hall/CRC (2008)
8. Hall, M.: Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning. In: 17th International Conference on Machine Learning (2000)
9. Berkhin, P.: Survey of clustering data mining techniques. In: Accrue Software Technical Report, USA (2003)
10. Faceli, K., Caravalho, A., Souto, M.: Validação de Algoritmos de Agrupamento. ICMC techinical reports (2005)
11. Xu, R., Wunsch, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16(3) (2005)
12. Bolshakova, N., Azuaje, F.: Estimating the Number of Clusters in DNA Microarray Data. Methods Inf. Med. 45(2), 153–157 (2006)
13. Suresh, K., Kundu, D., Ghosh, S., Das, S.: Data Clustering Using Multi-objective Differential Evolution Algorithm. Fundamenta Informaticae 21, 1001–1024 (2009)
14. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Press, Los Alamitos (1995)
15. Chuang, L.Y., Chang, H.W., Tu, C.J., Yang, C.H.: Computational Biology and Chemistry, vol. 32, pp. 29–38. Elsevier, Amsterdam (2008)
16. Carvalho, A.B., Pozo, A.T.R.: Otimização por Nuvem de Partículas Multiobjetivo na Aprendizagem Indutiva de Regras: Extensões e Aplicações. Universidade Federal do Paraná, Master thesis (2009)
17. Mostaghim, S., Teich, J.: Strategies for finding good local guides in multi- objective particle swarm optimization (MOPSO). In: IEEE Swarm Intelligence Symposium, pp. 26–33 (2003)
18. Coello, C.A., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd edn. Springer, Heidelberg (2007)
19. Borges, H.B.: Redução de Dimensionalidade em Bases de Dados de Expressão Gênica. Pontifíca Universidade Católica, Master thesis (2006)
20. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA Data Mining Software: An Update. SIGKDD Explorations 11(1) (2009)

SignalNet: Visualization of Signal Network Responses by Quantitative Proteome Data

Christoph Gernert, Frank Klawonn, and Lothar Jänsch

Helmholtz Centre for Infection Research
Christoph.Gernert@Helmholtz-HZI.de

Abstract. Interactome databases summarize our present knowledge of how proteins can interact at the molecular level under variable conditions. Signal networks, in which proteins and their interactions are represented by nodes and edges, constitute an essential part in these interactomes. The subset of nodes and edges, which become involved under certain biological conditions, necessitate the integration of further experimental information. *Mass spectrometry* used in proteomics can provide such data describing the expression and responses of nodes in signal networks.

SignalNet is a program that connects mass spectrometry (MS) data with a protein interaction database to recognize most likely utilized or affected signaling pathways. Regulatory information derived from *quantitative* MS analyses is used to calculate and visualize which nodes feature altered expression or response levels. Since signals naturally propagate from node to node, SignalNet also emphasizes edges, which are overconnected to several regulated nodes. Both the regulation factor and the robustness of the underlying MS data are statistically evaluated and assigned to the nodes and edges. Thus SignalNet can filter highly complex interactome data to extract information about signal networks coordinating certain biological conditions. Through this filtering ordinarily densely connected interaction networks get purged from irrelevant interactions. By the presentation of a reduced network with respect to the MS data, the actual observed state of the cell can be resolved.

1 Introduction

Current MS instruments produce enormous amounts of data in comparative proteomics experiments designed to decipher signal transduction pathways and networks in eukaryotic cells. The identification of the logic and dynamic of the observed biological networks is a challenge for systems biology as the data must be analyzed to build models that represent the observed biological systems. The huge amount of data generated is usually processed through computer-based analysis. However, comparison of the different datasets and hypothesizing is still difficult. Often the differences in two or more datasets resulting from various samples look interesting, but only statistical analyses can determine whether the observed changes are relevant for the investigated biological system. This

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 361–368, 2011.

results from the fact that observed effects in the datasets, e.g. differences in protein amounts detected, cannot directly be linked to biological functions.

Conventional visualization techniques present results irrespective of experimentally observed protein modifications, although they decisively control protein-protein interactions. Therefore, the development of new analysis and visualization techniques is becoming ever more important. Such tools must both provide an overview of the entire system, as well as give sufficiently detailed information about the individual components. To address this problem we introduce *SignalNet*, which generates protein interaction networks as multigraphs. Highlighting the most likely utilized informative signal transduction pathways and hiding irrelevant interactions, which were not proven by the MS data, SignalNet presents clear reduced interaction networks based on quantitative MS experiments.

In Section 2.1 *Sample preparation* a short introduction to quantitative MS with iTRAQ labeling is given. The remainder of Section 2 *Methods* deals with the methodology of the system. SignalNet was successfully applied to an MS dataset of a time series experiment involving HGF-Met signaling analysis described in Section 3 *Results*, while in Section 4 *Conclusion* the functionality of the system is reviewed.

2 Methods

2.1 Sample Preparation

Proteome data for signaling network analysis were generated as described before [7]. In brief, proteins from samples representing individual signaling stages were digested to peptides. These peptides were than labeled differentially with isobaric stable isotope tags (iTRAQ).

An *iTRAQ label* consists of a reporter, balance, and reactive region. Lighter reporter regions are combined with heavier balance regions. The complete different labels are isobaric and therefore the label mass is always the same and will add the same mass shift to a peptide. After labeling, the samples are mixed together and then analyzed by liquid chromatography–tandem mass spectrometry (LC-MS/MS). As part of this process the peptides are fragmented into peptide fragment ions and ion signals from the reporter group of the labels. The peptide fragment ions reveal the amino acid sequence of the peptide. The ion signals of the iTRAQ labels are present in the low mass region of the mass spectrum. Through this signals, quantitative information can be obtained regarding the relative amount of the peptide in different samples (Figure 1).

The resulting spectra are sent against a UniProtKB/Swiss-Prot database via the Mascot Daemon software. Peptide quantification is then performed by an in-house bioinformatics tool, named *iTRAQassist*. In brief, the Mascot .dat file of one experiment, resulting from a merged peak list of multiple MS runs, and the respective by-product calibration file are loaded. *iTRAQassist* [2] then performs a normalization of reporter ion intensities and provides an estimation of the

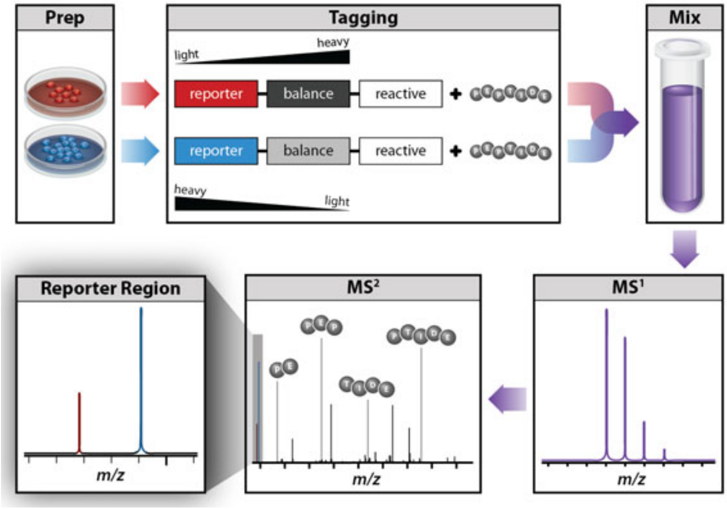


Fig. 1. Quantitative proteom analsis with iTRAQ. After labeling and combination of the samples, the ratio of the peptide mixture can be calculated from the ion signals in the mass spectrum. (Figure by A. J. Bureta CC BY-SA)

noise, based on statistical information, for the regulatory data of the individual peptide and protein levels.

2.2 Divergence

The divergence describes an aggregation of differently regulated posttranslational modifications, which indicates the extent in which a protein is involved in a cellular signaling pathway in a cell. This value is derived from changes in protein phosphorylation and measures the protein activity in the signal transduction network. The protein divergence is calculated from the sum of the single divergence values of the peptides assigned to a protein. Robustness values of the peptide regulations are gathered from the program *iTRAQassist*. The computation of the divergence values of peptides will be described below in detail. Phosphopeptides receive a score depending on their regulation and its robustness. Unmodified peptides do not get a score.

The *divergence* is calculated from the difference of the protein regulation factor (RFprot) and the phosphopeptide regulation factor (RFpep) taking into account the robustness of these factors (1). The robustness is described as an interval of the probable regulation (interval of robustness: IoR). The interval of robustness describes an 80% confidence interval for the possible regulation factor, based on the noise present in the signal [3].

The penalty (p) depends on the overlap of the two intervals (2). The overlap can be between zero for no overlap, and one for total overlap of the protein's intervals of robustness. The parameter m describes the denominator of the divergence for total overlap. E.g. if $m = 2$ the penalty would be 1/2. m can be increased if more robust results are wanted. For the divergence calculation $m = 2$ turned out as a good compromise between the raw MS and the noise corrected regulation data derived from the IoR.

$$div = \begin{cases} RF_{pep} - \left(RF_{prot} + \frac{IoR_{prot}}{2}\right) \cdot p \text{ for } RF_{pep} > RF_{prot} \wedge RF_{pep} \notin IoR_{prot} \\ \left(RF_{prot} - \frac{IoR_{prot}}{2}\right) - RF_{pep} \cdot p \text{ for } RF_{pep} < RF_{prot} \wedge RF_{pep} \notin IoR_{prot} \\ 0 \qquad \text{else} \end{cases} \tag{1}$$

$$p = \frac{1}{overlap(m-1)+1} \tag{2}$$

2.3 Parameters Q1 and Q2

In the protein-protein interaction network the parameters Q1 and Q2 determine how the protein divergence values are applied to the graph's edges to highlight a possible signal (Figure 2). Q1 affects all direct adjacent edges of a protein node. The edge thickness increases by an amount adjusted by this parameter.

Mass spectrometers usually detect peptides, which are specific protein fragments. In most cases proteins are identified by only a few peptides, covering only a fraction of the whole protein sequence. It is difficult to make a statement about the parts of the protein sequence which are not covered by the detected peptides. It is unknown whether these parts contain modifications important for the signal transduction or not. Q2 tries to compensate for this effect by affecting all edges that are adjacent to the neighbors of the corresponding protein node. By increasing the Q2 parameter, possible holes in the protein signal can be closed through forwarding a fraction of the divergence to the next node. On the other hand, the higher the Q2 parameter is chosen the less reliable the interaction network gets.

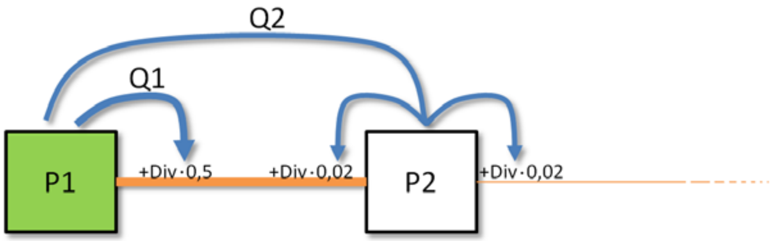
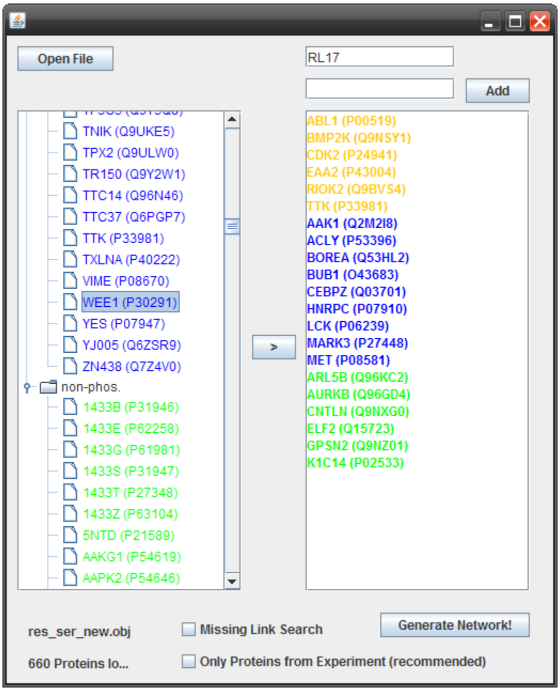


Fig. 2. The parameters Q1 and Q2 influencing the interaction network's edges. The case shows the results of Q1 = 0.5 and Q2 = 0.02.



RL17

TNIK (Q9UKE5)
TPX2 (Q9ULW0)
TR150 (Q9Y2W1)
TTC14 (Q96N46)
TTC37 (Q6PGP7)
TTK (P33981)
TXLNA (P40222)
VIME (P08670)
WEE1 (P30291)
YES (P07947)
YJ005 (Q6ZSR9)
ZN438 (Q7Z4V0)
non-phos.
1433B (P31946)
1433E (P62258)
1433G (P61981)
1433S (P31947)
1433T (P27348)
1433Z (P63104)
5NTD (P21589)
AAKG1 (P54619)
AAPK2 (P54646)

ABL1 (P00519)
BMP2K (Q9NSY1)
CDK2 (P24941)
EAA2 (P43004)
RIOK2 (Q9BVS4)
TTK (P33981)
AAK1 (Q2M2I8)
ACLY (P53396)
BOREA (Q53HL2)
BUB1 (O43683)
CEBPZ (Q03701)
HNRPC (P07910)
LCK (P06239)
MARK3 (P27448)
MET (P08581)
ARL5B (Q96KC2)
AURKB (Q96GD4)
CNTLN (Q9NXG0)
ELF2 (Q15723)
GPSN2 (Q9NZ01)
K1C14 (P02533)

res_ser_new.obj

660 Proteins lo...

☐ Missing Link Search

☐ Only Proteins from Experiment (recommended)

Fig. 3. SignalNet selection window. Phosphorylated proteins (blue) and non-phosphorylated proteins (green) can be transfered from the sample list (left) to the list for the network generation (right). Proteins that were not present in the sample can be added explicitly to list for network generation (orange).

2.4 Program Structure

SignalNet is written in Java and loads complex quantitative proteome data attached with robustness values calculated by iTRAQassist. After selecting phosphorylated and non-phosphorylated proteins out of the sample for the network building, protein interactions from the *HPRD*[1] are queried [5]. Proteins which were not found in the sample can be added explicitly for the network analysis (Figure 3). *Divergence* values are calculated and the network is saved as graph in a GML File. The GML file provides a graph with nodes including diagrams, representing the given protein's divergence of each reporter. The nodes are connected by directed and undirected edges depending on the protein interaction type. In addition, various data from the *HPRD* are queried and attached to the nodes and edges, representing proteins and their interactions. The regulation data from the mass spectrometry experiment are also attached to each node, if available.

[1] Human Protein Reference Database http://www.hprd.org

The GML file can be opened and edited with *VANTED*, a visualization and analysis software for signaling networks [4]. The view of the complex dataset can be switched with a plug-in developed for VANTED. The plug-in enables the view of the additional information attached to the nodes and edges. Furthermore, the parameters Q1 and Q2 can be adjusted to determine the edge thickness representing the protein divergence values.

3 Results

SignalNet was tested with an iTRAQ experiment made for HGF-Met signaling analysis [7]. Samples were extracted after the stimulation of the cells with HGF at t = 0, t = 2, t = 6 and t = 20 minutes. Relative quantification was realized through labeling of the samples with an iTRAQ 4-Plex-Kit. The possible candidates for the HGF-Met signaling were manually determined previously. Therefore the mass spectrometry data were analyzed manually and a literature research for possible protein-protein interactions was carried out without any automation.

Systems exist which can gather all known interactions between a number of given proteins. The interactions are queried from different sources like co-expression and signaling studies or pathway databases [6]. These interactions altogether do not reflect the actual signaling state of the analyzed sample, which only exhibit a small subset of these interactions.

The refined network calculated by *SignalNet* only shows the relevant interaction according to the mass spectrometry data and possible candidates for the HGF-Met signaling pathway could easily be identified (Figure 4). A good example in this case is the interaction with the protein *KS6A1.*

Among others the phosphorylation of *KS6A1 isoform 2* at the site *Serine-363* and *Serine-380* by *ERK1/2* could be validated. The relevant publication on this interaction [1] was automatically retrieved from the HPRD and could directly be accessed within SignalNet. Regarding the publication and the mass spectrometry results *S380* is autophosphorylated und further phosphorylated after binding with *ERK.* Because no significant changes at *S363* could be detected, the translocation of the *KS6A1/ERK* complex to the plasma membrane is unlikely (Table 1).

Table 1. Divergence (*div*) and RF (log_2 fold change) of the phosphosites S380 and S363 at *KS6A1*

	S380		S363	
t in min	*Div*	*RF*	*Div*	*RF*
2	0.43	0.48	0.08	0.07
6	5.36	2.68	0.31	-0.33
20	3.79	2.27	0.22	-0.22

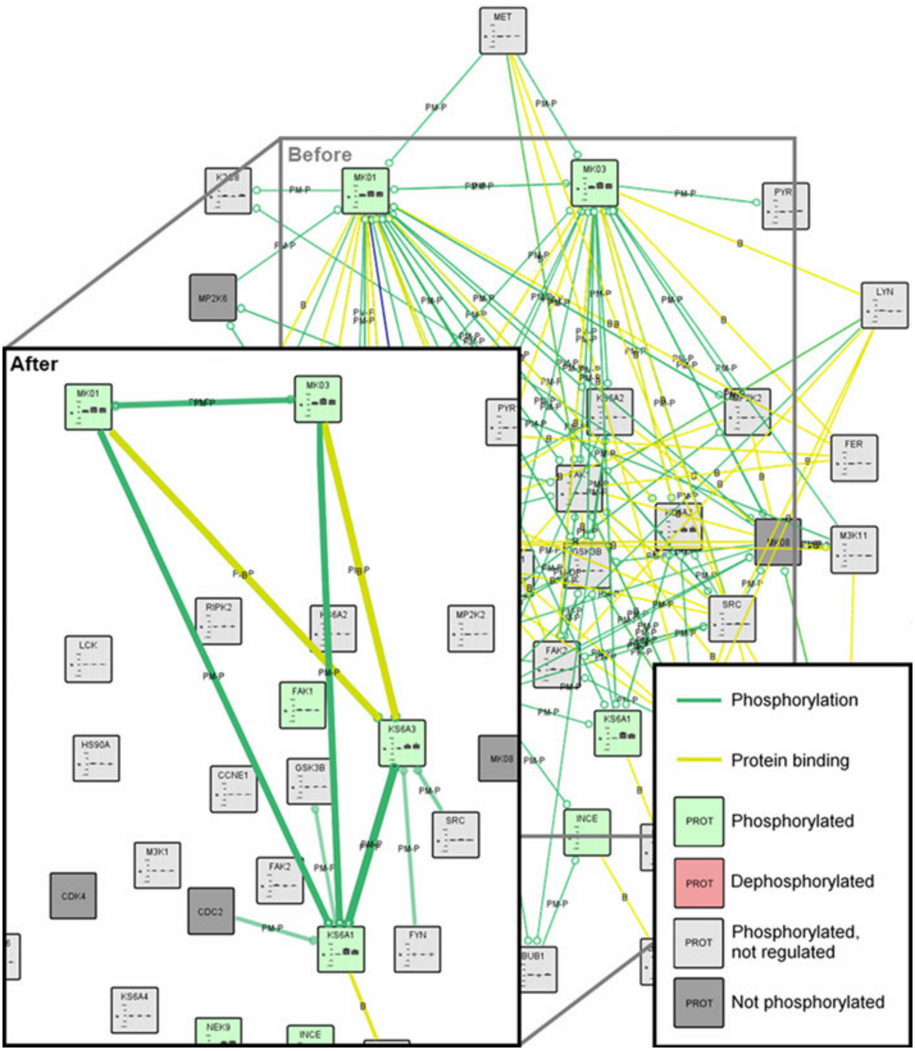


Fig. 4. This signaling network is based on stimulation of the c-MET receptor. The box "Before" shows the interaction network generated from the HPRD. The box "After" shows the refined network including the divergence values and the adjustment of the parameters Q1 and Q2 in the VANTED plug-in. In this case, only strong signaling events by phosphorylation are displayed.

4 Conclusion

The generated network shows the observed proteins and their interactions. Temporal or spatial changes are displayed and reveal the current most likely state of the signal transduction network. Unfortunately, the presence of false positive strongly regulated phosphopeptides is not excluded by the detection through mass spectrometry, but is strongly reduced by the way the divergence is calculated and used. The thickness of the edges is determined by robustness and strength of a regulation, derived from the divergence of the corresponding proteins. Inside of each protein node the phosphopeptide ratio is shown for each state of the experiment. The calculated interaction network contains detailed information and provides an overview of the complex proteome data. For time series experiments it even can reveal the signal flow in the cell.

References

1. Cavet, M.E., Lehoux, S., Berk, B.C.: 14-3-3beta is a p90 ribosomal S6 kinase (RSK) isoform 1-binding protein that negatively regulates RSK kinase activity. J. Biol. Chem. 278, 18376–18383 (2003)
2. Hundertmark, C., Fischer, R., Reinl, T., May, S., Klawonn, F., Jänsch, L.: MS-specific noise model reveals the potential of iTRAQ in quantitative proteomics. Bioinformatics 25, 1004–1011 (2009)
3. Hundertmark, C., Klawonn, F.: Clustering likelihood curves: Finding deviations from single clusters. In: Corchado, E., Abraham, A., Pedrycz, W. (eds.) HAIS 2008. LNCS (LNAI), vol. 5271, pp. 385–391. Springer, Heidelberg (2008)
4. Junker, B., Klukas, C., Schreiber, F.: VANTED: A system for advanced data analysis and visualization in the context of biological networks. BMC Bioinformatics 7(109), 1–13 (2006)
5. Keshava Prasad, T.S., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., et al.: Human Protein Reference Database - 2009 update. Nucleic Acids Res., D767–D772 (2009)
6. Mostafavi, S., Ray, D., Warde-Farley, D., Grouios, C., Morris, Q.: GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. Genome biology 9, S4 (2008), http://dx.doi.org/10.1186/gb-2008-9-s1-s4
7. Reinl, T., Nimtz, M., Hundertmark, C., Johl, T., Kéri, G., Wehland, J., Daub, H., Jänsch, L.: Quantitative phosphokinome analysis of the Met pathway activated by the invasin internalin B from Listeria monocytogenes. Mol. Cell Proteomics 8, 2778–2795 (2009)

Evolutionary Optimization of Regression Model Ensembles in Steel-Making Process

Miroslaw Kordos[1], Marcin Blachnik[2], and Tadeusz Wieczorek[2]

[1] University of Bielsko-Biala, Department of Mathematics and Computer Science,
Bielsko-Biala, Willowa 2, Poland
mkordos@ath.bielsko.pl

[2] Silesian University of Technology, Department of Management and Informatics,
Katowice, Krasinskiego 8, Poland
{marcin.blachnik,tadeusz.wieczorek}@polsl.pl

Abstract. In this paper we compare different evolutionary algorithm approaches and parameters used to optimize the output of neural network committee trained on regression problems. This is especially useful for large and complex datasets. We used the methodology presented in this paper to optimize the output of the committee to predict the temperature in the electric arc furnace in one of the steelworks.

Keywords: neural network committee, evolutionary algorithms.

1 Introduction

1.1 Temperature Control in the EAF

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

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

Although using the model described in [1] the temperature can be calculated 10 minutes before the tap and the end of the meltdown phase can be predicted, we

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 369–376, 2011.

need to know the temperature much earlier, to be able to optimize the parameters during the whole process.

Therefore there was a need to build a temperature prediction system that would allow us to limit the number of temperature measurements and thus shorten the EAF process. We previously built a system based on a single regression model [2], as a part of the whole intelligent system for steel production optimization [3]. However, as our recent experiments showed, a significant improvement can be obtained with a committee of regression models.

1.2 Committee Machines

The idea behind building a committee of regression models is to improve the accuracy of single models by weighting the outputs of particular models, where the sum of weights equals one. That works, because the bias of the whole committee is of the same level as biases of particular models, while the variance is smaller or at least not larger than variances of particular networks [4].

Bagging [5] provides decorrelation between the predictions of committee members by training each one on a different dataset, obtained by randomly drawing with replacement n data points from the original data set. In AdaBoost [5] the probability of selecting a vector in a next model is higher if the previous model made higher error on that vector.

It maybe easier to build a good model, dividing the whole data into several subsets, where a dedicated neural network is responsible for modeling each region of the input space. Based on that assumption Jacobs [6] created a Mixture of (local) Experts. Barbosa [7] used a population of feed-forward neural networks, which was evolved using the Clonal Selection Algorithm and the final ensemble was composed of the selected subset of the neural network population. Chen and Yao [8] proposed a Negative Correlation Learning, which introduces a correlation penalty term to the cost function of each network so that each network minimizes not only its mean-square-error but also the correlation with other networks. They used and evolutionary multi-objective algorithm to optimize the networks in the ensemble. Baruque and Corchado [9] instead of combining directly the results, formulated a single resultant model of the SOM networks, where the neurons composing the map were obtained as the fusion of neurons from different maps that are considered close enough to be fused into one. Bian and Wang [10] evaluated ensembles of 10 different learning algorithms, taking into account their ability for generating different types of diversity and analyzed their correlation with ensemble performance on fifteen popular data sets.

1.3 Evolutionary Algorithms

In the experimental part we compare how various selection, recombination and mutation schemes influence the convergence of the algorithms and based on the analysis we propose some solutions.

In selection individuals are chosen to produce the offspring with the probability proportional to the value of their fitness function. Recombination produces new individuals by combining the information contained in the parents. There maybe any number of parents from two up to the number of positions in the individual (here the number of neural networks) and any number of crossover points from the number of parents minus one to the number of positions minus one. In the intermediate recombination, the child's C_i position can be given a new value, which does not exist in any of the parents, but which is some combination of the parents' PX_i values, which for two parents is given by a rule (where a is randomly selected):

$$C_i = P1_i \cdot a + P2_i \cdot (1 - a), a \in (-0.25, 1.25) \tag{1}$$

In mutation of real variables small random values are added to the variables with a low probability.

With a high number of regression models, an intelligent model selection becomes an important issue of ensemble-based system design. A few approaches of applying evolutionary optimization for machine learning committees can be found in the literature. Ruta and Gabrys [11] used a two-stage majority voting committee for classification tasks with evolutionary-based committee selection. Jackowski and Wozniak [12] presented a machine learning algorithm used for training of a compound classifier system that consisted of a set of area classifiers. The evolutionary algorithms used to optimize the splitting of feature space into areas and selecting area classifiers, which outperformed the simple voting. However, most of the models were designed for classification and not regression problems.

2 Methodology

The solution, we develop takes advantage of the properties of bagging, Mixture of Experts and evolutionary approach and adds some specific dependencies of the dataset to find the optimal weights of each network in the committee, that can be different for different test vectors. The system is shown in Fig. 1.

There are several possible approaches to cluster the data, as fuzzy c-means or hierarchical clustering. Costa and Netto [13] proposed to use SOM networks for a hierarchical tree of maps, which represented the cluster's structure in levels of granularity, where each cluster of neurons gave rise to a new map, which was trained with the subset of data that were classified to it. Al-Harbi and Smith [14] constructed a supervised k-means algorithm to partition objects which have the same class label into one cluster, where by the means of Simulated Annealing they used the data outputs in developing a suitable metric defined on other fields. It also seems to be reasonable for the regression tasks after some adjustment of their method and we are planning to use that approach to improve the clustering in our future work.

However, because of already high complexity of the project, currently we decided to use the simplest method: the k-means clustering with Euclidean distance and then fuzzify the clusters (what for our purposes produced more adequate results than fuzzy c-means) to divide the data at four different levels:

- the whole training dataset
- 3 clusters
- 9 clusters
- 27 clusters

The rationale behind the 4-level hierarchy was to take into consideration local properties of the input space using the smaller clusters and still prevent the data over-fitting by using also the bigger clusters. These number of clusters and number of networks in each cluster was chosen, because it has been found to perform better than the 3-level hierarchy and the 4-level hierarchy would not provide enough data to train the last level models. Because the clusters had crisp boundaries and the neural networks should smoothly cover the entire input space, each point in the space was assigned a membership value to each cluster as the inverse of its Euclidean distance to the cluster center

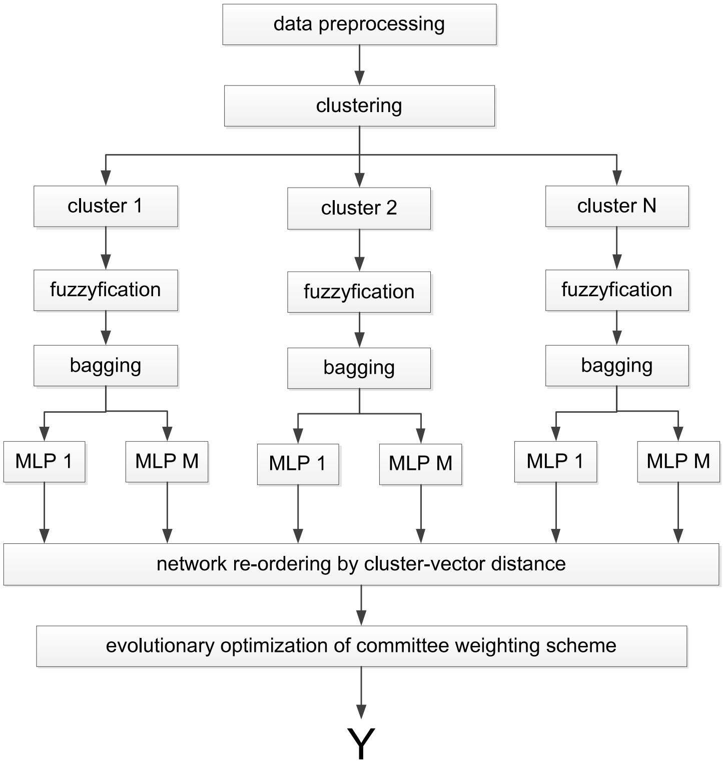


Fig. 1. The system used for temperature prediction in the EAF process

$$m = c/(d + c) \tag{2}$$

where c is the average distance of points in the current cluster to the cluster center and d is the distance between the given point and the cluster center.

During the network training, the error the network made on such a point was multiplied by the m value of the point. In this way the ideas of local experts was realized. In order to obtain five decorrelated local networks we implemented the simple bagging algorithm, because in the case of noisy data it can outperform AdaBoost. The number of selected vectors for each network dataset was: 3.000 for the whole training set and equal to 2/3 of the number of vectors in a given cluster for the lower levels. The probability of each vector being selected was proportional to its m value for the given cluster. In this way we obtained an ensemble of 5*(1+3+9+27)=200 neural networks. The number of five networks per cluster has been chosen because further increase of that number causes a growth of the system, which results in increasing the optimization difficulties and thus leads to no further improvement.

For the MLP network training we used the VSS algorithm [15]. The number of hidden layer neurons for each network was randomly chosen from 8 to 16 (based on the experiments this range of hidden layer neurons seemed optimal). The hidden and the output layer neurons had hyperbolic tangent transfer function. In the case of the output and input layer neuron this transfer function was chosen to limit the outliers' influence on the results as well as to make the data distribution closer to uniform [3]. However, the MSE we report in the experiment section are based on the original standardized data, that is after transposing the network output by an area tanh function. At the test phase, first, the distances of the test vector to the cluster centers are calculated. Then only a limited number of the neural networks trained on the clusters that are closer to the test vector take part in the temperature prediction; the networks trained on the whole dataset, all networks from the 3-cluster level, the networks of six of the 9-cluster level and the networks of nine of the 27-cluster level. Thus, together 5*(1+3+6+9)=95 neural networks predict the output value for each test vector. Then the networks of each level cluster that take part in the prediction are sorted according to the vector distance to their corresponding cluster centers and their results are saved to an array in the following order: first the whole data set networks, then the 3-cluster level networks starting from the cluster closest to the test vector, then the 9 and 27-cluster networks, each time starting from the cluster closest to the test vector. Thus, the positions in the chromosome represent the order of distances between the network and the test vector and not to a constant order of the networks.

The networks training and prediction is done only once. Only the procedure of determining the model weighing scheeme is iterative, what makes the process relatively fast. To find the weights w, the following algorithm is used:

1. Create a population of N individuals of randomly generated 95-entry array of weights from the interval (0,1).
2. When summing the weighted network outputs, multiply each weight by a value inversely proportional to the cluster order on the list plus 1, that is by 1/2 the weight of the closest cluster, by 1/3 the weight of the second closest cluster, etc. in each cluster level. Though the step may be omitted, it significantly improves the convergence of the algorithm.

3. Calculate the quality (MSE) of each solution over all test vectors using the following formula for the predicted output of every vector:
4. From here the evolutionary algorithms in different variants, as described in the previous section with different parameters are used.

There are two purposes of the experiments: to find the weighting scheme of the neural network committee that produces the lower error and to examine the influence of particular parameters on the process.

3 Experimental Results

The dataset and the software is available at *www.kordos.com/his.html*. The dataset consists of 50 continuous attributes, each of them describing one of the measured parameters, as particular temperatures, electric energy, amount of gas, time intervals, etc, denoted as x1...x50 and an output value - a temperature to be predicted, denoted as y. There are 7401 vectors in the dataset. The data is already standardized.

The table shows the accuracy in 10-fold cross-validation using different regression algorithms. MLP-EA is the method described in this paper. MLPTree is a regression tree with MLP networks in its nodes. Tree is a standard one-dimensional regression tree. Tree Forests consist of 10 such trees, where the output of each tree is weighed inversely proportional to its MSE on the test set. MLP-bagging and MLP-AdaBoost are committees of 10 MLP networks trained on points drawn from the whole dataset. SVR-L and SVR-G are support vector regression models with linear and Gaussian kernels. Fig. 2. presents the number of times the fitness function has to be evaluated till the system converged to the level that no further improvement grater than 0.1% can be observed.

Table 1. Comparison of results obtained with various methods in 10-fold cross-validation

method	MLP-EA	MLPTree	MLP	Tree	Tree Forest
MSE	0.46 ± 0.05	0.51 ± 0.07	0.60 ± 0.10	0.80 ± 0.08	0.70 ± 0.06

method	MLP-bagging	MLP-AdaBoost	SVR-L	SVR-G	default
MSE	0.55 ± 0.07	0.58 ± 0.07	0.61 ± 0.06	0.61 ± 0.06	1.00 ± 0.0

The obtained results had almost the same accuracy with the different parameters shown in the table, however, the differences in the convergence time were enormous. The horizontal axis shown the number of times fitness function had to be evaluated, what directly corresponds to the computational complexity. The number on the vertical axis stands for: fitness function (1-linear, 2-quadratic), number of parents, number of different sections in recombinations (number of crossover points minus one), mutation percentage. In all cases the population size was fixed at 256 individuals (as it was within the range of best results). The intermediate recombination was used (eq. 2).

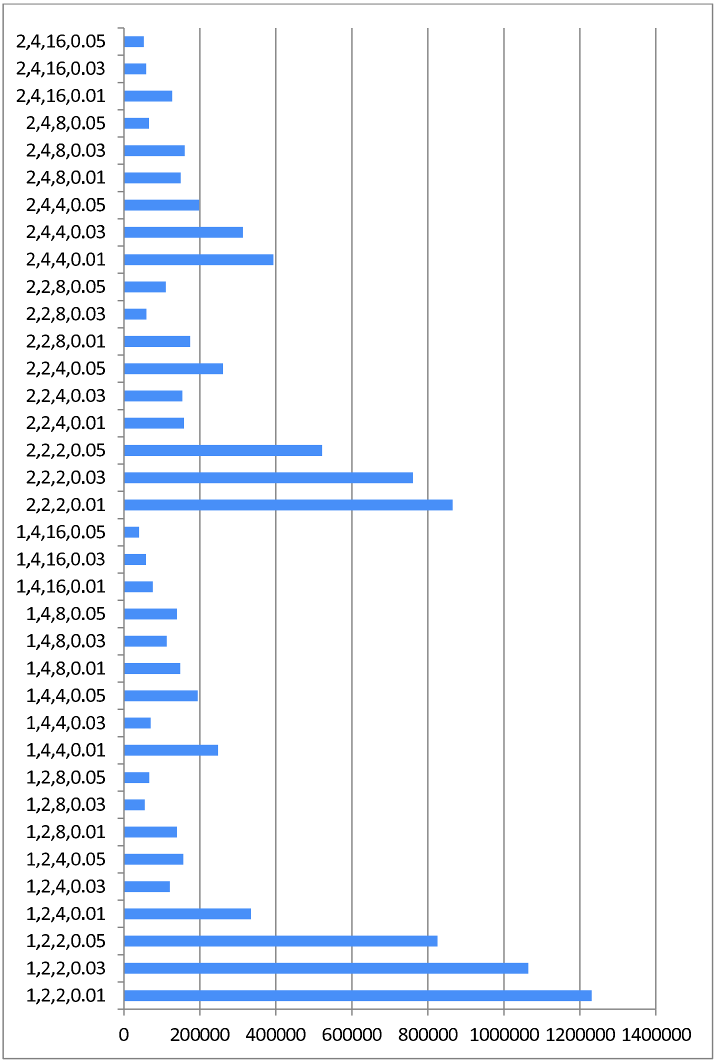


Fig. 2. Number of fitness function evaluations depending on fitness function (1-linear, 2-quadratic), number of parents, number of crossover points and mutation percentage

4 Conclusions

On case of big and complex datasets it is worth clustering the data into smaller parts to use the local expert models and then make smooth transitions between the models. The transition was implemented here by the cluster fuzzyfication. It is also beneficial to use ensembles of learning models and the best results maybe obtained by joining the both of the approaches. The committee with evolutionary optimized weighing scheme performs best, what was not difficult to guess, because here the apriori made assumption about the weighing scheme are precisely adjusted during the model learning process.

However, the results obtained with a decision tree allow for simple extraction of logical rules from the data and therefore maybe also very useful.

Currently, the biggest barrier to further improvement of the results is the quality of the data itself, which contains a lot of wrong and inaccurate value that are very difficult to eliminate.

Acknowledgement. The project was sponsored by the grant No. 4866/B/T02/2010/38 from the Polish Ministry of Education and Science.

References

1. Wendelstorf, J.: Analysis of the EAF operation by process modeling. Archives of Metallurgy and Materials 53(2), 385–390 (2008)
2. Wieczorek, T., Blachnik, M., Mączka, K.: Building a model for time reduction of steel scrap meltdown in the electric arc furnace (EAF): General strategy with a comparison of feature selection methods. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) ICAISC 2008. LNCS (LNAI), vol. 5097, pp. 1149–1159. Springer, Heidelberg (2008)
3. Kordos, M.: Neural Network Regression for LHF Process Optimization. In: Int. Conf. on Neural Information Processing, Auckland, New Zealand (2008)
4. Tresp, V.: Committee Machines. Handbook for Neural Network Signal Processing. CRC Press, Boca Raton (2001)
5. Breiman, L.: Combining predictors. In: Sharkey, A.J.C. (ed.) Combining Artificial Neural Nets, Springer, Heidelberg (1999)
6. Jacobs, R., et al.: Adaptive mixtures of local experts. Neural Computation (3(79)) (1991)
7. Barbosa, B., Bui, L.T., Abbass, H.A., Aguirre, L.A., Braga, A.P.: Evolving an Ensemble of Neural Networks Using Artificial Immune Systems. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abbass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 121–130. Springer, Heidelberg (2008)
8. Chen, H., Yao, X.: Multiobjective Neural Network Ensembles Based on Regularized Negative Correlation Learning. IEEE Trans. On Knowledge and Data Engineering 22, 1738–1751 (2010)
9. Baruque, B., Corchado, E.: A Weighted Voting Summarization of SOM Ensembles. Data Mining and Knowledge Discovery 21, 398–426 (2010)
10. Bian, S., Wang, W.: On diversity and accuracy of homogeneous and heterogeneous ensembles. Int. Journal of Hybrid Intelligent Systems 4(2), 103–128 (2007)
11. Ruta, D., Gabrys, B.: Genetic algorithms in classifier fusion. Applied Soft Computing (6), 337–347 (2006)
12. Jackowski, K., Wozniak, M.: Method of classifier selection using the genetic approach. Expert Systems 27(2), 114–128 (2010)
13. Costa, J.A.F., Netto, M.L.A.: Clustering of complex shaped data sets via Kohonen maps and mathematical morphology. In: Dasarathy, B. (ed.) Proceedings of the SPIE, Data Mining and Knowledge Discovery, vol. 4384, pp. 16–27 (2001)
14. Al-Harbi, S.A., Smith, R.: The use of a supervised k-means algorithm on real-valued data with applications in health. In: Chung, P.W.H., Hinde, C.J., Ali, M. (eds.) IEA/AIE 2003. LNCS, vol. 2718, pp. 373–387. Springer, Heidelberg (2003)
15. Kordos, M., Duch, W.: Variable step search algorithm for feedforward networks. Neurocomputing 71(13-15), 2470–2480 (2008)

Gradient Descent Decomposition for Multi-objective Learning

Marcelo Azevedo Costa and Antônio Pádua Braga

Universidade Federal de Minas Gerais
Departament of Statistics and
Department of Electronics
Av. Antônio Carlos 6627, 31.270-901, Belo Horizonte, Minas Gerais, Brazil
http://www.litc.cpdee.ufmg.br

Abstract. Multi-objective learning has been explored in neural network because it adjusts the model capacity providing better generalization properties. It usually requires sophisticated algorithms such as ellipsoidal, sliding-mode, genetic algorithms, among others. This paper proposes an affordable algorithm that decomposes the gradient into two components and it adjusts the weights of the network separately. By doing so multi-objective learning with L_2 norm control is accomplished.

Keywords: Multi-objective learning, bias and variance.

1 Introduction

It is well understood that Artificial Neural Networks (ANNs) learning should be accomplished by minimizing both the empirical risk $R_{emp}(\mathbf{w})$ and the model capacity h [1,2]. Convergence to the true risk [1] under the single-objective assumption, considering only $R_{emp}(\mathbf{w})$, can only be achieved in the large samples scenario. In such a situation, when the number of samples N tends do infinity ($N \to \infty$), the approximating function $f(\mathbf{x}, \mathbf{w})$ tends to expected value $E[y|\mathbf{x}]$ of the data for large enough model capacity. However, the large samples situation does not happen very often. In most real problems, the sample size is small and minimization of $R_{emp}(\mathbf{w})$ does not assure approximation to the generator function. In such situations, model capacity h should also be considered in order to improve convergence to the true risk. Since $R_{emp}(\mathbf{w})$ and h do not have coinciding minima, these two objective functions can only be traded-off instead of jointly minimized.

The conflicting behaviour of $R_{emp}(\mathbf{w})$ and h and the need for a trade-off between them suggests that the ANNs learning problem is inherently Multi-objective and that it should be treated like that. Therefore, from the optimization perspective, Pareto set solutions [3] are optimal and offer optimized trade-off between $R_{emp}(\mathbf{w})$ and h. In addition, any other solution, generated by any other method, that is not Pareto-optimal can still be minimized in both objectives. According to this Optimization definition of the problem, model selection for ANNs should be accomplished amongst the Pareto set solutions of $R_{emp}(\mathbf{w})$ and

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 377–384, 2011.

h. The availability of the Pareto-set solutions guarantee that the search is accomplished amongst the optimal solutions only, and not in the whole parameter's space.

A formulation of the Multi-objective learning problem, based on the ϵ-constrained optimization method [2,4] so that an approximation of the Pareto-front is achieved has been presented in the literature. In this work we propose a new method to approximate the Pareto-front solutions. The method first adjusts the model capacity to a target value, h_t, and then decomposes the gradient descent of the empirical risk into two components: one that increases the model capacity and one that decreases the model capacity. These two components are used separately to adjust the parameters of the approximation function, $f(\mathbf{x}, \mathbf{w})$. By doing so, the procedure minimizes the empirical risk subject to the target value of the model capacity.

The structure of the present paper is as follows. A review on model capacity and learning is presented in section 2. Section 3 presents the method of the gradient descent decomposition. In section 4, simulated results are presented. Discussion and Conclusion in section 5 ends the paper.

2 Model Capacity and Learning

The notion that both error and model capacity should be traded-off in order to reduce approximation error can be well understood from the interpretation of the inequality presented in Expression 1, one of the most important formal results of SLT [1]. The interpretation of such expression is that the true risk $R(\mathbf{w})$ is bounded by the empirical risk $R_{emp}(\mathbf{w})$ plus a square root term that depends both on the sample size N and on the model capacity h.

$$R(\mathbf{w}) \leq R_{emp}(\mathbf{w}) + \sqrt{\left(\frac{h(log(\frac{2N}{h}) + 1) - log(\frac{\eta}{4})}{N}\right)} \tag{1}$$

where N is the sample size, h is the VC-dimension [1] and η is the confidence interval.

It is clear from Expression 1 that, when the sample size is large, the true risk is bounded by the empirical risk only, since as $N \rightarrow \infty \Longrightarrow R(\mathbf{w}) \leq R_{emp}(\mathbf{w})$. This means that convergence to the true risk in such a situation is yielded by simply minimizing the empirical risk ($R_{emp}(\mathbf{w}) \approx 0 \Longrightarrow R(\mathbf{w}) \approx 0$ when $N \rightarrow \infty$). However, the real challenge for learning is to deal with small samples problems. In such a scenario, $\frac{h}{N}$ is relevant, and a small $R_{emp}(\mathbf{w})$ does not imply on small $R(\mathbf{w})$. Therefore, for small samples, reducing the empirical risk only does not guarantee a small approximation error. In other words, both the empirical risk $R_{emp}(\mathbf{w})$ and the model capacity h should be minimized in order to reduce the approximation error.

It is clear, therefore, that a single-objetive minimization of one of the objective functions may result on the maximization of the other, what paves the way to the Multi-Objective [3] treatment of the problem [6,2]. From this perspective, the general problem of model induction (Supervised or Semi-Supervised) can be regarded as the trade-off between $R_{emp}(\mathbf{w})$ and h.

The most usual representation for $R_{emp}(\mathbf{w})$ is the squared error loss function $\sum e^2 = \sum_{i=1}^{N}(y_i - f(\mathbf{x}_i, \mathbf{w}))^2$, whereas h is usually represented by the L_2 norm, $||\mathbf{w}|| = \sqrt{\sum w_j^2}$, of the network weights. Although the squared error representation of $R_{emp}(\mathbf{w})$ is well accepted, the choice of $||\mathbf{w}||$ may deserve an explanation. The first argument in favor of $||\mathbf{w}||$ is that, for better generalization, it has been shown in the literature that the magnitude of the weights is more important than the number of weights [7] and, of course, the magnitude of the weights can be directly controlled by $||\mathbf{w}||$. The second argument is that the class separation margin of two classes is inversely proportional to $||\mathbf{w}||$ [1,8] what implies that margin maximization is yielded by $||\mathbf{w}||$ minimization.

The two objective-functions $\phi_e(\cdot) = \sum e^2$ and $\phi_{\mathbf{w}}(\cdot) = ||\mathbf{w}||$ are conflicting for the simple reason that error reduction requires weight magnitude increase in order to reach the saturation regions of the sigmoidal activation functions. If they were not conflicting, a trivial solution would be characterized, since a single weight vector would minimize both objectives. The Pareto set [3] corresponds to the region of the objective's space where the non-dominated solutions are located. From the optimization point of view, the Pareto set contains *the optimal solutions* of the formulated optimization problem. From this perspective, any method that is not Pareto-set-based is likely to generate sub-optimal solutions that could still be minimized in both objectives. Further discussions and results related to the MOBJ principles were presented in other related works [2,4]. Since there isn't a single solution in the Pareto set, the learning problem now, after the Pareto set has been generated, becomes the selection of one of them.

2.1 Pareto Set Generation

In order to obtain the Pareto set efficient solutions for the Multi-layer Perceptron (MLP) learning problem, the ϵ-constraint algorithm [9] has been adopted [2]. Basically, the ϵ-constraint algorithm [9] turns the multi-objective problem into a constrained single-objective problem in order to generate one solution of the Pareto set at a time. From Equation 2, a discrete set of solutions is generated using the ellipsoid algorithm [10] for every ϵ_r constraint value.

$$\begin{array}{c} minimize \ \sum e^2 \\ subject\ to \ ||\mathbf{w}|| \leq \epsilon_r \end{array} \tag{2}$$

for all $r = 1, 2, ..., m$, where $\mathbf{w} \in S$, S is the feasible region and $\epsilon_j \leq \epsilon_k$, $\forall j \leq k$.

The analogy between the constraint yielded in the objective's space by limiting the upper bound of $||\mathbf{w}||$ can be best understood by observing Figure 1. Since the error is minimized within the disc with radius $||\mathbf{w}|| \leq \epsilon_j$, a Pareto set solution is generated for each disc. The optimization problem is then solved for pre-established values of ϵ_j so that a portion of the Pareto set is generated. Once the set of Pareto set solutions is obtained, a decision making procedure picks-up

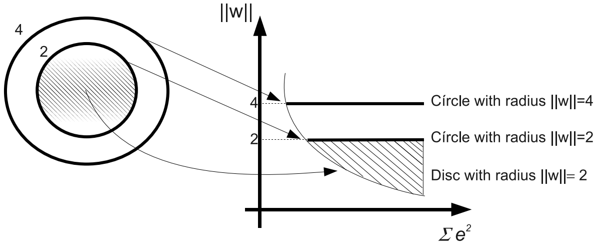

||w||
4
2
4
2
Círcle with radius ||w||=4
Círcle with radius ||w||=2
Disc with radius ||w||= 2
Σe²

Fig. 1. Mapping from parameter's space to objective's space by constraining the solutions to the region $||\mathbf{w}|| \leq \epsilon$

one of them according to a selection criteria. The method proposed here aims at minimizing the error within a single circle.

Although other selection criteria have been developed [11,12], the use of a validation set error seems to be a natural principle. It is based on a validation set $D_v = \{\mathbf{x}_i, d_i\}_{i=1}^{N_v}$ that is presented to every MLP within the Pareto set in the decision phase. Other strategies for selecting the final solution from the Pareto set have also been developed [13,14].

3 Proposed Gradient Descent Decomposition

The proposed method for the adjustment of network weights considers the optimization problem described in Equation 2. It is based on the *backpropagation* gradient descent update equation [15],

$$\mathbf{w}_{(k+1)} = \mathbf{w}_{(k)} - \alpha \cdot \mathbf{g}_{(k)} \tag{3}$$

where $\mathbf{g}$ is the gradient vector, $g_j = \frac{\partial \sum e^2}{\partial w_j}$, α is the learning rate and k represents the algorithm's iterations. The proposed method sequentially updates the network weights by applying a decomposition of the gradient descent vector. However, it requires that the norm of the weights have been initialized with the constraint value, $||\mathbf{w}_{(0)}|| = \epsilon_r$. This can be accomplished by normalizing an initial vector of weights and then multiplying it by ϵ_r, as shown in Equation 4.

$$||\mathbf{w}_{(0)}|| = \epsilon_r \cdot \frac{\mathbf{w}}{||\mathbf{w}||} \tag{4}$$

3.1 Analysis of the Gradient Descent Decomposition

The squared norm of the weights at the $k+1$ step using Equation 3 can be written as

$$||\mathbf{w}_{(k+1)}||^2 = \sum_j w_{j(k)}^2 - 2.\alpha. \sum_j \left(w_{j(k)} . g_{j(k)}\right) + \alpha^2 . \sum_j g_{j(k)}^2 \tag{5}$$

As can be seen in Equation 5 the norm of the weights decreases when $w_{j(k)}.g_{j(k)} > 0$ and it increases, otherwise. Therefore, the gradient descent vector can be decomposed into the vector $\mathbf{g}^-_{(k)}$,

$$g^-_{j(k)} = \begin{cases} g_{j(k)}, & \text{if } w_{j(k)}.g_{j(k)} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

and the vector $\mathbf{g}^+_{(k)} : \mathbf{g}^+_{(k)} + \mathbf{g}^-_{(k)} = \mathbf{g}_{(k)}$.

The proposed algorithm iteratively updates the network weights using two main steps.

Step 1: Decrease the norm of the weights, $\mathbf{w}_{(k+1)} = \mathbf{w}_{(k)} - \alpha \cdot \mathbf{g}^-_{(k)}$.

Step 2: Restore the norm of the weights to its constrained value ϵ_r using an adjusted learning rate, $\mathbf{w}_{(k+1)} = \mathbf{w}_{(k)} - \alpha_+ \cdot \mathbf{g}^+_{(k)}$.

The adjusted learning rate is calculated assuming that $||\mathbf{w}_{(k+1)} - \alpha_+ \cdot \mathbf{g}^+_{(k)}||^2 = \epsilon_r^2$.

In general, solutions with reduced error have greater values for the norm [15]. Namely, it is easier to reduce the error by increasing the norm than by decreasing it. Thus, achieving exactly the ϵ_r value is not critical. As a consequence, it can be assumed that

$$||\mathbf{w}_{(k+1)} - \alpha_+ \cdot \mathbf{g}^-_{(k)}||^2 \leq \epsilon_r^2 \tag{7}$$

From Expression 7 the following condition is written

$$\alpha_+^2 \cdot \sum_j (g_j^+)^2 + \alpha_+ \cdot \left[-2. \sum_j w_j . g_j^+ \right] + \left[\sum_j w_j^2 - \epsilon_r^2 \right] \leq 0 \tag{8}$$

Equation 8 represents a second order equation for α_+. Furthermore, it provides boundaries for the learning rate. It can be assumed that first, $\alpha_+ > 0$ and second, the larger positive root of Equation 8 is the upper bound for the learning rate. If no positive roots are available then the weights are not updated.

Empirical results show that increasing the norm of the weights provides larger error reduction than decreasing it. This empirical evidence can be used to improve the error optimization. The constrained value ϵ_r can be multiplied by a factor of 1.01 (1%) or 1.02 (2%), that is, replace ϵ_r with $1.01 \cdot \epsilon_r$ or $1.02 \cdot \epsilon_r$ in Equations 7 and 8.

It is important to notice that the learning rates α and α_+ are related to the decrease and to the increase of the norm of the weights, respectively. However, proper values must be chosen in order to decrease the error function too. This problem can be stated as $\sum e^2_{(k+1)} < \sum e^2_{(k)}$, or as the following:

$$\sum \left[y_i - f(\mathbf{x_i}, \mathbf{w}_{(k)} - \alpha \cdot \mathbf{g}_{(k)}) \right]^2 < \sum \left[y_i - f(\mathbf{x_i}, \mathbf{w}_{(k)}) \right]^2 \tag{9}$$

In this case, the convergence analysis of the error and consequently, convergence boundaries for α requires the specification of the approximation function, $f(\mathbf{x}, \mathbf{w})$.

If $f(\mathbf{x}, \mathbf{w})$ is a linear function then Equation 9 provides boundaries for the learning rates. The work presented in [5] applies a first-order Taylor expansion to a multi-layer-perceptron in order to locally linearize the activation functions of the hidden layer. Following [5], an upper bound for the learning rate based on the Taylor expansion is

$$\alpha \leq \min_i \left\{ \xi / \left| \sum_j^{n+1} \mathbf{x}_j \cdot g1_{ji(k)} \cdot w1_{ji(k)} \right| \right\} \tag{10}$$

where $w1_{ji}$ is the weight of the i-th neuron in the hidden layer, $g1_{ji}$ represents its error derivative, $\mathbf{x}_j$ is the input vector and ξ is a parameter associated to the accuracy of the Taylor expansion. Proposed values for ξ are $0.1 \leq \xi \leq 1.0$.

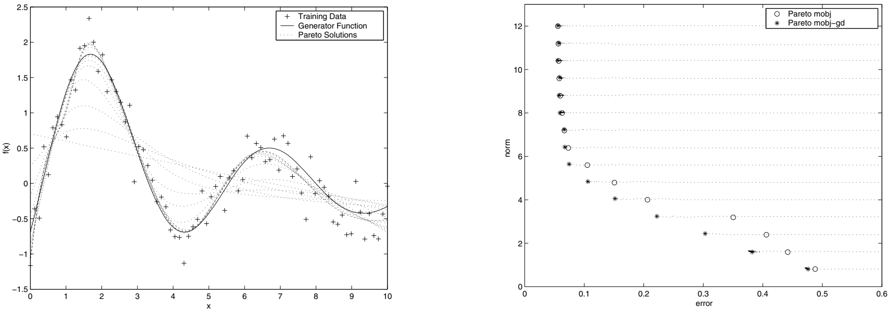
Equation 10 generally estimates smaller values for α. Currently, we are investigating formal convergence conditions for α. An empirical approach is to select one value for the learning rate and test whether it minimizes the error throught the iterations of the proposed algorithm or not.

4 Results

The use of approximated Pareto set solutions for modelling real data sets are presented in [2,5,6,4,11,12,13,16] and elsewhere. Nevertheless, the main concern in multi-objective training is to provide accurate Pareto set solutions. That is, for a particular value of the norm the method achieves the minimum error. Therefore, the following results show the ability of the proposed method in generating an accurate set of approximated solution of the Pareto set.

Figure 2 shows Pareto set solutions generated using the proposed method, named *mobj-gd* (*multi-objective using gradient decomposition*), and using the *mobj* (multi-objective) method proposed in [2]. The approximation function is a multi-layer-perceptron with one linear output neuron and one hidden layer with 10 neurons and hyperbolic activation functions. Figure 2 (a) illustrates that as the norm of the weights increases the approximation function fits better the data set. Figure 2 (b) shows the Pareto set solutions and the trajectories generated by the proposed method. As can be seen, the *mobj-gd* method minimizes the error while the norm of the weights is kept constant during the iterations of the algorithm. Furthermore, the *mobj-gd* method generated more accurate solutions than the *mobj* method for this data set.

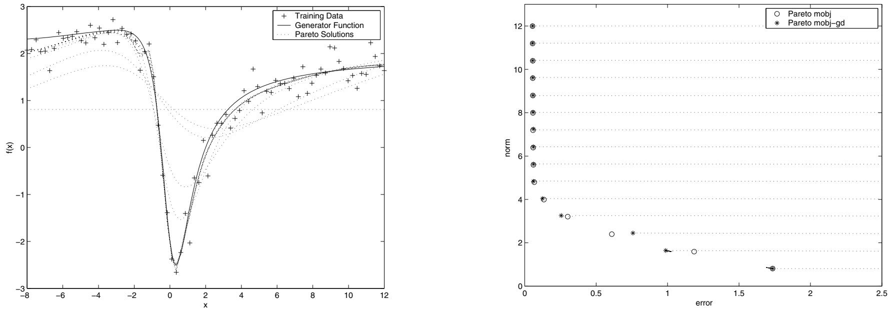
Figure 3 compares *mobj-gd* and *mobj* methods using a more complex generator function. Figure 3 (b) shows that for one particular value of the norm the *mobj* method provided a Pareto solution with better error when compared to the *mobj-gd* solution. This can be improved by setting a larger number of iterations for the proposed algorithm.



(a) Generated solutions

(b) Estimated Pareto set

Fig. 2. Results using the mobj-gd and mobj algorithms for simulated function (1)



(a) Generated solutions

(b) Estimated Pareto set

Fig. 3. Results using the mobj-gd and mobj algorithm for simulated function (2)

5 Discussion and Conclusion

It is well known that Pareto set solutions with small norm value have lower complexity. In practice, multi-layer-perceptron with a small norm behave as a linear model. Comparing Figures 1, 2 (b) and 3 (b) it is evident that the proposed method minimizes the error by following a horizontal trajectory in the objectives' space. Therefore, each line represents an error curve. For small norm values the error curve is approximately convex. As the norm increases the error curve becomes more complex. This feature can be used to design a strategy to generate the Pareto set approximations. Start by minimizing the error subject to a small norm value, then add to the norm constraint a small increment and use the previous weights as a *warm* start. This procedure substantially reduces the effects of local minimum and provides convergence optimization.

One drawback of the proposed method is its low convergence of the error because it is based on the gradient descent. Nevertheless, accurate Pareto set approximations can be obtained as long as proper algorithms for gradient optimization is chosen.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)
2. Teixeira, R.A., Braga, A.P., Takahashi, R.H.C., Saldanha, R.R.: Improving generalization of mlps with multi-objective optimization. Neurocomputing 35, 189–194 (2000)
3. Chankong, V., Haimes, Y.Y.: Multiobjective Decision Making: Theory and Methodology, vol. 8. North-Holland (Elsevier), New York (1983)
4. Braga, A.P., Takahashi, R.H.C., Teixeira, R.A., Costa, M.A.: Multi-Objective Algorithms for Neural Networks Learning. In: Jin, Y. (ed.) Multiobjective Machine Learning, pp. 151–172. Springer, Heidelberg (2006)
5. Costa, M.A., Braga, A.P., Menezes, B.R., Teixeira, R.A., Parma, G.G.: Training neural networks with a multi-objective sliding mode control algorithm. Neurocomputing 51, 467–473 (2003)
6. Liu, G.P., Kadirkamanathan, V.: Learning with multi-objective criteria. In: International Conference on Neural Networks, UK, pp. 53–58 (1995)
7. Bartlett, P.L.: For valid generalization, the size of the weights is more important than the size of the network. In: Proceedings of NIPS, pp. 134–140 (1997)
8. Boser, B., Guyon, I., Vapnik, V.: A Training algorithm for optimal margin classifiers. In: Fifth Annual Workshop on Computational Learning Theory, pp. 144–152. Morgan Kaufmann, San Mateo (1992)
9. Haimes, Y.Y., Lasdon, L.S., Wismer, D.A.: On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. IEEE Transactions on Systems, Man, and Cybernetics 1(3), 296–297 (1971)
10. Bland, R.G., Goldfarb, D., Todd, M.J.: The Ellipsoidal Method: A Survey. Operations Research 29(6), 1039–1091 (1981)
11. Medeiros, T., Braga, A.P.: A new decision strategy in multi-objective training of artificial neural networks. In: European Symposium on Neural Networks (ESANN 2007), pp. 555–560 (2007)
12. Teixeira, R.A., Braga, A.P., Saldanha, R.R., Takahashi, R.H.C., Medeiros, T.H.: The Usage of Golden Section in Calculating the Efficient Solution in Artificial Neural Networks Training by Multi-objective Optimization. In: de Sá, J.M., Alexandre, L.A., Duch, W., Mandic, D.P. (eds.) ICANN 2007. LNCS, vol. 4668, pp. 289–298. Springer, Heidelberg (2007)
13. Kokshenev, I., Braga, A.P.: Complexity bounds and multi-objective learning of radial basis functions. Neurocomputing 71, 1203–1209 (2008)
14. Braga, A.P.: New decision strategies for multi-objective learning (2011) (in preparation)
15. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning Representations by Back-Propagating Errors. Nature 323, 533–536 (1986)
16. Costa, M.A., Braga, A.P., Menezes, B.R.: Improving generalization of MLPs with Sliding Mode Control and the Levenberg-Marquadt algorithm. Neurocomputing 70, 1342–1347 (2007)

P^2LSA and P^2LSA+: Two Paralleled Probabilistic Latent Semantic Analysis Algorithms Based on the MapReduce Model

Yan Jin[1], Yang Gao[1,*], Yinghuan Shi[1], Lin Shang[1], Ruili Wang[2],
and Yubin Yang[1]

[1] State Key Laboratory for Novel Software Technology, Nanjing University,
Nanjing 210093, China
[2] School of Engineering and Advanced Technology Massey University Palmerston
North, New Zealand
jyannju@gmail.com, gaoy@nju.edu.cn, Yinghuan.shi@gmail.com,
Shanglin@nju.edu.cn, R.Wang@massey.ac.nz, yangyubin@nju.edu.cn

Abstract. Two novel paralleled Probabilistic Latent Semantic Analysis (PLSA) algorithms based on the MapReduce model are proposed, which are P^2LSA and P^2LSA+, respectively. When dealing with a large-scale data set, P^2LSA and P^2LSA+ can improve the computing speed with the Hadoop platform. The Expectation-Maximization (EM) algorithm is often used in the traditional PLSA method to estimate two hidden parameter vectors, while the parallel PLSA is to implement the EM algorithm in parallel. The EM algorithm includes two steps: E-step and M-step. In P^2LSA, the Map function is adopted to perform the E-step and the Reduce function is adopted to perform the M-step. However, all the intermediate results computed in the E-step need to be sent to the M-step. Transferring a large amount of data between the E-step and the M-step increases the burden on the network and the overall running time. Different from P^2LSA, the Map function in P^2LSA+ performs the E-step and M-step simultaneously. Therefore, the data transferred between the E-step and M-step is reduced and the performance is improved. Experiments are conducted to evaluate the performances of P^2LSA and P^2LSA+. The data set includes 20000 users and 10927 goods. The speedup curves show that the overall running time decrease as the number of computing nodes increases.Also, the overall running time demonstrates that P^2LSA+ is about 3 times faster than P^2LSA.

Keywords: Paralleled PLSA, PLSA, MapReduce.

1 Introduction

Probabilistic latent semantic analysis (PLSA) is a statistical technique that can be used to analyze the two-mode and co-occurrence data [1].For a small data set, it is a powerful algorithm to infer the underlying probabilistic relationships

[*] Corresponding author.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 385–393, 2011.

behind the data [2]. However, for a large scale data set, it is difficult to apply PLSA directly due to the time complexity and space complexity of PLSA is proportional to the product of data set and the number of hidden parameters [3]. For a large data set, the overall running time might increase drastically and it is difficult to fit the growing data set to the main memory of a computer.

Many parallel algorithms have been proposed to reduce the time complexity. Mamitsuka has proposed a parallel algorithm, which is based on multi-computing nodes [4]. Zheng *et al.* uses multi-core with all process units reading and processing the data into a same array [5]. However, all of these algorithms do not solve the problem of too large space complexity.

This paper uses MapReduce [6] to reduce the space complexity,and we propose two parallel PLSA approaches based on MapReduce called P^{2}LSA and P^{2}LSA+. They perform the EM which is the core algorithm of PLSA [7]. In P^{2}LSA, the Map function infers all the posterior probability variable Q, and the Reduce function of P^{2}LSA calculates two hidden parameters using the results of the previous step. Different from P^{2}LSA, P^{2}LSA+ uses two different jobs to finish the whole task. Each job is to estimate one of these two parameters separately. P^{2}LSA+ alleviates the burden of the Reduce function and decreases the data transfer between the E-step and M-step.

The major contributions of this paper include: (i) using MapReduce to parallel PLSA and solving the time/space complexity problem, and (ii) proposing two parallel algorithms: P^{2}LSA and P^{2}LSA+. P^{2}LSA+ decreases the data transfer between the E-step and M-step, and reduces the overall computing time.

This paper is organized as follows. In Section 2, we firstly introduce the procedures of PLSA. Then the details of P^{2}LSA and P^{2}LSA+ are explained in Section 3. We present our experimental results in Section 4. Finally, the conclusion is given in Section 5.

2 PLSA

PLSA has been widely used in analyzing user preferences by exploring the relationships between a set of users and a set of items. Two domains in PLSA consist a set of users $U = \{u_1, u_2, \ldots, u_n\}$ and a set of items $Y = \{y_1, y_2, \ldots, y_m\}$. We assume the co-occurrence of u and y is available, and the total number of co-occurrence are s. A co-occurrence of u and y represents a event "user u buys item y". For every $u \in U$ and $y \in Y$, the probability $P(y|u)$ is the user preference which needs to be inferred.

2.1 Model Construction and Solution

The main idea of PLSA is to introduce hidden variables Z. For each user-item pairs (u, y), there will be a set of states z for them. Therefore, user u and item y are conditionally independent. The resulting model is a mixture model that can be written as follows:

$$P(y|u;\theta) = \sum_{z=1}^{k} P(y|z)(z|u) \tag{1}$$

The size of the possible value of z is k, and the right side of Eq.(1) is the accumulation of z for all k states. In this model, the parameter vector θ summarizes the probabilities $P(z|u)$ and $P(y|z)$. $P(z|u)$ can be described by $(k-1)*n$ independent parameters, and $P(y|z)$ requires $(m-1)*k$ independent parameters [8].

Our goal is to find the best model parameter θ by maximizing the conditional log-likelihood estimation using Eq.(2)

$$R(\theta) = -\frac{1}{N}\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} log P(y_j|u_i;\theta) \tag{2}$$

In Eq.(2), a_{ij} is the occurrence of every (u_i, y_j). N is the total number of occurrence of every pairs.

2.2 EM Algorithm

EM is an iterative algorithm which can be used to infer parameter vector θ in PLSA[8]. Every iteration of EM contains the following two steps: E-step and M-step. The E-step is used to compute the posterior probabilities Q of the hidden variables according to Eq.(3) [9].

$$Q^*(z;u,y;\theta) = P(z|u,y;\theta) = \frac{\hat{P}(y|z)\hat{P}(z|u)}{\sum_{z'=1}^{k}\hat{P}(y|z')\hat{P}(z'|u)} \tag{3}$$

Obviously, the posterior probabilities need only to be computed for user-item pairs (u, y) that have actually been observed. In the M-step, variation distribution Q obtained from the E-step is used to compute the following parameter vector using Eq.(4) and Eq.(5)

$$P(y|z) = \frac{\sum_{u=1}^{n} Q^*(z;u,y;\theta)}{\sum_{y'=1}^{m}\sum_{u=1}^{n} Q^*(z;u,y';\theta)} \tag{4}$$

$$P(z|u) = \frac{\sum_{y=1}^{m} Q^*(z;u,y;\theta)}{\sum_{z'=1}^{k}\sum_{y=1}^{m} Q^*(z';u,y;\theta)} \tag{5}$$

The E-step and M-step can be integrated as one step. We can use Eq.(6)(7) (8)(9) to update $P(y|z)$ and $P(z|u)$ before converge,and $x(u, y)$ co-occurrence of u and y.

$$P(y|z) \leftarrow P(y|z)\sum_{u'=1}^{n}\frac{x(u',y)}{\sum_{z'=1}^{k} P(y|z')P(z'|u')} P(z|u') \tag{6}$$

$$P(y|z) \leftarrow \frac{P(y|z)}{\sum_{y'}^{m} P(y'|z)} \tag{7}$$

$$P(z|u) \leftarrow P(z|u) \sum_{y'=1}^{m} \frac{x(u,y')}{\sum_{z'=1}^{k} P(y_{'}|z')P(z'|u)} P(y_{'}|z) \tag{8}$$

$$P(z|u) \leftarrow \frac{P(z|u)}{\sum_{z'}^{k} P(z'|u)} \tag{9}$$

3 Parallel PLSA

For a large-scale data set, it is difficult to apply the traditional PLSA directly. The reason is that a computer has to read the model file and the input data set into its main memory. Thus, when the memory size is limited and smaller than that required by the data set, applying the traditional PLSA directly can cause some problems. MapReduce can be used to solve the memory problem in parallel. The whole input can be partitioned into smaller chunks, and the model file can be read line by line. If we can parallel PLSA, the data set that need to be read into the main memory will reduce greatly. Thus, the size of an input data set can be big. In the following section, two PLSA parallelizing approaches based on MapReduce will be introduced, which are named as P^{2}LSA and P^{2}LSA+.

3.1 P^{2}LSA

P^{2}LSA accomplishes the E-step in the Map function and uses the Reduce function to accomplish the M-step.

Data file. The data files has three model files, which are $X(U,Y)$, $P(Z|U)$ and $P(Y|Z)$. The last two files should be manually initialized by other programs before the first iteration. These data are recalculated in one iteration and the calculated results are used as the input files of the next iteration. This process is shown in Figure 1.

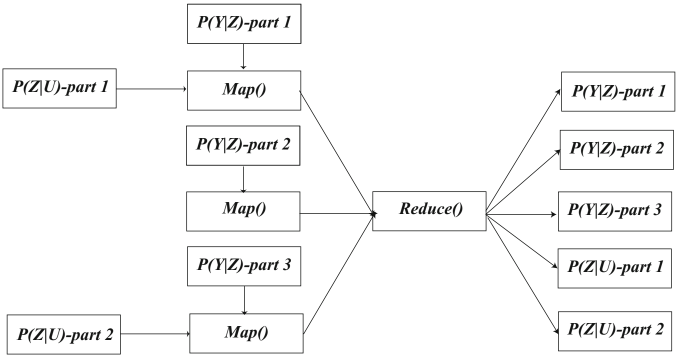


Fig. 1. The iteration process of P^{2}LSA

Mapper. The Setup function is the function executed before initialization. It reads the entire model file x into the global array $Model$. The Map function reads the input file which corresponds to its Mapper, and stores it to $A(Y)$. The cleanup function is called at the end of Mapper, which accomplishes the E-step of EM. For each u in $P(Z|U)$ and each y in $A(Y)$, if $Model[u][y]$ is not equal to 0, compute all the Q for one (u, y) pair using Eq.(3). Then we shall send three key/value pairs (A, Q), (B, Q), (C, Q) to Reducer. $A.first$ equals $'type1'$; $A.second$ equals $'max'$; $B.first$ equals $'type1'$; $B.second$ equals $'y'$; $C.first$ equals $'type2'$; $A.second$ equals $'u'$.

In P^{2}LSA and P^{2}LSA+, the Compare function of the GourpComparator function and KeyComparator function need to be rewritten. When the Reducer deals with the key/value pairs with the same key.first, the one whose key.second equals 'max' will be in front of others [10].

Table 1. Pseudo-code to compute $P(y|z)$ in P^{2}LSA

```
– Input: key/value pairs received by reducer
– Output: $tmp, v$
– For $i : k$
    • $sum[i] \leftarrow 0$; $part\_sum[i] \leftarrow 0$
– flag ← 'max'
– Repeat for all key/value pairs
    • case: key.second $\doteq$ 'max'
        ∗ For $i : k$
            · $sum[i] \leftarrow sum[i] + value.get[i]$ % accumulate the global value
    • case: key.second $\neq$ 'max' and key.second $\neq$ flag
        ∗ For $i : k$
            · add $(part\_sum[i]/sum[i])$ to $v$
            · $part\_sum[i] \leftarrow 0$
        ∗ flag ← key.second
        ∗ $tmp.first$ ← flag; $tmp.second$ ← 'pzy'
        ∗ context.write($tmp, v$)
    • case: key.second $\doteq$ flag
        ∗ For $i : k$
            · $part\_sum[i] \leftarrow part\_sum[i] + value.get[i]$ % accumulate the part sum
              of current $y$
```

Reducer. In the Reduce function, if key.first equals $'type1'$, it means that the value is to compute $P(y|z)$. Those key/value pairs, whose key.second is $'max'$, cumulate the value and the result is the global accumulation that is the underside of $P(y|z)$ in Eq.(4). If key.second is not $'max'$, the code sets a flag that equals current y. When a new key/value pair comes, if key.second of that pair equals the flag, the flag does not change. The code cumulates the value to the part sum of current y. If key.second of that pair is different from the flag, the flag changes to the key.second of the pair. The code uses the cumulated part sum of the last y to divide the global accumulation. The key of the final output is y and the value is all $P(y|z)$ correspond to y.The pseudo-code can be referred to Table 1: If the key.first equals $'type2'$, it means that the value is to compute $P(y|z)$. The code sets a flag that equals current u. When a new key/value pair comes, if key.second of that pair equals current u, the flag does not change and the code cumulates the value to the part sum of that u. If the flag changes, the code uses the part sum with regard to every z to divide the sum of them, the result final output value.The pseudo-code can be referred to Table 2:

Table 2. Pseudo-code to compute $P(z|u)$ in P^{2}LSA

```
– Input: key/value pairs received by reducer
– Output: $tmp, v$
– For $i : k$
    • $part\_sum[i] \leftarrow 0$
– flag ← 'max'
– Repeat for all key/value pairs
    • If key.second $\neq$ 'flag'
        ∗ $sum \leftarrow 0$
        ∗ For $i : k$
            · $sum \leftarrow part\_sum[i] + sum$
        ∗ For $i : k$
            · add $(part\_sum[i]/sum[i])$ to $v$
            · $tmp.first \leftarrow$ flag; $tmp.second \leftarrow$ 'puz'
            · context.write$(tmp, v)$
    • else
        ∗ For $i : k$
            · $part\_sum[i] \leftarrow part\_sum[i] + value.get[i]$ % accumulate the part sum
              of current u
```

3.2 P^{2}LSA+

It can be inferred from the pseudo-code that P^{2}LSA has to send the Q value three times. This method increases the burden of Reducer. Besides, the size of Q is $'s * k'$, transferring a large amount of data between the E-step and M-step slows down the speed of operation. P^{2}LSA+ uses the Eq.(6)(7)(8)(9) to transform the procedures of the E-step and M-step into one iteration equation, and infers $P(z|u)$ and $P(y|z)$ in two different jobs. The whole process of one iteration is to alternatively run both jobs: Job1 and Job2.

Job1. Job1 is responsible for computing $P(Y|Z)$. The data files first include model file $X(U, Y)$, $P(Z|U)$ and an input file of Mapper $P(Y|Z)$.

In Mapper, the Setup function can be used to read and store model files $X(U,Y)$ and $P(Z|U)$ before initialization. In the Map function, the code first reads $P(Y|Z)$ of one y each time. And according to Eq.(6), the code calculates a new $P(Y|Z)$ of this y. The inputs of the Reducer are two key/value pairs. In the first key/value pairs, key.first is $'FK'$ and key.second is $'max'$. In the second key/value pairs, key.first is FK and key.second is the corresponding y. The values of these two pairs are identical, which include a new $P(Y|Z)$ with reference to this y.

In Reducer, if key.second equals $'max'$, we shall respectively cumulative all k parts of the value, and store the result in $sum[k]$. On the other condition, we need to use the k part of the value to divide the corresponding part in $sum[k]$, the result is the final output.

The final output will be divided into small files and stored in a specific folder named $P(Z|U)$. These files are used for the next iteration as the inputs of Job2.

Job2. The task of Job2 is to compute $P(Z|U)$. In Mapper, the Setup function can be used to read and store model file $X(U,Y)$ and $P(Y|Z)$ before initialization. The Map function reads $P(Z|U)$ of one u each time. And according to Eq.(8), the code calculates the new $P(Z|U)$ of this u. Besides, we can use Eq.(9) to normalize the result and send it to Reducer.

Reducer only needs to output the key/value pairs received from Mapper.

4 Experiment Result

Our experimental environment is cloud computing stage Hadoop. It includes 6 nodes: 1 namenode and 5 datanode. The maximal load of Mapper and Reducer are 3. We test our proposed two algorithms using two data sets. The first one is a MovieLen data set, which contains the ratings for 3883 movies by 6040 users [11]. The second one is a larger MovieLen data set, which includes the ratings for 10297 movies by 20000 users. Two data files are cut into 15 pieces and 13 pieces, respectively. k is 15, and the time of iteration is 15.

Table 3. The running time of P^{2}LSA and P^{2}LSA+

	1-node	2-node	3-node	4-node	5-node
P^{2}LSA in set1	3109s	2908s	2717s	2448s	1988s
P^{2}LSA+ in set1	1766s	1272s	1048s	989s	819s
P^{2}LSA+ in set2	6982s	3892s	3180s	2474s	2245s

Figure 2 illustrates the time changes with the increase of the number of nodes. The x-coordinate is the number of nodes:

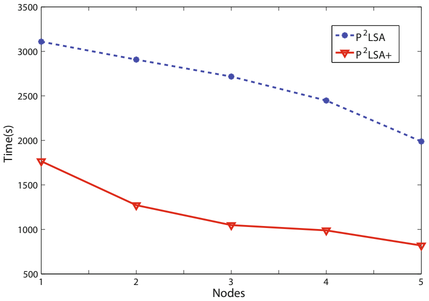


Fig. 2. Acceleration curves in data set 1

The description of P^{2}LSA shows that, in the Setup function, P^{2}LSA reads one model file into the main memory and in the Map function, P^{2}LSA reads the divided input data set to the main memory. On contrast to the traditional PLSA, the utmost files needed to be read into the main memory are decreased. In P^{2}LSA+, the utmost files in the main memory are these two model files. Since the input data are read line by line, the required memory is reduced.

The results shows that, P^{2}LSA+ outperforms P^{2}LSA in both acceleration curve and running time. Because calculation is concentrated in the Map end, and the parallelization of data mostly refers to the parallelization of the Map side, P^{2}LSA+ makes a good use of the environment and the platform of Hadoop. P^{2}LSA+ reduces the amount of computation, and makes the calculation in the Reducer side much simpler, thereby alleviates the bottleneck. The acceleration between three nodes and four nodes is not as better as the situation between one node and two nodes. This is because the number of the running mappers is 15. Three nodes can run 9 tasks at the same time, and four nodes can run 12 tasks at the same time. Therefore, they need to run two rounds to finish one iteration. With the same round number, the only difference between these two situations is the computing time of the second round. Therefore, the difference of the overall running times is not obvious. Besides, Table 6 shows that, the speed of data increasing is more than that of computing time. It is because that except for the fixed time consumption, the increased data set only increases the time in loading the data file and the model files.

5 Conclusion

This paper proposes two paralleled implementations of PLSA, named P^{2}LSA and P^{2}LSA+. After introducing these two algorithms, we compare these two algorithms on two different data sets and give some related analysis. The results show that thsee two algorithms work well on large-scale data sets; the overall time decreases as the computing nodes increases. Besides, the time curve shows that P^{2}LSA+ is better in terms of a shorter running time.

These two algorithms still have some shortcomings. If the input data set exceeds a certain level, they may not work well. We will solve this problem in the future.

Acknowledgments. We would like to acknowledge the support for this project from the National Science Foundation of China (Grant Nos.61035003, 60721002), the Major National Science and Technology Project (Grant No. 2010DFA11030, 2009ZX01043-001-06), the National 973 Program of China (Grant No. 2009CB320700) and Program for New Century Excellent Talents in University (Grant No. NCET-10-0476).

References

1. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57. ACM, New York (1999)
2. Kong, S.Y., Shan Lee, L.: Improved spoken document summarization using probabilistic latent semantic analysis. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2006, vol. I, pp. 941–944 (2006)
3. Newman, D.J., Block, S.: Probabilistic topic decomposition of an eighteenth-century american newspaper. J. Am. Soc. Inf. Sci. Technol. 57(15), 753–767 (2006)
4. Wan, R., Anh, V.N., Mamitsuka, H.: Efficient probabilistic latent semantic analysis through parallelization. In: Lee, G.G., Song, D., Lin, C.-Y., Aizawa, A., Kuriyama, K., Yoshioka, M., Sakai, T. (eds.) AIRS 2009. LNCS, vol. 5839, pp. 432–443. Springer, Heidelberg (2009)
5. Hong, C., Chen, W., Zheng, W., Shan, J., Chen, Y., Zhang, Y.: Parallelization and characterization of probabilistic latent semantic analysis. In: Proc. 37th International Conference on Parallel Processing, pp. 628–635 (2008)
6. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters, pp. 10–10. USENIX Association (2004)
7. Jin, X., Zhou, Y., Mobasher, B.: Web usage mining based on probabilistic latent semantic analysis. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 197–205. ACM Press, New York (2004)
8. Hofmann, T.: Latent semantic models for collaborative filtering. ACM Trans. Inf. Syst. 22(27), 89–115 (2007)
9. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society. Series B 39(1), 1–38 (1977)
10. White, T.: Hadoop: The Definitive Guide. O'Reilly Media, Inc., Sebastopol (2009)
11. MovieLens: Movielens datasets of the university of minnesota, http://www.movielen.org

Purging False Negatives in Cancer Diagnosis Using Incremental Active Learning

Catarina Silva[1,2] and Bernardete Ribeiro[2]

[1] School of Technology and Management, Polytechnic Institute of Leiria, Portugal
[2] Department of Informatics Engineering, Center for Informatics and Systems of the University of Coimbra (CISUC), Portugal
{catarina,bribeiro}@dei.uc.pt

Abstract. Cancer is becoming a human plague, and decision-support tools to help physicians better diagnosing are a fulsome research field. False negatives can be a huge problem for cancer diagnosticians, since while a false positive can result in time and money lost, a false negative can result in the lost of human lives, which puts an overwhelming burden on diagnosis.

In this framework, we propose a two-fold approach to purge false negatives in cancer diagnosis without compromising precision performance. First, we use an incremental background knowledge method and then, an active learning strategy completes the procedure. The defined incremental active learning SVM method was tested in the Wisconsin-Madison breast cancer diagnosis problem showing the effectiveness of such techniques in supporting cancer diagnosis.

1 Introduction

This paper proposes an approach for purging false negatives in cancer diagnosis, using incremental background knowledge and active learning in the setting of the diagnosis of breast cancer using a well-known benchmark [1].

1.1 Introduction to Breast Cancer Diagnosis and Related Work with Machine Learning Algorithms

Breast cancer is the most common cancer among women, and accounts for more than one in four of all cancer cases in women. It is the second leading cause of cancer death in women, after lung cancer, and the lifetime risk for breast cancer in women is 1 in 9 [2]. Treatment includes several techniques from surgery, radiation therapy and chemotherapy. The problem has significant clinical importance since the therapy to be used depends on the type of patient malignancy. Although several classical methods have been used in the past [3] machine learning methods, constitute doubtless a means to provide an initial aid to physicians in early diagnosis, improving the fastness of medical care follow-up. A significative number of methods have been published employing neural networks (NN) covering aspects from either classification, through diagnosis [4], or prediction where

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 394–402, 2011.

prognosis models are the main goal [3]. Many techniques based on clustering [5], or unsupervised learning, such as the self-organizing maps [6] have been applied in the breast cancer domain, however they have a strict application to prediction prognosis [17]. In recent years, Support Vector Machines (SVM) have spawn a great deal of applications in the biomedical area, in particular in breast cancer diagnosis [18,19]. There is a broad coverage in the literature of a wide range of other intelligent techniques such as fuzzy set theory (FS), Neuro-Fuzzy (NF), decision trees (DT), case-based reasoning (CBR), data envelopment analysis and soft computing [20]. Among the better designed and validated studies it is clear that machine learning methods can be used to substantially (15-25%) improve the accuracy of predicting cancer susceptibility, recurrence and mortality. At a more fundamental level, it is also evident that machine learning is also helping to improve basic understanding of cancer development and progression [20].

Apart from the development of automated learning methods for successful breast carcinoma diagnosis, a common goal is to reduce false negatives. Although the prospect of improving the diagnostic performance in terms of detection accuracy has led to a broad research in the area, little attention has been paid in terms of a focused approach to increase sensitivity in an incremental mode.

1.2 Learning Methods with Unlabeled Data

Most learning methods, e.g., K-Nearest Neighbor, Naïve Bayes, Neural Nets and Support Vector Machines, have their performance greatly defined by the training set available. To achieve the best classification performance with a machine learning technique, there has to be enough labeled data. However, these data are costly and sometimes difficult to gather. This is one key difficulty with current algorithms, since they require manual labeling of more cases than a setting permits [21].

Labeling data is expensive but, unlabeled can often be inexpensive, abundant and readily available. Therefore, to achieve the purpose of using relatively small training sets, the information that can be extracted from the testing set, or even unlabeled examples, is being investigated as a way to improve classification performance [7,8]. Seeger in [9] presents a report on learning with unlabeled data that compares several approaches.

In general, unlabeled examples are much less expensive and easier to gather than labeled ones. Collecting these data can frequently be done semi-automatically, so it is feasible to collect a large set of unlabeled examples. If unlabeled examples can be integrated into supervised learning, then building cancer diagnosis systems can be significantly faster, less expensive and more effective.

There is a catch however, because, at first glance, it might seem that nothing is to be gained from unlabeled data, since unlabeled data do not contain the most important piece of information - the classification.

Consider a generic breast cancer diagnosis problem, where a researcher has access to a small labeled dataset and a large unlabeled number of samples. By looking at just the labeled data, the researcher can determine that patients with a given feature tend to belong to a specific class. If she uses this fact to estimate the classification of the many unlabeled patients, she might find that some other feature, not present in the labeled set, occurs frequently in the patients now classified in that specific class. This co-occurrence of features over the large set of unlabeled training data can provide useful information to construct a more accurate classifier that considers both features as indicators of positive examples of that class.

In this work we extend to a cancer diagnosis framework, the incremental setting of an incremental background knowledge approach previously proposed by the authors in [10]. The Incremental Background Knowledge approach is itself an improvement of the previously proposed Basic Background Knowledge (BBK) in [11,12]. The proposed extention includes an active learning strategy that retrieves the (few) active examples using an SVM margin-based approach.

The rest of the paper is organized as follows. Section 2 introduces Support Vector Machines, setting foundations to Section 3, where the proposed two-fold incremental active learning SVM approach is presented. Sections 4 and 5 expose the experimental setup and results and Section 6 presents some conclusions and future work.

2 Support Vector Machines

SVM are a learning method introduced by Vapnik [13] based on his Statistical Learning Theory and Structural Minimization Principle. When using SVM for classification, the basic idea is to find the optimal separating hyperplane between the positive and negative examples. The optimal hyperplane is defined as the one giving the maximum margin between the training examples that are closest to it. Support vectors are the examples that lie closest to the separating hyperplane. Once this hyperplane is found, new examples can be classified simply by determining on which side of the hyperplane they are. SVM start from a simple linear maximum margin classifier. Given an i.i.d. sample $(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l)$, where $\mathbf{x}_i$ for $i = 1, ..., l$ is a feature vector of length l and $y_i = \{+1, -1\}$ is the class label for $\mathbf{x_i}$, find a classifier with the decision function $f(x)$, such that $y = f(x)$, where y is the class label for x. The performance of the classifier is measured in terms of classification error, as defined in (1).

$$E(y, f(x)) = \begin{cases} 0 & if \quad y = f(x), \\ 1 & otherwise. \end{cases} \tag{1}$$

Learning machines have a set of adjustable parameters, λ. Given the above classification task, the machine will tune its parameters λ to learn the mapping $\mathbf{x} \rightarrow y$. This will result in a mapping $\mathbf{x} \rightarrow f(\mathbf{x}, \lambda)$, which defines the particular learning machine. The performance of this machine can be measured by the expectation of the test error, as shown in (2).

$$R(\lambda) = \int E(y, f(\mathbf{x}, \lambda)) \quad dP(\mathbf{x}, y) \tag{2}$$

This is called the expected risk and requires that at least an estimate of $P(\mathbf{x}, y)$ is known, which is not available for most classification tasks. Thus the empirical risk measure, defined in (3), has to be used.

$$R_{emp} = \frac{1}{l} \sum_{i=1}^{l} E(y, f(\mathbf{x}, \lambda)). \tag{3}$$

The empirical risk provides a measure of the mean error over the available training data and most training algorithms implement its minimization (Empirical Risk Minimization), i.e., minimize the empirical error using maximum likelihood estimation for parameters λ. These conventional training algorithms do not consider the capacity of the learning machine and this can result in over fitting, i.e., using a learning machine with too much capacity for a particular problem.

In contrast with ERM, the goal of SRM [13] is to find the learning machine that yields a good trade-off between low empirical risk and small capacity. There are two major problems in achieving this goal: (i) SRM requires a measure of the capacity of a particular learning machine or, at least, an upper bound on this measure; (ii) an algorithm to select the desired learning machine according to SRM's goal is needed.

3 Proposed Approach

In this section we propose a two-fold approach that makes use of unlabeled data to improve cancer diagnosis performances by purging false negative results. First, we use an incremental background knowledge method and then, an active learning strategy completes the procedure.

3.1 Incremental Background Knowledge

Some authors [14] refer to unlabeled data as background knowledge, defining it as any unlabeled collection of data from any source that is related to the classification task. Joachims presents in [15] a study on transductive SVMs (TSVMs) introduced by Vapnik [13]. TSVMs make use of the testing set and extend inductive SVMs, finding an optimal separating hyperplane not only of the training examples, but also of the testing examples [16].

The Incremental Background Knowledge (IBK) technique we adopt is in fact a development of a Basic Background Knowledge (BBK) approach already proposed by the authors in [11]. In the BBK, first an inductive SVM classifier (see Section 2) is inferred from the training set, and then it is applied to the unlabeled examples. The BBK approach incorporates, in the training set, new examples classified by the SVM with larger margin, which can be assumed as the ones where the SVM classifier presents more confidence. Fig. 1 illustrates an example where four unlabeled examples (black dots) are classified with small and large margins. Formally, the BBK approach proceeds by incorporating unlabeled examples (only the features, not the classification) from the unlabeled/testing set

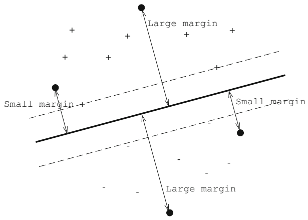


Fig. 1. Unlabeled examples (black dots) with small and large margins

directly into the training set as classified by the baseline inductive SVM, i.e., an example $(\mathbf{x}_i, y_i)$ will be chosen if Equation (4) holds:

$$(\mathbf{x}_i, y_i) : \rho(\mathbf{x}_i, y_i) = \frac{2}{\|w\|} > \Delta, \tag{4}$$

Δ was heuristically defined. Notice that Δ is intrinsically related to the margin, i.e when Δ is decreased, in fact we are decreasing the classification margin of accepted unlabeled examples and thus accepting examples classified with less confidence. This level of confidence should depend on the capabilities of the base classifier, or in other words, the better the base classifier the lower we can set the threshold on Δ (and thus on the margin) to introduce newly classified unlabeled examples into the training set.

In the IBK approach we now adopt for cancer diagnosis, a structural change is proposed to deal with the weaker point of the BBK technique, i.e. the definition of Δ. We proposed the iterative procedure illustrated in Fig. 2. As can be gleaned from this figure, the training set is incrementally constructed by iteratively decreasing the value of Δ, i.e. reducing the confidence threshold for an unlabeled example to be added as classified by the SVM. This approach rational is that as Δ is decreased, the classifiers are also getting better due to the additional information in the training set, thus justifying lowering the confidence threshold. Algorithm 1 below more formally defines the IBK procedure.

Algorithm 1. Incremental Background Knowledge Algorithm.

```
Current training set ← Initial dataset
$\Delta$ ← initial $\Delta$ value
WHILE not all unlabeled examples added
    Infer an SVM classifier with current training set
    Classify unlabeled examples with the classifier
    Select the newly classified examples with margin larger that $\Delta$
    Add the selected examples to the current training set
    Decrease $\Delta$
ENDWHILE
```

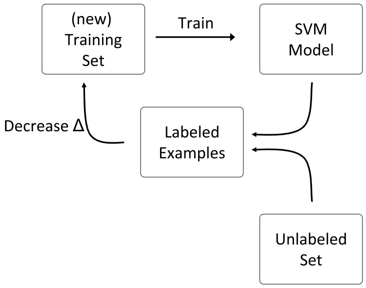


Fig. 2. Proposed approach: Incremental Background Knowledge

3.2 Active Learning

The active learning approach includes a certain number of examples from the testing set (only the features, not the classification) in which the SVM has less confidence (smaller margin, see Figure 1) after they are correctly classified by the supervisor. Thus, an example $(\mathbf{x}_i, y_i)$ will be included if Equation (5) holds.

$$(\mathbf{x}_i, y_i) : \rho(\mathbf{x}_i, y_i) = \frac{2}{\|w\|} < \Delta_2 \tag{5}$$

This number of examples can not be large, since the supervisor will be asked to manually classify them. After being correctly classified, they are integrated in the training set. This approach can be regarded as a form of active learning, where the information that an example can introduce in the classification task is considered inversely proportional to its classification margin.

4 Experimental Setup

The widely used Wisconsin Diagnosis Breast Cancer (WDBC) dataset provided by the University of Wisconsin Hospital (ftp.cs.wisc.edu) was used in the experiments. It was derived from a group of images using fine needle aspiration (FNA) of the breast, also referred as biopsies. Features are computed from digitized images of the FNAs. They describe characteristics of the cell nuclei present in the image.

The dataset includes 32 features, the first is an ID number and the second is the class of the case: M = malignant, B = benign. Then ten real-valued features are computed for each cell nucleus:

1. radius (mean of distances from center to points on the perimeter)
2. texture (standard deviation of gray-scale values)
3. perimeter
4. area

5. smoothness (local variation in radius lengths)
6. compactness ($perimeter^2/area - 1.0$)
7. concavity (severity of concave portions of the contour)
8. concave points (number of concave portions of the contour)
9. symmetry
10. fractal dimension ($coastline\ approximation{-}1$)

The mean, standard error, and *worst* or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is the mean radius, field 13 is Radius standard error, and field 23 is the worst Radius. All feature values are recoded with four significant digits. The dataset has no missing values in its 569 instances, which are divided in 357 benign and 212 malignant. In the experiments, the dataset was randomly divided in 285 examples for training and 284 examples for testing.

In order to evaluate a binary decision task we first define a contingency matrix representing the possible outcomes of the classification: true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). Several measures have been defined based on these values, such as, accuracy ($\frac{TP+TN}{TP+TN+FP+FN}$), recall ($\frac{TP}{TP+FN}$), and precision ($\frac{TP}{TP+FP}$), as well as combined measures, such as, F_β measure or precision-recall breakeven point, which combines recall and precision in single scores. In the experiments we compute accuracy, precision and recall measures.

5 Experimental Results

For Incremental Background Knowledge, experiments were carried out varying the values of Δ starting with no inclusion of new unlabeled examples. i.e high values of Δ corresponding to a baseline setting. Then, Δ was stepwisely decreased allowing the introduction of new recently labeled data. The training set in each iteration, corresponding to a value of Δ, was used as baseline for the next iteration, where it was again incremented. Then, an active learning technique was applied to enhance performance results. Table II presents the performances for background Knowledge (BK), Incremental Background Knowledge (IBK) and Incremental Background Knowledge with Active Learning (IBKAL). IBKAL performances surpass all previous approaches, and exhibits an outstanding behaviour for recall values with no false negative classifications.

Table 1. Comparison of performance results

	ACC	PREC	REC	F1
BK	95.77%	96.23%	92.73%	94.45%
IBK	96.83%	95.50%	96.36%	95.93%
IBKAL	99.50%	99.07%	100.0%	99.53%

6 Conclusions and Future Work

This paper presented a two-fold approach to introduce unlabeled documents information into the learning procedure: first we use an incremental background knowledge method and then, an active learning strategy completes the procedure. The proposed framework was tested in the Wisconsin-Madison breast cancer diagnosis problem and the results presented in the Section 5 are extremely encouraging to the improvement achieved by both methods.

The absence of false negatives in the test results is the main outcome of the proposed approach. This accomplishment is of great importance when patients are the main concern. Notice that when a patient is wrongly classified, if it is a false positive, only time and money are lost, while facing a false negative can be tragic.

The incremental background knowledge method has the advantage of being completely automated, while the proposed margin-based active learning method has potential to substantially improve performance.

Future work is expected in further validating the strategy in different applications, namely different cancer diagnosis and also prognosis applications.

References

1. Merz, C.J., Murphy, P.M.: UCI repository of machine learning data bases, Irvine, CA (1998), http://www.ics.uci.edu/mlearn/MLRepository.html
2. Wrensch, M., Georgianna Farren, T.C., Flavia Belli, J.B., Clarke, C., Erdmann, C.A., Lee, M., Moghadassi, M., Peskin-Mentzer, R., Quesenberry, C.P., Souders-Mason, V., Spence, L., Suzuki, M., Gould, M.: Risk factors for breast cancer in a population with high incidence rates. Breast Cancer Res. 5, 88–102 (2003)
3. Mangasarian, O., Street, W., Wolberg, W.: Breast cancer diagnosis and prognosis via linear programming. Operations Research 43(4), 570–577 (1995)
4. Fogel, D.B., Wasson, E.C., Boughon, E.M., Porto, V.W., Angeline, P.J.: Linear andneural models for classifying breast masses. IEEE Transactions on Medical Imaging 17(3), 485–488 (1998)
5. Xing, K., Chen, D., Henson, D., Sheng, L.: A clustering-based approach to predict outcome in cancer patients. In: ICMLA, pp. 541–546 (2007)
6. Oprea, A., Strungaru, R., Ungureanu, G.: A Self Organizing Map approach to breast cancer detection. In: International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2008, pp. 3032–3035 (2008)
7. Hong, J., Cho, S.: Incremental Support Vector Machine for Unlabeled Data Classification. In: ICONIP, pp. 1403–1407 (2002)
8. Liu, B., Dai, Y., Li, X., Lee, W., Yu, P.: Building Text Classifiers Using Positive and Unlabeled Examples. In: ICDM, pp. 179–188 (2003)
9. Seeger, M.: Learning with Labeled and Unlabeled Data, Technical Report, Institute for Adaptive and Neural Computation. University of Edinburgh (2001)
10. Silva, C., Ribeiro, B., Lopes, N.: Improving Recall Values in Breast Cancer Diagnosis with Incremental Background Knowledge. In: WCCI 2010 (2010)
11. Silva, C., Ribeiro, B.: On Text-based Mining with Active Learning and Background Knowledge using SVM. Journal of Soft Computing - A Fusion of Foundations, Methodologies and Applications 11(6), 519–530 (2007)

12. Silva, C., Ribeiro, B.: Improving text classification performance with incremental background knowledge. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009. LNCS, vol. 5768, pp. 923–931. Springer, Heidelberg (2009)
13. Vapnik, V.: The Nature of Statistical Learning Theory, 2nd edn. Springer, Heidelberg (1999)
14. Zelikovitz, S., Hirsh, H.: Using LSI for text classification in the presence of background text. In: Tenth International Conference on Information Knowledge Management, pp. 113–118 (2001)
15. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. In: International Conference on Machine Learning, pp. 200–209 (1999)
16. Silva, C., Ribeiro, B.: Labeled and Unlabeled Data in Text Categorization. In: IEEE International Joint Conference on Neural Networks (2004)
17. Alex, G., Monaco, J., Doyle, S., Basavanhally, A., Reddy, A., Seiler, M., Ganesan, S., Bhanot, G., Madabhushi, A.: Towards improved cancer diagnosis and prognosis using analysis of gene expression data and computer aided imaging. Experimental Biology and Medicine 234(8), 860–879 (2009)
18. Ribeiro, B.: Learning Adaptive Kernels for Model Diagnosis. Frontiers in Artificial Intelligence and Applications, vol. 104, pp. 563–571. IOS Press, Amsterdam (2003)
19. Akay, M.: SVM combined with feature selection for breast cancer diagnosis. Expert Systems with Applications, Part 2 36(2), 3240–3247 (2009)
20. Cruz, J., Wishart, D.: Applications of machine learning in cancer prediction and prognosis. Cancer Informatics 2, 59–77 (2006)
21. Schohn, G., Cohn, D.: Less is more: Active Learning with Support Vector Machines. In: International Conference on Machine Learning, pp. 839–846 (2000)
22. Schölkopf, B., Burges, C., Smola, A.: Advances in Kernel Methods - Introduction to Support vector Learning, pp. 1–15. MIT Press, Cambridge (1999)

Classification of Household Devices by Electricity Usage Profiles

Jason Lines[1], Anthony Bagnall[1], Patrick Caiger-Smith[2], and Simon Anderson[2]

[1] School of Computing Sciences
University of East Anglia
Norwich
UK
{j.lines,anthony.bagnall}@uea.ac.uk
http://www.uea.ac.uk/cmp

[2] Green Energy Options
Hardwick
Cambridge
UK
http://www.greenenergyoptions.co.uk

Abstract. This paper investigates how to classify household items such as televisions, kettles and refrigerators based only on their electricity usage profile every 15 minutes over a fixed interval of time. We address this time series classification problem through deriving a set of features that characterise the pattern of usage and the amount of power used when a device is on. We evaluate a wide range of classifiers on both the raw data and the derived feature set using both a daily and weekly usage profile and demonstrate that whilst some devices can be identified with a high degree of accuracy, others are very hard to disambiguate with this granularity of data.

Keywords: time series classification, electricity device classification.

1 Introduction

This paper investigates how to classify household items such as televisions, kettles and refrigerators based only on their electricity usage profile over a fixed interval of time. This research is part of a wider project investigating data mining electricity usage patterns generated by 'smart meters'. Smart meters record and transmit data on electricity usage of a whole home, a specific circuit or even an individual plug.

This project is supported by Cambridge based company Green Energy Options (GEO), who have developed a range of smart metering devices. GEO have conducted a preliminary trial of the technology. Their monitoring devices were installed in 187 homes across East Anglia and the usage of individual devices and total household power consumption was recorded at 15 minute intervals for approximately a year in each household. The resulting data is described in Section 3.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 403–412, 2011.

One of the key components of the UK's strategy to reduce carbon emissions is the national plan to roll out smart metering devices to the 27 million homes in the country within the next decade. The cost of this program has been estimated to be £10 billion [8].

This cost is justified by the commonly cited statistic that smart meters often reduce electricity usage by around 2.5% [2]. This source also states an example where a reduction of 20% is recorded. If this is accurate, smart meters offer a cost effective way of significantly reducing carbon emissions. However, this oft cited statistic has little basis in data and very little is known about the actual effect of smart meters and whether any observed initial reduction can be sustained in the medium or long term. Clearly, the act of collecting power usage data is in itself unlikely to modify long term behaviour; all smart meters are required to have an in-home display that describes usage. Very little is known about how people will react to smart meters and how best to use their output to encourage reduced consumption without a detrimental effect on a household's lifestyle. The success of a smart meter in altering consumer behaviour will thus be strongly influenced by:

1. what information can be extracted from the usage data;
2. how this information is presented to best inform the consumer; and
3. whether the consumer can be encouraged to interact with the device in order to act on this information.

GEO have included a range of features in their devices to help achieve these goals. Whilst our primary concern is how to extract knowledge from the data collected from smart meters, the nature of the models we form from the data are influenced by the second and third factors and thus ultimately help GEO provide the consumer with useful information. For example, one of the secondary uses of a smart meter could be to notify the consumer when a monitored device is malfunctioning or using more power than necessary. This offers the potential for saving the consumer money through reduced consumption and is a good way of demonstrating the utility of the device. A prerequisite for identifying faulty or inefficient behaviour is the classification of the type of device being monitored and a description of normal/efficient behaviour. Whilst it is possible to require the consumer to manually identify every device monitored, it is thought that this level of engagement will be hard to achieve. It is far more consumer friendly to be able to automatically identify a device through its usage profile. Hence we consider the time series classification problem of identifying device type through a daily or weekly demand profile. The main contribution of this paper is to define a new time series classification problem and to evaluate a range of strategies for best solving it.

To our knowledge this problem has not been addressed before. The rest of this paper is structured as follows. Section 2 provides background into time series classification and an overview of the strategies we have adopted for this problem. Section 3 details the trial data used to form the classification problem and the preprocessing steps required. Section 4 presents the results of our experiments. Finally, we conclude with Section 5.

2 Time Series Classification

Suppose we have a set of n time series, $T = \{\mathbf{t_1}, \ldots, \mathbf{t_n}\}$ where each series has m ordered real valued observations $\mathbf{t_i} = \{t_{i1}, \ldots, t_{im}\}$ and a class label c_i (note for simplicity we assume the series are of equal length, but this is not a requirement). Time series classification involves finding a function from the space of possible time series to the space of possible class labels. This differs from traditional classification problems in that the discriminating factors are assumed to be primarily embedded within the auto-correlation structure. All time series data mining relies to some degree on a measure of similarity between series. There are essentially three types of similarity: similarity in time (correlation based); similarity in shape (shape based) and similarity in change (autocorrelation based). A fuller discussion can be found in the literature [4,7]. Section 3 details the trial data used to form the classification and the preprocessing steps required. There are a variety of approaches to time series classification, which can be summarised as follows:

Ignore the time element. If the series are of equal length and interval, it is possible to simply ignore the ordering and treat the problem as a traditional classification problem. This approach puts the onus on the classifier to model the interdependency between the attributes. It is potentially useful when attempting to classify based on correlation, but shape based similarity will be hard to detect and autocorrelation similarity impossible. One problem with this approach is that time series tend to have many features, hence some form of attribute selection or more usually feature creation is often employed.

Specialised similarity measure. Recent data mining research has focused on using specialised similarity measures such as dynamic time warping (DTW) [5] in conjunction with lazy classifiers [4] to capture both correlation based and shape based similarity. DTW is a natural generalisation of using Euclidean distance based methods and is often seen as a means of compensating against slight phase shift rather than capturing phase independent similarity.

Extract bespoke features. The most common approach in the machine learning literature is to derive a set of summary features prior to classification (for example, see [10]). This can include time independent summary measures such as mean, variance, kurtosis and skewness and series characterisations such as slope and runs measures. Clearly, the nature of similarity captured is dependent on the features extracted.

Transform into a different feature space. An alternative approach seen in both the machine learning and data mining literature is to use transformations such as Spectral transforms, Autocorrelation function or Wavelets and classify in the transformed space. The aim of such as transformation is either dimensionality reduction to better approximate Euclidean distance [6] or to allow for the detection of shape based or change based similarity [3].

Construct a model. The final commonly used approach is to construct a generative model of each series such as an autoregressive moving average (ARMA) model or hidden Markov model (HMM) and then to use the model parameters as features for classification [1,9]. The generative model based schemes are best suited for detecting similarity in change and are hence the least used approach, since most problems used in research are more suited to similarity in shape.

Clearly the approaches can be mixed. The main distinction is whether to preprocess the data to capture the different types of similarity or to embed the method within the classification algorithm. This is analogous to the difference between a filter and a wrapper approach to feature selection/creation. In this paper we concentrate on bespoke feature extraction.

3 The Data

The trial involved measuring the power consumption of 187 households for a variety of devices as identified by the participants. We extracted data on the ten most commonly identified devices: immersion heater; washing machine; fridge; freezer; fridge/freezer; oven/cooker; computer; television; and dishwasher. We created two classification problems: For the first set a case consisted of the daily measurements of the specified device (96 attributes), for the second set we used a week of readings (672 attributes). After data cleansing and validation, the daily set has 78,869 cases, the weekly set 9,215 cases. Figure 1 gives some examples of the resulting demand profiles.

This problem has several confounding factors that will make classification difficult. Firstly the fact that measurements are summed over 15 minutes makes it harder to detect devices that peak over a short period. For example, a kettle will consume a large amount of power whilst on, but will only be on for a two

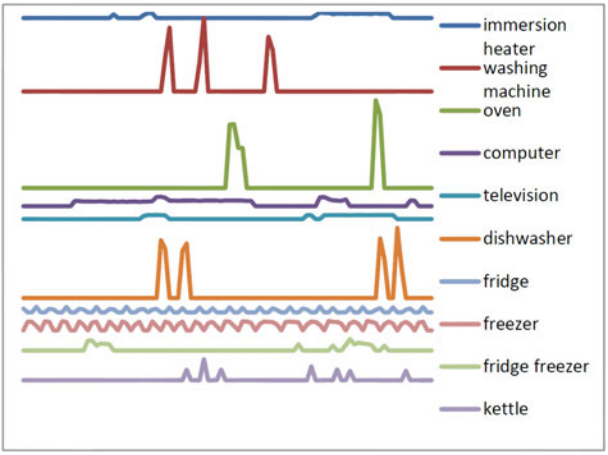


Fig. 1. Examples of daily profiles for the ten devices considered

or three minutes; when summed over 15 minutes it will be harder to distinguish from a device such as a dish washer or washing machine which consume lower power but will be on for the whole period. Secondly, there will be a seasonal variation in the use of devices such as immersion heaters. Thirdly, we would also expect it to be hard to distinguish between similar devices such as a fridge and a fridge/freezer and finally, we would expect considerable variation between different devices of the same class.

Since our objective is to be able to identify a device for a new user with no labelled usage history we need to design our experiment to avoid a potential bias into our experiments. It will surely be easier to identify a device for a single household than across all households, thus if measurements from a single household appear in both the training and testing sets, our accuracy estimates are liable to be over optimistic for unseen households. Hence we design all cross validation experiments so that the test and train splits are always composed of different households.

An examination of Figure 1 highlights the nature of the similarity measures this problem will require and hence the transformations we consider. Firstly, when a device is on and the level of power used are clearly important. Hence our first approach is to simply use the raw data. However, it seems unlikely that a correlation based approach will be sufficient, given the variability of usage pattern within each class. Our second approach is to derive a set of features describing the distribution of power used when a device is on and the distribution of length of time on. Table 1 lists these features and presents the summary statistics averaged over all devices of each class of device. The mean values in Table 1 shows that there are clusters of power consumption over a 15 minute interval, which can be characterised as low (computer, freezer, fridge, fridge, TV and kettle) and high (dishwasher, immersion heater, oven and washing machine). There is also a wide variation between classes in the duration of usage; the average time on for computers is approximately 7 hours, for cookers 42 minutes and for kettles 16 minutes (skewed because the minimum on-time is 15 minutes). This indicates that these statistics may be useful in constructing classifiers.

4 Results

Table 2 gives the accuracy results for a ten fold cross validation (where no single household appears in both the training and testing fold) using five different classifiers on the daily and weekly data sets. For the daily data, we observe that using the derived features improves the performance of all the classifiers except random forest, but that this improvement is small and the best overall performance is with random forest on the raw data. This suggests that the inbuilt ensemble mechanism of the random forest classifier is at least as good at capturing the inherent similarity as our derived features. Further experiments (not reported here) showed that there was also no improvement with dynamic time warping and FFT derived feature sets. However, for multi-class problems such as this accuracy tends not to tell the whole story. Tables 3 and 4 show the

Table 1. Summary statistics for the daily data set. Each data is averaged over all cases of the given class. So, for example, the minimum power usage in any one 15 minute period for a computer is 26.35 Wh when we average across all computers, assuming any power is being used at all.

	Computer	Dish Washer	Freezer	Fridge	Fridge/ Freezer	Immersion Heater	Kettle	Oven/ Cooker	Television	Washing Machine
Summary statistics for power usage when a device is in use (Wh)										
min	26.35	276.01	17.61	12.34	19.07	151.07	61.44	252.05	29.65	274.96
max	39.79	457.92	37.34	27.07	45.42	245.49	113.98	423.94	45.24	375.88
mean	33.13	365.13	27.14	20.10	28.91	201.80	84.94	328.79	39.36	324.14
std dev	6.42	102.69	8.11	4.55	7.28	75.65	27.10	114.63	11.67	81.13
skewness	0.07	0.03	-0.22	-0.55	0.45	0.11	0.17	0.26	-1.08	0.03
kurtosis	2.62	-1.44	1.35	0.64	5.05	0.53	-0.94	-0.73	3.42	-1.04
Summary stats of device usage tendencies										
% on	0.40	0.04	0.47	0.39	0.45	0.20	0.05	0.06	0.25	0.03
first usage	33.73	51.57	1.69	1.83	5.87	28.88	32.34	61.15	45.09	45.78
Summary stats of the number of time steps a device is on for										
num runs	3.03	2.13	23.18	17.91	14.27	6.55	4.55	1.94	2.71	1.70
run min	28.00	1.61	5.99	2.30	5.86	3.58	1.02	2.45	8.04	1.32
run max	33.68	2.17	9.60	5.90	13.17	7.99	1.27	3.36	16.80	1.59
run mean	30.54	1.88	6.97	3.55	8.73	5.43	1.07	2.84	11.82	1.44

Table 2. Classification accuracy (and standard deviation) for a ten fold cross validation on the daily and weekly data sets

Daily data set			
	Naive Bayes	**C4.5**	**Random Forest**
Raw Data	38.40% (4.69)	56.60% (4.17)	61.34% (4.67)
Derived Features	44.01% (4.74)	58.89% (5.01)	59.04% (4.11)
	SMO (SVM)	**NN ($k = 21$)**	**NN ($k = 51$)**
Raw Data	43.52% (5.38)	56.72% (4.85)	53.82% (5.36)
Derived Features	59.46% (4.24)	60.86% (5.82)	60.95% (6.3)
Weekly data set			
	Naive Bayes	**C4.5**	**Random Forest**
Raw Data	41.43% (3.16)	46.42% (4.46)	55.81% (6.27)
Derived Features	44.80% (6.44)	62.90% (4.62)	64.81% (5.5)
	SMO (SVM)	**NN ($k = 21$)**	**NN ($k = 51$)**
Raw Data	48.79% (6.81)	31.50% (6.89)	26.48% (6.51)
Derived Features	54.40% (5.73)	63.25% (6.44)	63.17% (6.28)

confusion matrices for the random forest classifier on the daily raw and derived data sets. These tables demonstrate that the confusion for the raw data seems to be more widely distributed between all the classes, whereas on the derived feature set random forest is more systematic in it's mistakes.

The results for the weekly data set are more clear cut, in that the classifiers trained on the derived features clearly outperform those trained on the raw data. The random forest confusion matrices given in Table 5 and 6 further demonstrate the improved performance.

Table 3. Confusion matrix for Random Forest on the daily raw data

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**
a = computer	4471	36	520	126	39	280	69	273	3649	156
b = dishwasher	174	5021	5	5	2	121	16	928	28	320
c = freezer	596	3	4060	2016	318	12	56	33	173	12
d = fridge	108	9	2403	3801	385	4	3	9	35	7
e = fridgeFreezer	93	2	1177	755	49	2	0	13	122	0
f = immersionHeater	631	263	188	98	18	520	583	409	543	146
g = kettle	28	26	36	7	0	72	8000	668	40	145
h = ovenCooker	564	600	268	80	29	139	1066	8166	632	305
i = television	4547	81	206	167	167	310	82	504	9392	284
j = washingMachine	66	358	11	13	0	30	323	547	118	4897

Table 4. Confusion matrix for Random Forest on the daily derived features

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**
a = computer	2912	0	388	20	76	332	173	90	2825	2
b = dishwasher	0	3785	1	0	0	52	0	1414	6	932
c = freezer	306	8	3704	2106	369	12	154	6	65	1
d = fridge	66	0	1952	3717	486	1	2	0	222	0
e = fridgeFreezer	272	0	447	907	166	5	126	0	96	0
f = immersionHeater	734	169	10	3	16	814	379	481	211	103
g = kettle	78	0	205	5	13	145	7954	72	97	32
h = ovenCooker	52	1100	5	0	0	486	71	6158	37	1802
i = television	2950	12	41	125	20	168	159	103	7789	36
j = washingMachine	0	1217	0	0	0	30	103	1990	40	2230

Table 5. Confusion matrix for Random Forest on the weekly raw data

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**
a = computer	360	13	85	22	2	31	5	24	437	26
b = dishwasher	14	389	0	6	0	25	25	243	27	160
c = freezer	49	1	409	177	34	1	10	4	32	2
d = fridge	39	4	245	352	17	1	3	3	11	0
e = fridgeFreezer	9	1	98	78	6	0	0	1	16	0
f = immersionHeater	48	61	21	21	2	35	76	39	63	32
g = kettle	4	27	1	0	0	14	733	69	20	113
h = ovenCooker	48	94	40	3	2	8	56	1169	87	79
i = television	365	56	41	35	15	23	19	58	985	42
j = washingMachine	7	124	0	0	0	7	99	146	26	705

The largest source of confusion is the expected problem of distinguishing between fridge, freezer and fridge freezer. Computer and television are also often confused. Table 7 shows the accuracy on the daily data set when we merge the classes fridge, freezer and fridge freezer (cold group) and computer and television into screen group. Unsurprisingly, the accuracy is much higher. For the daily data, we again observe that the derived features improve accuracy across all reported classifiers except random forest, which again recorded the best accuracy using the raw data. Tables 8 and 9 show the confusion matrices for the random forest classifier on the daily raw and derived data sets. These tables demonstrate that confusion has been significantly reduced when compared to using the full set of classes, and the difference between the confusion of raw and derived features has also been reduced.

Table 6. Confusion matrix for Random Forest on the weekly derived features

	a	b	c	d	e	f	g	h	i	j
a = computer	382	0	45	0	3	32	18	1	358	1
b = dishwasher	0	520	2	0	0	2	0	125	3	220
c = freezer	23	1	375	184	62	2	24	1	3	0
d = fridge	2	0	201	358	69	1	0	0	14	0
e = fridgeFreezer	24	0	72	70	22	7	0	0	1	0
f = immersionHeater	86	18	1	1	5	71	53	48	34	5
g = kettle	8	0	24	1	2	17	886	3	4	5
h = ovenCooker	9	61	1	0	4	69	1	1233	14	150
i = television	271	25	4	14	1	32	2	17	952	4
j = washingMachine	2	207	0	0	0	0	12	178	4	668

Table 7. Classification accuracy using cold and screen groups

	Weekly data set			
	C4.5	**Random Forest**	**SMO (SVM)**	**NN ($k = 21$)**
Raw Data	77.09% (3.92)	81.27% (4.62)	56.47% (7.18)	74.08% (5.20)
Derived Features	78.25% (4.18)	77.38% (3.69)	74.90% (4.19)	78.56% (4.20)
	Weekly data set			
	C4.5	**Random Forest**	**SMO (SVM)**	**NN ($k = 21$)**
Raw Data	65.36% (4.52)	72.96% (5.56)	60.53% (9.10)	39.89% (6.40)
Derived Features	78.19% (4.11)	80.32% (4.69)	69.02% (3.57)	78.03% (4.95)

Table 8. Confusion matrix for Random Forest on the daily raw data with cold and screen groups

	a	b	c	d	e	f	g
a = screenGroup	22614	88	1174	443	127	554	359
b = dishwasher	244	5029	19	100	14	888	326
c = coldGroup	1033	8	15096	8	59	36	16
d = immersionHeater	1347	215	309	425	594	355	154
e = kettle	105	23	44	73	7980	661	136
f = ovenCooker	1443	605	262	116	988	8117	318
g = washingMachine	257	339	34	42	323	535	4833

Table 9. Confusion matrix for Random Forest on the daily features with cold and screen groups

	a	b	c	d	e	f	g
a = screenGroup	16746	10	625	370	237	192	41
b = dishwasher	6	3781	1	48	0	1442	912
c = coldGroup	1125	8	13900	13	145	4	1
d = immersionHeater	961	170	50	705	365	560	109
e = kettle	232	0	215	136	7913	73	32
f = ovenCooker	70	1137	2	509	69	6148	1776
g = washingMachine	50	1195	1	28	93	2018	2225

The results of the weekly data are again much more clear cut, with derived features clearly outperforming raw data. Tables 10 and 11 show the confusion matrices for the random forest classifier on the raw and derived feature data. They demonstrate a pattern similar to the first round of experiments, where the confusion for the raw data appears to be more widely distributed than the derived features.

Table 10. Confusion matrix for Random Forest on the weekly raw data with screen and cold groups

	a	**b**	**c**	**d**	**e**	**f**	**g**
a = screenGroup	22614	88	1174	443	127	554	359
b = dishwasher	244	5029	19	100	14	888	326
c = coldGroup	1033	8	15096	8	59	36	16
d = immersionHeater	1347	215	309	425	594	355	154
e = kettle	105	23	44	73	7980	661	136
f = ovenCooker	1443	605	262	116	988	8117	318
g = washingMachine	257	339	34	42	323	535	4833

Table 11. Confusion matrix for Random Forest on the weekly features with screen and cold groups

	a	**b**	**c**	**d**	**e**	**f**	**g**
a = screenGroup	16746	10	625	370	237	192	41
b = dishwasher	6	3781	1	48	0	1442	912
c = coldGroup	1125	8	13900	13	145	4	1
d = immersionHeater	961	170	50	705	365	560	109
e = kettle	232	0	215	136	7913	73	32
f = ovenCooker	70	1137	2	509	69	6148	1776
g = washingMachine	50	1195	1	28	93	2018	2225

5 Conclusions and Future Work

In this paper we have proposed the time series classification problem of classifying household goods based solely on the electricity usage of the device as measured by a GEO smart meter. The ability to automatically detect the type of a device gives insights into the breakdown of the household usage pattern and offers the potential for providing useful feedback to the consumer, both in terms of minimizing their usage and in fault detection. We have assessed alternative classifiers and transformation for this problem and conclude that with a weekly profile we can accurately discriminate between classes of device by deriving a set of descriptive features and using a random forest or nearest neighbour classifier.

Data mining of smart meter data is going to be crucial in order to get the best value out of the massive investment required for the national rollout program. This problem represents just one potential secondary use of the data. We may be able to achieve improved classification performance through consideration of more complex transformations and ensemble classifiers.

References

1. Bagnall, A.J., Janacek, G.J.: Clustering time series from arma models with clipped data. In: 10th International Conference on Knowledge Discovery in Data and Data Mining (ACM SIGKDD 2004), pp. 49–58 (2004)
2. Darby, S.: Making it obvious: designing feedback into energy consumption, pp. 685–696 (2001)

3. Janacek, G.J., Bagnall, A.J., Powell, M.: A likelihood ratio distance measure for the similarity between the fourier transform of time series. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 737–743. Springer, Heidelberg (2005)
4. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining and Knowledge Discovery 7(4), 349–371 (2003)
5. Keogh, E., Pazzani, M.: Scaling up dynamic time warping for datamining applications. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2000, pp. 285–289 (2000)
6. Mörchen, F.: Time series feature extraction for data mining using DWT and DFT, Tech. Report 3. Department of Mathematics and Computer Science Philipps-University Marburg (2003)
7. Mörchen, F., Mierswa, I., Ultsch, A.: Understandable models of music collections based on exhaustive feature generation with temporal statistics. In: 12th International Conference on Knowledge Discovery in Data and Data Mining (ACM SIGKDD 2006), pp. 882–891 (2006)
8. Department of Energy and Climate Change, Smart metering implementation programme, White paper, Department of Energy and Climate Change (July 2010)
9. Smyth, P.: Clustering sequences with hidden markov models. In: Advances in Neural Information Processing Systems, vol. 9 (1997)
10. Toshniwal, D.: Feature extraction from time series data. International Journal of Computational Methods in Sciences and Engineering 8(1), 99–110 (2009)

From Fuzzy Association Rule Mining to Effective Classification Framework

Osama Alhawsawi[1], Mayad AL-Saidi[1], Michael Phi[1],
Tamer N. Jarada[1], Mohammad Khabbaz[2], Negar Koockakzadeh[1],
Keivan Kianmehr[3], Reda Alhajj[1,4], and Jon Rokne[1]

[1] Computer Science Dept, University of Calgary, Calgary, Alberta, Canada
[2] Computer Science Dept, University of British Columbia, Vancouver, BC, Canada
[3] Dept of Elect. & Comp. Eng., University of Western Ontario, London, ON, Canada
[4] Department of Computer Science, Global University, Beirut, Lebanon

Abstract. Given a set of known classes, classification is a two steps process which uses part of the data to build a model capable of determining the class of new objects not used in the training phase. The accuracy of the classifier is one of the main criteria to judge its usefulness. However, most of the existing classification approaches decide on a single class for a given object. We argue that fuzzy classification is more attractive because it is closer to the real case where it is hard to identify a unique one class per object. To tackle this problem, we developed a framework which produces fuzzy association rules and uses them to build the classifier model. There are two important factors to consider: the method to create fuzzy association rules must be accurate, and the method to build a classifier must be accurate as well. In this paper, we will describe a method to perform fuzzy association rule mining and classification and we will test our results based on numerous factors including accuracy, varying levels of support and confidence.

Keywords: Fuzzy Sets, Fuzzy Association Rules, Data Mining, Classification, APRIORI, Fuzzy Partitioning, Sharp Boundary.

1 Introduction

The last few decades witnessed tremendous increase in the amount of data collected and kept for further analysis. Unfortunately, traditional data processing techniques are query driven and are only capable of retrieving data that explicitly exists in the repository. On the other hand, for long time statisticians have well established techniques for data analysis. They are powerful enough to fit the data to a certain model that could be used later on for inference. However, statistical techniques are mostly mathematical models and hence their capabilities are limited. Fortunately, the combined efforts of the database and artificial intelligence communities reported a major accomplishment by developing the applied methods for data analysis, including the various data mining techniques like classification, clustering and association rules mining. The latter techniques

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 413–420, 2011.

proved to be more comprehensive and powerful that the statistical techniques though the former inspired a lot from the latter. In other words, the most powerful data mining techniques are the ones that have some mathematical and statistical basis.

The work described in this paper benefits from association rules mining to develop an effective classifier. More than that, we utilize fuzzy association rules to build fuzzy classification framework. Classification is a supervised learning technique that requires two main inputs, a set of predefined classes and a set of objects where each object has certain characteristics and is expected to belong to one of the given classes. The process of building the classifier proceeds as follows. First, the set of features is reduced to concentrate only on features that well characterize the given classes. Second, a classifier model is constructed by considering the reduced set of features to decide on the class of each object based on its features.

A fuzzy rule is expressed as follows: X is $A \longrightarrow Y$ is B, where X and Y are disjoint sets of features, and A and B are corresponding sets of fuzzy sets, respectively. In other words, if X has n features then A will contain n fuzzy sets, one fuzzy set per feature. Each feature value may belong to its corresponding fuzzy sets with a degree of membership per fuzzy set as determined by the corresponding membership function. In this study we use the triangular membership function though other membership functions like trapezoidal may be equally used.

For the study described in this paper, we concentrate on fuzzy association rules where the consequent includes only the class label. This facilitates constructing the classification framework described in this paper. We have tested the proposed framework using five datasets. The reported results are encouraging; they demonstrate the applicability and effectiveness of the proposed approach.

The rest of this paper is organized as follows. Related work is covered in Section 2. The proposed framework is described in Section 3. The experimental results are reported in Section 4. Section 5 is conclusions.

2 Related Work

To solve the qualitative knowledge discovery problem, Au and Chan proposed the F-APACS method [1] for discovering fuzzy association rules. Hong *et al.* [4] proposed an algorithm that integrates fuzzy set concepts and Apriori mining algorithm to find interesting fuzzy association rules from given transactional data. Hong et al. [5] also proposed definitions for the support and confidence of fuzzy membership grades and designed a data mining approach based on fuzzy sets to find association rules with linguistic terms of human knowledge. Ishibuchi et al. [6] illustrated fuzzy versions of confidence and support that can be used to evaluate each association rule. The approach developed by Zhang [11] extends the equi-depth partitioning with fuzzy terms.

As described in the literature, associative classification includes two major techniques: classification based on single class association rule and classification based on multiple class association rules. Alternative approaches have been also proposed to combine different techniques to improve the classification efficiency. For classification based on single class association rules, e.g., CBA [8] and C4.5 [9], a subset of strong rules, called candidate set, is sorted in descending order of rules accuracy (a rule is accurate if both its confidence and support are above the pre-determined thresholds). For classification based on multiple class association rules, the candidate set of strong rules is not sorted and most matching rules may participate in the classification process. Simply, if the new data object is covered by all candidate rules, it will be assigned to the class label of candidate rules; otherwise the majority vote of candidate rules will specify the class label of the new object. CPAR [10] employs this technique to classify new data objects. Another method is to divide candidate rules into groups according to class labels, and then to compute the actual effect obtained from the multiple rules for all groups. The group with the highest efficiency will identify the class of the new object; CMAR [7] employs this method. Fuzzy associations rules have been used for associative classification in a recent study by Chen *et. al.* in [2].

3 The Proposed Classification Framework

In this paper, a solution will be introduced where fuzzy association rule mining issued to classify quantitative data. The proposed solution has been implemented in three main phases as described in this section.

Representing Data as Fuzzy Sets: The first phase is reading all data elements and representing each attribute in terms of membership in a number of fuzzy sets that cover the attributes domain. This is implemented as follows:

```
1.  Read data from training set
2.      For each record:
3.          For each attribute in the record
4. GetDomain(attribute)
5. CreateFuzzySets(attribute)
6.  For each record:
7.      For each attribute in the record
8. getFuzzyMembershipValues(attribute)
```

We start by reading in all the training data and finding the domain of each attribute (lines 1-4). The domain of an attribute is the area between its minimum and maximum value in the training dataset. The domain is then partitioned into n fuzzy sets (n is the user-specified number of fuzzy sets used to partition the domain of each attribute) (line 5). Now that the fuzzy partitions for each attribute are known, we scan the data again to find the membership value of each attribute in each fuzzy set using the function shown in Figure 1(for $n = 5$).

Classification Rule Mining: In this phase, we mine the data looking for rules associating different attributes to class labels. We use the membership values generated in the previous phase in searching for such rules as follows:

```
fuzzyVals = getFuzzyVals(Data)
 classSets = split(fuzzyVals)
 for each classSet
     oneItemRules = getOneItemRule(fuzzySet)
```

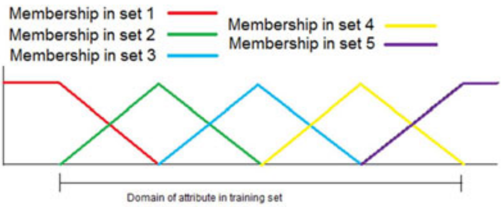


Fig. 1. Membership values for each fuzzy set

```
5.      rules.append(oneItemRules)
6.      nextItemRules = getNextItemRule(fuzzySet, oneItemRules)
7.      rules.append(nextItemRule)
8.      while nextItemRules NOT empty
9.  nextItemRules = getNextItemRule(fuzzySet,nextItemRules)
10. rules.append(nextItemRule)
11. for each rule
12.     confidence = getConfidence(Rule)
13.     if confidence < minConfidence then drop the rule from rules
14. rules = pruneRules(rules)
15. return rules
```

In line (1), the fuzzy representation of the training dataset is found as described previously. In line (2), the algorithm splits the dataset based on the class label. So classSets is going to be a set of sets where the first set contains all the records that have class label 1, the second set contains all the records that have class label 2, and so on. Then, for each of these classSets, find one item rules (line 4) and append them to the set of rules (line 5). To find one item rules, the algorithm goes through the whole classSet and finds candidate one item rules then calculates their support. Rules with support less than the minimum support are not added to the result. After the set of oneItemRules is found for each classSet, the set of nextItemRules are found (line 6). nextItemRules are found in the same way that oneItemRules were found, then they are appended to the set of rules (line 7). The while-loop (line 8) is needed to determine when the algorithm should stop trying to find nextItemRules. For example, if we couldn't find frequent 4-item rules then it is impossible to find frequent 5-item rules since the Apriori works under the assumption that every subset of a frequent item set must also be frequent. When the algorithm gets to line (11), "rules" will contain all rules that satisfy the minimum support set by the user. The confidence of the rules is not yet considered, and that is where the for-loop on line (11) comes into play. For each rule, calculate the confidence (line 12). If the confidence of the rule is less than the minimum confidence set by the user (line 13), drop the rule from the set of rules. Then, remove the rules that are not "interesting" (line 14). Finally, "rules" will have all the rules that are going to be used to classify the records in the test dataset.

Applying Classification Rules on Test Data: A classifier must be built to classify the data elements. This classifier must be appropriate for the given data element. The strongest class label that classifies a given data element is the classifier for the data element. To determine the strongest classifier, three different methods are presented. . Each method chooses a different criterion to determine which class most strongly classifies a data element. In the first method, the classification strengths of each rule for a given data element is calculated and the class belonging to the rule of the highest classification strength is the class that classifies the data element. In the second method, the classification strengths for each class are summed up and the class with the highest sum total is the class that classifies the data element. In the last method, the class with the most number of rules belonging to it is the class that classifies the data element.

Method 1: There are two steps in building an appropriate classifier for this method. The first step is to calculate the classification strengths of each rule for a specific data element. In the second step, the class that belongs to the rule of the highest classification is chosen to be the classifier. The pseudo code for step one is explained below:

```
1. For each data element in the testing set
2.  For each rule
3.      minDegree = $\infty$
4. For each attribute in the rule
5.          degreeMembership = calcDegreeMembership(data element, fuzzy set)
6. If degreeMembership < minDegree
7.              minDegree = degreeMembership
8. classificationStrength=calcClassificationStrength(mindegree, support, confidence)
9.      Add the classificationStrength to the current rule of the current data element
10. Choose the class belonging to the rule of the highest classification strength
11. This class is the class that classifies the current data element
```

For each data element, we will traverse through all the rules to find the strongest classification strength. We set minDegree to a very high number to initialize the process(line 3). For each attribute of the given rule, we calculate the degree membership of it by using the partitioning method as described earlier in the paper (line 4-5) If the degree membership is less than the minDegree, the degree membership is the new minDegree. Next, we find the classification strength by perform the function calcClassificationStrength by multiplying the minDegree by the support and confidence (line 8). Next, we add the classification strength for each rule to a list(line 9). The index of the list represents the rule the order of rules. Thus, the first element in the list is the classification strength corresponding to the first rule and the second element is the classification strength belonging to the second rule and so on. Finally, we traverse through the list to find the maximum classification strength (line 10). The class that corresponds to the rule is the class that classifies the data element (line 11).

Method 2: The second method is very similar to method 1. The only difference is in the determination of a proper classifier. Where as method 1 chose the class belonging to the rule of the highest classification strength, method 2 propose that if the sum of the classification strengths of each class is added up, the class with the highest sum is the class that classifies the data element.

```
1. For each data element in the testing set
2.  For each rule
3.      minDegree = infinite
4.      For each attribute in the rule
```

```
5.          degreeMembership = calcDegreeMembership(data element, fuzzy set)
6. If degreeMembership < minDegree
7.              minDegree = degreeMembership
8. classificationStrength=calcClassificationStrength
9.      Add the classificationStrength to the current rule of the current data element
10. For each distinct class presented in the rules list
11.     If the class of the rule matches the current class
12.         Add up the classification strength for the current class
13. This class with the highest summed classification strength is the class that classifies
    the current data element
```

Lines 1-8 is the same as in the previous method. The method begins to differ from method 1 starting at line 9. In line 9, we find all the distinct classes that were set in the rules. The rules are traversed and if the rule corresponds with the current class, the classification strength of the rule is added up to find the sum of all the classification strengths of all the rules that belong to a class(line 10-11). The class with the highest sum is the class that classifies the current data element (line 12).

Method 3: The last method presents another way to determine which class classifies the current data element. This method does not use classification strength, but counts up the number of rules that belong to a class. The class with the highest number of rules is the class that classifies the data element.

```
1. For each data element in the testing set
2.  For each distinct class in the rules list
3.    Count=0
4. For each rule
5.      If the rule belongs to the current class
6.              Count++
7. The class with the highest count is the class that classifies the data element.
```

Count is initialized to 0 to count the number of occurrences of a rule in a unique class (line 3). For each rule, the rule is checked to see if it belongs to the class. If it belongs to the class, count is incremented (line4-6). At the end of traversing through all the unique classes, the class with the highest number is the class that classifies the data element (line 7).

Table 1. UCI ML datasets used in the experiments

Dataset	# Records	# Attributes	# Classes
Wine	178	13	3
bupa	345	6	2
Ionosphere	351	35	2
PageBlocks	5,473	11	5
Waveform	5,000	22	3

4 Experiments

We tested the proposed framework on five datasets from UCI ML repository [3]. Table 1 describes these test datasets along with some related statistical information. In the first set of experiments, we have compared the performance of different methods described in Section 3 using the first two datasets in Table 1. We used the 10-fold validation procedure to test the three different classification methods. The classifiers were tested using different fuzzy set representations and

Table 2. The effects of the number of fuzzy sets, support, and confidence on the success rate of the three classification methods on the wine and bupa datasets

Dataset	No. of Fuzzy Sets	Minimum Support	Minimum Confidence	Classification Accuracy (method1)	Classification Accuracy (method2)	Classification Accuracy (method3)
wine	3	20%	30%	75.62%	50.21%	40.61%
wine	3	20%	40%	76.90%	50.21%	40.61%
wine	3	20%	50%	93.44%	51.95%	40.61%
wine	5	30%	40%	0.00%	0.00%	0.00%
wine	5	30%	50%	0.00%	0.00%	0.00%
bupa	3	20%	30%	56.41%	56.34%	57.26%
bupa	3	25%	50%	56.41%	56.34%	57.26%
bupa	5	30%	30%	56.41%	54.69%	57.26%
bupa	5	30%	40%	56.41%	54.69%	57.26%
bupa	5	30%	50%	56.41%	54.69%	57.26%

with several minimum support and confidence value. The results are reported in Table 2.

With the wine dataset there were no classification rules generated for as the support went over 25%, this resulted in the poor classification results that can be seen in Table 2. The best classification results were obtained with the wine dataset using 3 fuzzy sets to represent attributes, 20% minimum support and 50% minimum Confidence as can be seen in the fourth row of the table. Overall with classification method 1 gave the best results with the wine dataset. For the bupa dataset, the classification results were quite similar as we experimented with different minimum support and confidence levels. The results were just above 50% and changes in the number of fuzzy sets used did not have a big impact on them.

Table 3. Accuracy results with the best support and confidence

Dataset	CBA			CMAR			FCAR		
	Acc.	S	C	Acc.	S	C	Acc.	S	C
Ionosphere	89.5	10.0	49.2	91.5	2.6	50.0	89.2	26.4	80.0
PageBlocks	91.0	1.6	50.0	90.3	0.2	50.0	95.8	26.4	88.0
Waveform	78.2	2.6	50.0	77.2	0.6	50.0	99.9	17.4	50.5

In the second set of experiments, we conducted tests using our method extended by adapting a greedy search algorithm for finding the best support and confidence values for each dataset. In terms of classification accuracy,we compared the performance of our model with CBA and CMAR from the family of associative classifiers. For the experiments conducted to evaluate the performance of the proposed method with the best support and confidence values, we used last three datasets shown in Table 1.

Table 3 shows the accuracy results associated with the best support and confidence thresholds found by the greedy search algorithm. As can be seen from Table 3, in all cases except the ionosphere dataset, our method outperforms the other two methods, all combined with the greedy search algorithm for finding the best support and confidence values. For the ionosphere dataset, our result is very close and still comparable. A large support value usually works better in our model when the input dataset is a uniformly distributed binary-class. This will have the effect of producing less rules in the system, which in turn will provide computation efficiency benefits. When the binary-class dataset is not monotonously distributed (like ionosphere), medium support values perform better. For multi-class datasets with small number of classes (waveform has 3 uniform classes, so a high value of support like 17.4% performs well). As the confidence threshold is concerned, we noticed that by increasing its value, the performance of our model does not improve. Furthermore, increasing the confidence above a certain value leads to a decrease in the model accuracy.

5 Conclusions

In this paper, we proposed a classification system that employs fuzzy association rule mining to generate a set of classification rules. These rules are used to classify data in three different methods in order to get a better idea of how accurate these rules are. We performed some experiments and presented observations that were seen in the experiments. Future work could include implementing different association rule mining techniques such as ECLAT or FP-Growth to enhance and optimize the performance of the system. Using more datasets to test the performance of the system under different minimum support and confidence requirements will help provide a better idea on what support and confidence values produce the best results.

References

1. Au, W.-H., Chan, K.C.C.: An effective algorithm for discovering fuzzy rules in relational databases. In: Proc. of IEEE International Conference on Fuzzy Systems, pp. 1314–1319 (May 1998)
2. Chen, Z., Chen, G.: Building an associative classifier based on fuzzy association rules. Journal of Computational Intelligence Systems 1(3), 262–273 (2008)
3. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
4. Hong, T.-P., Kuo, C.-S., Chi, S.-C.: A fuzzy data mining algorithm for quantitative values. In: Proc. of KES, pp. 480–483 (1999)
5. Hong, T.-P., Kuo, C.-S., Chi, S.-C.: Mining association rules from quantitative data. Intell. Data Anal. 3(5), 363–376 (1999)
6. Ishibuchi, H., Nakashima, T., Yamamoto, T.: Fuzzy association rules for handling continuous attributes. In: Proc. of IEEE ISIE, pp. 118–121 (2001)
7. Li, W., Han, J., Pei, J.: Cmar: Accurate and efficient classification based on multiple class-association rules. In: Proc. of IEEE ICDM, pp. 369–376 (2001)
8. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proc. of ACM-KDD, pp. 80–86 (1998)
9. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
10. Yin, X., Han, J.: Cpar: Classification based on predictive association rules. In: Proc. of SDM (2003)
11. Zhang, W.: Mining fuzzy quantitative association rules. In: Proc. of IEEE ICTAI, p. 99 (1999)

From Alternative Clustering to Robust Clustering and Its Application to Gene Expression Data*

Peter Peng[1], Mohamad Nagi[2], Omer Şair[3], Iyad Suleiman[2], Ala Qabaja[1], Abdallah M. ElSheikh[1], Shang Gao[1], Tansel Özyer[3], Keivan Kianmehr[4], Ghada Naji[5], Mick Ridley[2], Jon Rokne[1], and Reda Alhajj[1,6]

[1] Computer Science Dept, University of Calgary, Calgary, Alberta, Canada
[2] School of Computing, University of Bradford, Bradford, UK
[3] Dept of Computer Engineering, TOBB University, Ankara, Turkey
[4] Dept of Elect. & Comp. Eng., University of Western Ontario, London, ON, Canada
[5] Department of Biology, Lebanese University, Tripoli, Lebanon
[6] Department of Computer Science, Global University, Beirut, Lebanon

Abstract. The major contribution of the work described in this paper could be articulated as a parameter free clustering approach that leads to appropriate distribution of the given data instances into the most convenient clusters. This goal is realized in several steps. First, we apply multi-objective genetic algorithm to determine some alternative clustering solutions that constitute the pareto-front. The result is a pool of the clusters reported by all the solutions. Then, we determine the homogeneity of each cluster in the pool to keep the most homogeneous clusters which may not be select from one solution because a solution which is favored the most by considering the multiple objectives might have some clusters which are less homogeneous compared to best clusters in other solutions. Finally, as a given data instance may belong to more than one cluster in the solution set we reduce this membership to the cluster in which the instance is closest to the centroid. Many applications like gene expression data analysis are in need for such parameter free approach because the correctness of the post processing is directly affected by the outcome form the clustering process. We demonstrate the applicability and effectiveness of the proposed clustering approach by conducting experiments using two benchmark data sets.

Keywords: multi-objective genetic algorithm, clustering, knowledge discovery, gene expression data.

1 Introduction

Clustering is one of the oldest mining techniques that has influenced research in various domains for several decades. It is the process of distributing objects

* This study was supported by Scientific and Technical Research Council of Turkey (Grant number TÜBITAK EEEAG 109E241).

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 421–428, 2011.

into classes such that the similarity between objects in the same class and the dissimilarity between the classes are both high. Achieving these two targets simultaneously is a challenge that has received considerable attention in the research community. In other words, clustering continues to be an active research area because of its wide range of existing and emerging applications.

Our approach presented in this paper has been designed to handle the clustering of a given set of instances without requiring any parameter be specified in advance. Approaches like k-means require the number of clusters explicitly specified and approaches like DBSCAN are based on some parameters that implicitly simulate the number of clusters. A parameter free clustering approach is critical for many domains where finding the most appropriate clustering is directly reflected into the analysis of the results. One such domain is gene expression data analysis, which is one of the domains considered in the experiments conducted in this study. Our approach starts by applying a multi-objective k-means genetic algorithm (MOKGA) in order to determine several alternative clustering solutions without taking the weight values into account [9]. We run cluster validity analysis indexes, namely Dunn [2], Davies-Bouldin [1], Silhouette [8], C [6], SD [3] and S_Dbw [4] on the alternative solutions to determine the number of compact clusters to have in the final solution. Then, we collapse all the alternative solutions obtained from MOKGA to form a common pool of clusters, where clusters coming from the same solution are disjoint and clusters from different solutions mostly overlap. Analyzing all the clusters in the pool at once gives equal opportunity to every cluster to show up in the final solution which should include the most compact clusters which may come from various solutions. This is more natural process than individually analyzing the alternative solutions. In other words, we zoom into the details of each solution because some solutions may include more compact individual clusters than a single favored solution. At the end of this process, we will have a collection of compact clusters that mostly overlap. The overlap is eliminated by keeping each data instance only in the cluster where the data instance is closest to the centroid of the cluster. In case of objects that do not end up in any of the identified clusters, we first measure the distance between these objects and the centroids of the clusters in the final solution set. Then we have two choices, either to add an object to a cluster based on shortest distance and provided that it does not destroy the compactness of the cluster or to consider the object as outlier otherwise.

As gene expression data analysis is concerned, benefiting from the advantage of the proposed clustering approach, we use the gene closest to the centroid as reduced feature to represent the cluster. Thus, after the clustering is over and the most appropriate clustering is identified, the genes closest to centroids (one gene per cluster) represent the whole data. The latter genes form valuable source of information for further analysis of the gene expression data to discover the biomarkers [10]. The compact solution produced by the clustering approach described in this paper provides the opportunity to consider more appropriate biomarker genes. Finally, the applicability and effectiveness of the proposed approach has been tested using two benchmark data sets; the results are promising.

Here it is worth mentioning that the proposed approach is capable of locating outliers, but this property is still to be validated by considering some other synthetic or real data sets with outliers. We have left this out as future study because none of the data sets used in the testing contains outliers.

The rest of the paper is organized as follows. Section 2 describes the proposed approach. Section 3 discusses the experiments. Section 4 is summary and conclusions.

2 From Alternative Solutions to Robust Solution

In this section, we describe the clustering approach that starts by applying the Multi-Objective Genetic K-means algorithm (MOKGA) to produce alternative solutions which are collapsed into one pool of clusters to be further analyzed. It is a general purpose approach for clustering datasets from various domains. The only tuning required is modifying the fitness functions and changing the proximity values as distance or non-decreasing similarity function according to the requirements of the dataset to be clustered.

After running MOKGA, we get the pareto-optimal front that gives the alternative solutions. Then, the system analyzes the clustering results by applying six of the cluster validity indexes proposed in the literature, namely Silhoutte, C, Dunn, SD, DB and S_Dbw. The favored number of clusters guides the process in selecting the most compact clusters from the pool.

The employed clustering approach MOKGA is basically the combination of the Fast Genetic K-means Algorithm (FGKA) [7] and Niched Pareto Genetic Algorithm [5]. MOKGA uses a list of parameters which has nothing to do with the clustering process; these parameters are particular to the process of the genetic algorithm: population size (number of chromosomes), t_dom (number of comparison set) representing the assumed non-dominated set, mutation probability and the number of iterations that the execution of the algorithm needs run in order to report the result. Sub-goals can be defined as fitness functions; instead of scalarizing them to find the goal as the overall fitness function with the user defined weight values, we expect the system to find the set of best solutions, i.e., the pareto-optimal front. By using the specified formulas, at each generation, each chromosome in the population is evaluated and assigned a value for each fitness function.

The coding of our individual population is a chromosome of length N (number of data points). Each allele in the chromosome takes a value from $\{1, 2,..., K\}$, and represents a pattern. The value indicates which cluster the corresponding pattern belongs to.

2.1 The Objectives

The multi-objective genetic algorithm considers four objectives, namely separateness, homogeneity, number of clusters, and cluster density. As separateness and homogeneity are concerned, we used the following formulas. For separateness, we used the inter-cluster separability formulas described next, where P

and R denote clusters and $|P|$ and $|R|$ are the cardinalities of the aforementioned clusters; $d(x, y)$ is the distance (similarity) metric where $x \in P,\ \ y \in R$ and $P \neq R$.

Average Linkage between two clusters is the average of pairwise distances. The cardinalities of P and R may be omitted to reduce the scaling factor. It is computed as:

$$D(P, R) = \frac{1}{|P|.|R|} \sum_{\substack{x \in P,\\ y \in R}} d(x, y)$$

Centroid Linkage is the distance between the centroids v_P and v_R of the two clusters P and R. It is computed as: $D(P, R) = d(v_P, v_R)$

For homogeneity, we used the intra-cluster distance formula, namely **Total Within Cluster Variation** (TWCV), which calculates the intra-cluster distance of the cluster by the following formula:

$$TWCV = \sum_{n=1}^{N} \sum_{i=1}^{S} X_{nd}^2 - \sum_{k=1}^{K} \frac{1}{Z_k} \sum_{d=1}^{S} SF_{kd}^2$$

where S is the number of features, $X_1, X_2, ..\ , X_N$ are N objects, X_{n_i} denotes feature i of pattern X_n (n= 1 to N); SF_{k_i} is the sum of the i^{th} features of all the patterns in cluster $k(G_k)$; Z_k denotes the number of patterns in cluster $k(G_k)$. Actually, SF_{k_i} is computed as:

$$SF_{k_i} = \sum\nolimits_{\overrightarrow{x_n} \in G_k} X_{n_i}\ , \quad (i = 1, 2, ...S).$$

The objectives are utilized in the process as minimization; the separateness value is multiplied by -1 for the minimization. After that, the objectives are normalized by dividing their values by the corresponding maximum values.

2.2 The Multi-Objective Genetic Algorithm

Deciding on the encoding scheme is the first essential step of the genetic algorithm process, and directly affects the whole setup. In this sense, integer encoding is used, and individual coding in the population is a chromosome of length n, the number of instances in the set to be clustered. The genetic algorithm process involves the steps outlined in Algorithm 1.

Algorithm 1 (Genetic Algorithm).

```
*current generation* is assigned to zero,
Generate $M$ initial individuals (interchangeably called chromosomes);
**For** each chromosome
    In round order, each gene takes values 1 to $k$ (inclusive) in order. Once
    value $k$ is assigned to a gene the count continues again from 1 to $k$
    Shuffle the allele value assignments within the chromosome randomly by processing
    the random pairwise swap operation inside the chromosome.
```

```
EndFor
Repeat
  Apply selection using pareto domination tournament and the crowding measure
  of NSGA-I
  If (candidate $x_i$ has TWCV fitness value and number of clusters value are
    both larger, and separateness value is smaller than those of all of the
    chromosomes in the comparison set) then
      Candidate $x$ is dominated by the comparison set
      Delete $x$ from the population permanently.
  Else
      $x$ resides in the population.
  While (not all chromosomes are selected)
      Choose two chromosomes $x$ and $y$ randomly
      Apply one-point crossover operator on $i$ and $j$ with probability $p_c$ to produce two
      new chromosomes $x_{new}$ and $y_{new}$
      Keep track of $x$, $y$, $x_{new}$ and $y_{new}$ for the new generation.
  EndWhile
  Apply mutation operator on the current population to guarantee better convergence.
  Compute the fitness of each chromosome
  Rank all the chromosomes (new and old) based on their fitness values
  Keep in the new generation only the best $M$ individuals
Until either the difference between the last two generations satisfies the threshold
      value or the prespecified maximum number of generations is reached.

EndAlgorithm
```

After getting the pareto-front and deciding on the most appropriate number of clustering k using validity analysis, the alternative solutions are collapsed into a pool of all clusters. Of course, a given object belongs to n clusters in the pool where n is the number of alternative solutions. Further, it is not necessary that all clusters in the best solution reported by the validity analysis have best TWCV value compared to the rest of the clusters in the pool. This leads to ranking clusters in the pool based on their TWCV and selecting from the pool only the top k clusters which have the best TWCV. As it is not guaranteed to have the k clusters coming from the same solution, it is possible to have some objects exist in more than one of the selected k clusters. Also, it is possible for some objects not to show up in any of the selected k clusters. For the former case, we compute the distance between each object and the centroid of every cluster to which the object belongs. As a result, every object survives only in the cluster that satisfies the minimum distance. At the end, objects that do not belong to any of the identified compact clusters are classified into two sets: some of them join the existing compact clusters if they are not destroying the compactness; the rest of the objects are classified as outliers. The conducted experiments did not report any outliers for the utilized two benchmark data sets. We will run the proposed approach on some other data sets (including some synthetic data) that do report some outliers; this will give us better insight into the power of the proposed approach in identifying real outliers.

3 Experiments

To evaluate effectiveness of the developed clustering approach, experiments were conducted using two benchmark datasets, namely Glass and Lukemia; this demonstrates the applicability of the proposed approach to various domains. The system was implemented using Visual C++. The running platform is Microsoft Visual Studio.NET.

3.1 Testing with Glass

In the first experiment, we used a real dataset Glass, which refers to the glass identification database that studies the classification of the types of glass left at the crime scenery for criminological investigation. It has 9 continuous features, 214 examples and 6 classes (70 building windows, 17 vehicle windows, 76 non-float processed, 76 building windows, 0 vehicle windows, 51 Non-window glass, 13 containers, 9 tableware, 29 headlamps).

MOKGA has been run for the Glass dataset with the following parameters: population size = 100, tdom (number of comparison set = 10) and crossover = 0.8, mutation = 0.005, gene mutation rate= 0.005, and threshold = 0.0001 which is used to check whether the population stops evolution for 50 generations or the process needs to be stopped. The range of [2, 15] was picked to find the optimal number of clusters.

Table 1. Glass dataset cluster validity results

	WCV & Centriod		*TWCV & Average*	
	best	*2^{nd}-best*	*best*	*2^{nd}-best*
dunn	*6*	*5*	*4*	*3*
Davies-Bouldin	*6*	*5*	*6*	*7*
Silhouette	*6*	*7*	*6*	*7*
C	*3*	*5*	*3*	*6*
SD	*6*	*5*	*6*	*7*
S_Dbw	*6*	*7*	*6*	*7*

The collapse and select approach works as follows on Glass. First, the multi-objective genetic algorithm is applied. Then, the majority voting is done to decide on the most appropriate results. The results are reported in Table 1, with the best and second best number of clusters for each of the utilized indexes; these values range between 3 and 7. This demonstrates that not every index works the same for each dataset; some indexes are more successful than others. But from experience and by empirical testing, we realized that there is a trend to favor the most appropriate number of clusters as the number of indexes utilized in the process increases. As a result, 6 is reported as the best number of clusters by majority voting. Then the best 6 clusters are selected from the pool of clusters coming from all the solutions. We realized that three of the six clusters are from the most appropriate solution, two clusters are from the next appropriate

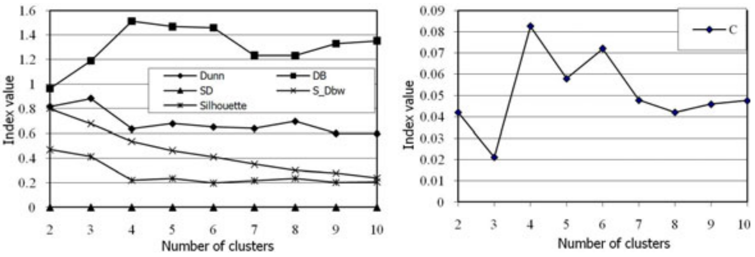


Fig. 1. Leukemia dataset cluster validity results

solution and one cluster is from the fourth solution. Overall the produced final solution ranks better than the solution which was selected from the pareto-optimal front as the most appropriate.

3.2 Testing with the Leukaemia Dataset

The second experiment uses the Leukemia dataset, which has 38 acute leukemia samples and 50 genes. The purposes of the testing include clustering cell samples into groups and finding subclasses in the dataset.

The proposed genetic algorithm-based approach has been run for the Leukemia dataset with the following parameters: population size = 100, tdom (number of comparison set = 10) and crossover = 0.8, mutation = 0.005, gene mutation rate = 0.005, and threshold = 0.01 which is used to check if the population stops evolution for 50 generations and if the process needs to be stopped. The range of [2, 10] was picked for finding the optimal number of clusters.

According to the curves plotted in Figure 1, the validity analysis results for the Leukaemia dataset are consistent with the literature where it is indicate that 2 (AML and ALL) is the best number of clusters; this two clusters as the best results has been concurrently reported by Dunn index, DB index, SD index, C index and Silhouette and 3 (AML, B-cell ALL and T-cell ALL) has been reported the second best. By analyzing the results from the validity indices further, we discovered that S_Dbw is an exception; it may not be suitable to test small datasets with fewer than 40 instances. In our attempt to produce the best solution one cluster was selected from each of the two best solutions.

4 Summary and Conclusions

In this paper, we proposed a new clustering approach which depends on MOKGA as a multi-objective genetic algorithm based clustering approach. MOKGA is a combination of the Niched-Pareto optimal and fast k-means genetic algorithm. This way, we overcome the difficulty of determining the weight of each objective function taking part in the fitness. Otherwise, the user would have been expected to do many trials with different weighting of objectives as in traditional genetic algorithms. By using MOKGA, we aim at finding the pareto-optimal front so that the user will be able to see at once all possible alternative solutions identified by the system; then cluster validity index values are evaluated for each pareto-optimal front value which is the number of clusters value considered to be optimal. Then the solutions are all collapsed into a single pool of clusters which are individually evaluated to identify the most compact clusters to form the final solution. Comparing the clusters in the final solution produced by the proposed clustering approach with the ones in the best clustering solution reported by the validity analysis, we realized that the former clusters are all compact and well separated while compactness of the latter clusters vary as well as their separateness. To validate the propose approach better, we still need to run more tests for data from different domains and with different characteristics. The outcome from this research project has interesting characteristics and it is very essential for several applications. The user is no more in need for expertise in the domain of the data to be clustered because number of clusters is not needed but determined by the system. The process does not suffer from local minima kind of drawbacks because it leads to the most natural distribution of the data instances into the clusters leading to the most compact and separable clusters.

References

1. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Transactions on Pattern Recognition and Machine Intelligence (1), 224–227 (1979)
2. Dunn, J.: Well separated clusters and optimal fuzzy partitions. J. Cybernetics 4, 95–104 (1974)
3. Halkidi, M., Vazirgiannis, M., Batistakis, Y.: Quality scheme assessment in the clustering process. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. PKDD, vol. 1910, pp. 265–276. Springer, Heidelberg (2000)
4. Halkidi, M., Vazirgiannis, M.: Clustering Validity Assessment: Finding the optimal partitioning of a data set. In: Proceedings of IEEE ICDM, California (November 2001)
5. Horn, J., Nafpliotis, N., Goldberg, D.E.: A niched pareto genetic algorithm for multiobjective optimization. In: Proceedings of IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Computation, Piscataway, NJ, vol. 1, pp. 82–87 (1994)
6. Hubert, L., Schultz, J.: Quadratic assignment as a general data-analysis strategy. British Journal of Mathematical and Statistical Psychology 29, 190–241 (1976)
7. Lu, Y., Lu, S., Fotouhi, F., Deng, Y., Brown, S.: FGKA: A Fast Genetic K-means Clustering Algorithm. In: Proceedings of ACM Symposium on Applied Computing, Nicosia, Cyprus, pp. 162–163 (2004)
8. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comp. App. Math. 20, 53–65 (1987)
9. Özyer, T., Alhajj, R.: Parallel Clustering of High Dimensional Data by Integrating Multi-Objective Genetic Algorithm with Divide and Conquer. Applied Intelligence (in press)
10. Tan, M., Alshalalfa, M., Alhajj, R., Polat, F.: Influence of Prior Knowledge in Constraint-Based Learning of Gene Regulatory Networks. In: IEEE/ACM TCBB, vol. 8(1), pp. 130–142 (2011)

A New Approach to Neural Network Based Stock Trading Strategy

Miroslaw Kordos and Andrzej Cwiok

University of Bielsko-Biala, Department of Mathematics and Computer Science,
Bielsko-Biala, Willowa 2, Poland
mkordos@ath.bielsko.pl

Abstract. The paper presents an idea of using an MLP neural network for determining the optimal buy and sell time on a stock exchange. The inputs in the training set consist of past stock prices and a number of technical indicators. The buy and sell moments on the training data that will become the output to the neural network can be determined either automatically or manually by a user on past data. We discuss also the input space transformation and some improvements to the backpropagation algorithms.

Keywords: neural networks, stock prediction.

1 Introduction

Prediction of stock prices is considered to be a very difficult problem. However, what we really need to trade effectively is not to predict the future stock price but the optimal moment to buy or sell the stock. On one hand machine learning methods can be used to predict the stock price and on the other hand technical analysis indicators determine the optimal transaction points. To all of that the human factor is added since it is the user of the trading software, who can finally decide whether to agree or not with the program decision.

In our approach we try to take into account the best features of all the three sources (machine learning, technical analysis and human factor) together to build a robust trading system.

The first market hypothesis was that stock prices follow a random walk [1]. However, researchers and economists were able to extract rules associated with the stock price movements. That kind of rules can significantly increase the quality of their predictions [1,2].

Since stock prediction is a complex, non-deterministic issue, it is a good ground for the application of artificial intelligence. During the last two decades, stocks and futures traders stared to believe in the decisions made by various types of intelligent systems, based on different algorithms. The list of implemented solutions includes among others: genetic algorithms used to find optimal values of various technical indicators and their combinations thus reducing the computational complexity of the search [3], fuzzy logic controllers and fuzzy neural

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 429–436, 2011.

networks [4,5], Hidden Markov Models to assess the parameters of a Markov process that best fits the stock price fluctuations [6]. Neural networks were the most widely used methods, because of their superior performance [7,8,9,10,11,12].

However, even the best model is only as good as the data it is trained on. In stock price prediction selection of the right input variables is a much more difficult problem than building and training the predictive model itself. In our work we choose to use MLP neural networks as the model and focus on searching for an optimal set of input variables, which allows to maximize the gains in the trade rather than only predict the stock price most accurately. We also discuss a simple improvement to the backpropagation algorithm.

2 Input Variables

In this section we describe the indicators we use as input variables and how we transfer and combine them to generate the training set for the neural network as well as the way of determining output variables. Technical indicators indicates the time when one should buy or sell stocks and by default they use daily closing prices. Only the candle formations use the opening, daily minimum, maximum and closing price. The indicators are only shortly listed here due to limited space. An in-depth analysis can be found in [13].

Simply Moving Average (SMA) shows the average value of stock price over a period of time, while Exponential Moving Average (EMA) pays more attention to recent prices:

$$SMA(t) = \frac{1}{n}\sum_{i=0}^{t} C_i \tag{1}$$

$$EMA(t) = \frac{2}{n+1}C_t + (1 - \frac{2}{n+1}) \cdot SMA(t) \tag{2}$$

where C_t is a stock closing price at day t. 15 days SMA is frequently the default. A moving average crossing the actual price from down when the price rises generates a buy signal, while a MA crossing the price from top generates a sell signal. Moving Average Convergence/Divergence (MACD) is a difference between two exponential moving averages: a shorter and a longer one (typically 12 and 26 days):

$$MACD(t) = EMAshort(t) - EMAlong(t) \tag{3}$$

Momentum oscillators measure the speed of change of a stock price. As the price is rising/falling, the momentum increases/decreases proportionally to the speed of the price rise/fall. Relative Strength Index (RSI):

$$RSI = 100 - [100/(1 + RS)] \tag{4}$$

where RS = (Average Gain of n-day up)/(Average Loss of n-day down). A high RSI means the market is rallying suddenly and a low RSI that the market is selling off suddenly. Rate of Change (ROC) indicator is at a high peak and is beginning to move down generates a sell signal. A ROC at a low peak that is beginning to move up is a buy signal. The advantage of a ROC oscillator in comparison to moving average based indicators is that it gives signals before the actual change in the direction of a stock price occurs.

$$ROC = [(Today'sclose - CloseNdaysago)/(CloseNdaysago)]) \cdot 100 \tag{5}$$

The Commodity Channel Index (CCI) is designed to identify cyclical turns in commodities. It is recommended to use 1/3 of a complete cycle as a time frame for the CCI. If the cycle runs 60 days (a low about every 60 days), then a 20-day CCI would be recommended. For the purpose of this example, a 20-day CCI is used:

$$CCI = (TP - SMATP)/(0.015 \cdot MD) \tag{6}$$

where : Typical Price (TP) = (H+L+C)/3 where H = high, L = low, ad C = close, SMATP is a 20 day-period SMA of the Typical Price (SMATP), MD is calculated as the mean deviation of the TP over the past 20 periods. From oversold levels, a buy signal is given when the CCI moves back above -100. From overbought levels, a sell signal is given when the CCI moved back below +100. The Average True Range (ATR) is calculated with the following steps: multiply the previous 14-day ATR by 13, add the most recent day's TR value, divide by 14. Extreme levels (both high and low) can mark turning points or the beginning of a move. Stochastic Oscillator (STO) is a momentum indicator that shows the location of the current price relative to the high/low range over a set number of periods. Price levels that are consistently near the top of the range indicate accumulation (buying pressure) and those near the bottom of the range indicate distribution (selling pressure).

$$STO = 100 \cdot (RecentClose - LowestLow)/(HighestHigh - LowestLow) \tag{7}$$

We use also two candle formation in our analysis, which shows the reversion of the current trend. A bullish hammer, which is a buy signal that occurs after an established downtrend and a bearish shooting star, which is the opposite of the hammer. Both formations indicate that the price extremum was reached during the day and the trend is now likely to reverse [14].

The training set consists of the following 15 features:

- Price change in 1,2 and 3 days in relation to the current price
- SMA with periods of 6, 9, 14 and 21 days
- ROC with periods of 6, 9, 14 and 21 days
- RSI with periods of 6, 9, 14 and 21 days
- CCI with periods of 6, 9, 14 and 21 days
- STO with periods of 6, 9, 14 and 21 days
- ATR with periods of 6, 9, 14 and 21 days
- The candle formation (hummer and shooting star)

That may seem an excessive number of parameters, but as our experiments showed it gives significantly better results than using only a single length period indicators (e.g. 14 days only). Moreover, neural networks have the ability to perform feature selection by setting the weights that connect the less important features to the network to very low values. While most other works try to predict the stock price, we try only to predict the optimal moment of buying and selling stocks.

The output of the training set can be determined automatically, but our software also enables the user to determine the output manually by choosing the optimal buy and sell moments, which will become the outputs in the training data set. In this way the user can balance different trading strategies, mostly the gain-risk trade-off according to their preferences.

We found that it very difficult to obtain good results when the network was trained on the exact data (that is with one day resolution). The result of that was, that different technical analysis indicators give signals in different days. Usually the difference is one or two days. However, if we consider only single day signals, they very rarely are enough strong to cause the neural network to generate a buy or sell signal.

Therefore if we determine the best day to buy or to sell stocks, first we shift the signal one day earlier (it gives better results) and then we spread the buy/sell signals (the network targets) two days backward and two days forward. However, the signal had the value of one on a given day, of 0.67 on the day before and after and of 0.33 on the days two day apart from the current date. We consider the network buy or sell signal correct if it appears on o given day or up to two days before or after. We do not discuss here the choice of the right stock to trade.

3 Neural Network Architecture and Training

We use two separate networks. One is trained to recognize the buy signals and the other one to recognize sell signals. When the stock is once bought it is kept until the first sell signal occurs, even if there are more buy signals in the meantime. We use a multilayer perceptron with two hidden layers and with hyperbolic tangent activation functions in all layers, including the input one [15]. A good practice is to standardize the data before the training to make particular inputs independent of their physical range. It may be also be beneficial to remove the outliers from the training set. Moreover, a model with higher sensibility in the intervals with more dense data may sometimes be preferred. To address the problem, the idea of transforming the data to make it distributed more evenly was proposed. For example, the data can be transferred by a hyperbolic tangent function.

The other advantage of such a transformation is the automatic reduction of the outliers' influence on the model. We do not consider the outliers as erroneous values and thus do not reject them, but rather reduce their influence on the final model. The neural network can either learn the optimal slopes of the transfer function during the training or they can be set a priori.

We tried three learning algorithms: standard backpropagation with momentum, modified backpropagation and variable step search algorithm [16]. In the

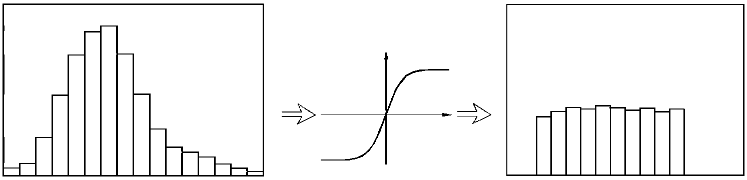


Fig. 1. The idea of transforming data from a Gaussian-like distribution to uniform distribution

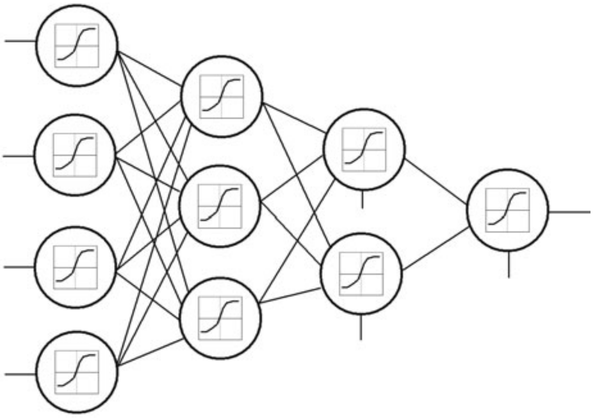


Fig. 2. Neural Network Architecture

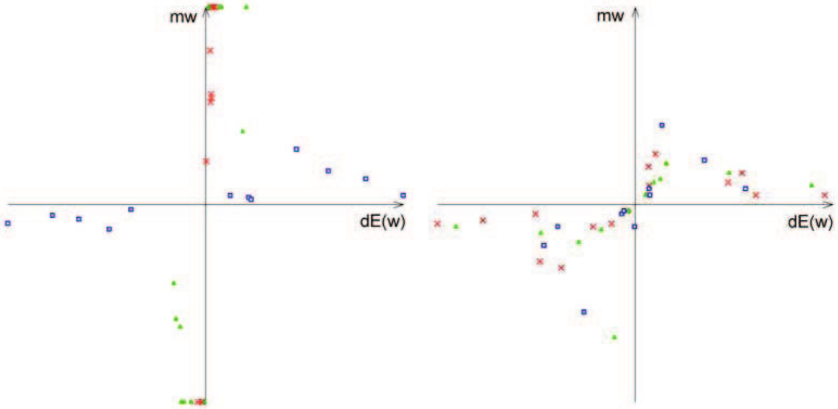


Fig. 3. Dependence between the gradient component dE(w) and the distance from the actual point actual point to the error minimum in a given weight direction *mw* at the beginning (left) and at the end of the training (right) cross = first hidden (counting from input), triangle = second hidden, square = output layer)

modified backpropagation we used a directional minimization and a change of the direction by making the gradient components not linearly proportional to that given by the backpropagation algorithm, but rather proportional to the root square of them. Moreover, we multiplied the gradient component in the first hidden layer by four and in the second hidden layer by two to better optimize the trajectory direction (Fig. 3.) without reverting to the second order methods. Based on the experiments we conducted with several datasets as well for classification as for regression tasks (the current task is closer to classification) the following formula can be used to optimize the next step length $dS(w)$ in each weight direction in the backpropagation algorithm:

$$dS(w) = (1 + a \cdot \exp(-bT_c)) \cdot sign(dE(w))) \cdot \sqrt{(|dE(w)|))} \tag{8}$$

where a=0 for the output layer, $a = 4 \pm 1$ for the second layer (closer to the output), $a = 15 \pm 5$ for the first hidden layer weights, $b = 0.12 \pm 0.3$. $dE(w)$ is the gradient component in weight w direction and T_c is the training cycle (epoch). The simplest explanation of the equation is contained in Fig. 3.

4 Experimental Results

We created software for this project as a Windows Forms Application in C#. The application including the source code can be downloaded from our web site [18]. Using that application users can select with a mouse the moments used to train the network and can also obtain the results in a graphical form. We performed ten experiments with four stocks of the USA market: Amazon, Apple, Microsoft and Yahoo, the average results are reported in Table. 1. The training set comprised the data from 1/1/1995 to 12/31/2004 and the test set the data from 1/1/2005 to 1/1/2008.

We performed some simple tests in the Amibroker software [17], where we optimized the period of moving average (opt. SMA) on the training sets and then performed tests on test data. Then we compared the results also with the typical 15 day SMA and with a buy and hold strategy. However, as it can be seen from Table 1, the optimization of SMA does not cause any improvement in the prediction results. Sample signals, which the network generated on the Microsoft stock are shown in Fig 4. Only the MLP output signals that have the value of 0.8 or more are interpreted as effective buy or sell signals and only these signals are shown.

Table 1. The profit (in percent respect to the price on 1/3/2005) for the test period on four stocks

stock	APPL	MSFT	YHOO	AMZN
buy and hold	518	33	-39	108
opt. SMA	12	17	-36	-1
SMA15	116	10	-20	8
Our Method	425	53	14	184

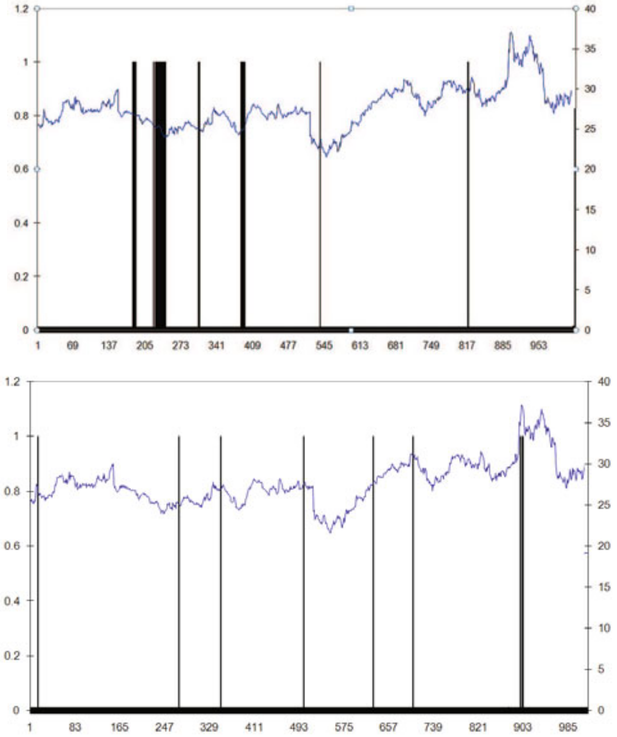


Fig. 4. Closing prices for Microsoft stock with buy (top) and sell (bottom) signals generated by the neural network

5 Conclusions

We presented a method for determining optimal buy and sell moments at stock exchange using a neural network based prediction. The output of the training set is either chosen automatically or determined by a user, who chooses the optimal moments themselves. The input consists of past prices and a number of technical indicators with 6, 9, 14 and 21 days each. The method seems to be a very interesting approach that allows for including many different factors as well past prices and technical indicators, as some other information (e.g. financial condition of the company). We are currently working a feature selection method, which seems to be the most important task in improving the method.

Also the modification to the backpropagation direction connected with an approximate linear search along the modified step direction makes the backpropagation algorithm much more efficient in the terms of better convergence and convergence in cases where the standard backpropagation algorithms cannot find a satisfactory solution.

References

1. Gencay, R.: Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules. Journal of International Economics 47, 91–107 (1999)
2. Zhou, W.X., Sornette, D.: Testing the stability of the 2000 US stock market antibubble. Physica A: Statistical and Theoretical Physics 348(15), 428–452 (2005)
3. Che, S.-H.: Computationally intelligent agents in economics and finance. Science Direct 117(5), 1153–1168 (2007)
4. Gradojevic, N.: Non-linear, hybrid exchange rate modeling and trading profitability in the foreign exchange market. Journal of Economic Dynamics and Control 31(2), 557–574 (2007)
5. Chang, P.-C., Liu, C.-H.: A TSK type fuzzy rule based system for stock price prediction. Expert Systems with Applications 34(1), 135–144 (2008)
6. Hassan, M.R., Nath, B.: StockMarket Forecasting Using Hidden Markov Model: A New Approach. In: ISDA, pp. 192–196 (2005)
7. Tilakaratne, C.D., Mammadov, M.A., Morris, S.A.: Development of Neural Network Algorithms for Predicting Trading Signals of Stock Market Indices
8. Khan, A., et al.: Stock Rate Prediction Using Backpropagation Algorithm: Results with Different Numbers of Hidden Layers. Journal of Software Engineering (1), 13–21 (2007)
9. Mpofu, N.: Forecasting Stock Prices Using a Weightless Neural Network. Journal of Sustainable Development in Africa 8 (2006)
10. Zorin, A.: Stock Price Prediction: Kohonen Versus Backpropagation. In: Proceedings of the International Conference on Modeling and Simulation of Business Systems, Vilnius, Lithuania, pp. 115–119 (2003)
11. Afolabi, M., Olude, O.: Predicting Stock Prices Using a Hybrid Kohonen Self Organizing Map (SOM). In: Proceedings of the 40th Hawaii International Conference on System Sciences (2007)
12. Klassen, M.: Investigation of Some Technical Indexes inStock Forecasting Using Neural Networks. Proceedings of World Academy of Science, Engineering and Technology 5 (2005)
13. http://stockcharts.com/school/doku.php?id=chart_school
14. http://forex-candelistic.blogspot.com/2008/09/trade-with-candlestick-part-4.html
15. Kordos, M.:Neural Network Regression for LHF Process Optimization. LNCS, vol. 5506, pp. 453–460. Springer, Heidelberg (2009)
16. Kordos, M., Duch, W.: Variable step search algorithm for feedforward networks. Neurocomputing 71(13-15), 2470–2480 (2008)
17. http://www.amibroker.com
18. Software used in this work, http://www.kordos.com/pdf/nnstock.zip

A Comparison of Dimensionality Reduction Methods Using Topology Preservation Indexes

Claudio J.F. de Medeiros[1], José Alfredo Ferreira Costa[1], and Leandro A. Silva[2]

[1] Departament of Electrical Engineering, Federal University, UFRN, Brazil
c.j.franca@ig.com.br, alfredo@ufrnet.br

[2] School of Computing and Informatics, Mackenzie Presbyterian University, S. Paulo, Brazil
prof.leandro.augusto@mackenzie.br

Abstract. Due to the remarkable technological developments experienced in recent decades, the vast amount of data had created new opportunities and challenges in the field of knowledge discovery and data mining. Factors like size and high dimensionality of databases adds difficulties to the complex task of discovering patterns hidden in masses of data. The feasibility of high-dimensional data exploration depends on techniques known as dimensionality reduction methods. When class labels are available, an optimization function can be used to maximize intra class cohesion and inter class separation. However, in many practical situations information about class is not available. This paper focuses on unsupervised dimensionality reduction techniques, an important phase in exploratory data analysis. Six important methods are described: Principal components analysis, Sammon projection, Auto-associative Neural network, Kohonen maps, Isomap and Locally Linear Embedding. Three quality indexes are proposed to try to quantify to some degree the topology preservation between input and output spaces. Comparisons are performed using benchmark data sets. Results and tests focused two-dimensional projections for data visualization purposes.

Keywords: dimensionality reduction, projections, data visualization, unsupervised methods, neural networks, data mining, intelligent systems.

1 Introduction

Dimensionality reduction (DR) is a fundamental operation to enable visualization of multidimensional data in the processes of knowledge discovery and data mining. DR methods are based on transformations on high dimension data projecting them into smaller spaces trying to keep topological relations among them [1]. These methods are used for various purposes, mainly as a preprocessing stage for other algorithms (e.g., classification or data clustering). An important application of DR is visualization of multidimensional data, typically used in exploratory data analysis to uncover clues about the structure of unlabeled data sets. Detailed and useful reviews of DR methods can be found in [13], [14], [15].

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 437–445, 2011.

Given the diversity of existing DR methods, this paper presents an empirical comparison of selected methods: Principal Component Analysis (PCA) [2]; Sammon projection [11]; Auto-associative Neural Network (AANN) [4]; Kohonen Self-Organizing Maps (SOM) [5]; Isomap [6] and Locally Linear Embedding (LLE) [7]. PCA and Sammon are traditional methods. AANN and SOM are two classic and widely used connectionist methods. Isomap and LLE are two relatively new methods based on manifold preservation. The techniques were applied using synthetic and real databases.

The work focuses particularly on the projections in two dimensions that can be viewed directly in two-dimensional graphics. The idea is to compare the methods among themselves, using indexes that aim to evaluate the preservation of data relations, with their neighborhoods, of the original space. The rest of the article is composed of the following sections: Section 2 briefly reviews the methods used. Section 3 describes the proposed indexes used to compare the techniques and section 4 presents results and analysis. Section 5 concludes the paper with final remarks.

2 Methods for Dimensionality Reduction

The Sammon projection is a variety of Multidimensional Scaling (MDS), a traditional method for DR. MDS can be defined as the search for a set of points in a lower dimension space in which each point represents an object of the *n*-dimensional space. The aim of the operation is to make distances between points in the reduced space the most similar as possible to the distances (dissimilarities) between corresponding points in the input space [3]. The Sammon algorithm uses Newton's second order method to minimize a cost function, representing the global error of interpoint distances between the input and output spaces.

Principal Component Analysis (PCA) [2] is a classical method of linear projection, still widely used in dimensionality reduction. The method applies a linear transformation on a set of *n*-dimensional data searching to represent it in a new coordinates system so that the projection of the largest variance of all possible input data coincides with the first axis of this new system (called the first principal component), the second largest variance, with the second orthogonal axis and so on, for *n* new axes. PCA can be used to obtain a reduction of an original dimension n to a lower dimension *m* by selecting the *m* first principal components of a given data set and ignoring the less important ones.

Autoassociative networks (AANN) [4] are MLP (Multi-Layer Perceptron) artificial neural networks, presenting a particular type of architecture. The use of such networks in reducing the dimensionality of *n*-dimensional data is obtained by imposing a reduced number of neurons, *m*, in the hidden (or central) layer. The same data are applied to input and desired output (*n* features). Usually there are at least two hidden layers with nonlinear activation functions between the central layer and outer layers. The number *m* represents the desired output dimension. On a trained network, such a type of architecture provides a mechanism for reducing dimensionality, the projection of each input point been represented by the outputs of the *m* neurons of the middle layer.

The Self-Organizing Map (SOM) defines a mapping of an *n*-dimensional continuous data to a finite set of vectors, or neurons, usually arranged as a regular two-dimensional lattice. The main goal of SOM training is to reduce the dimensionality of the input patterns trying to preserve at most of the topology of the input space [5]. The lattice of the 2-D grid is either hexagonal or rectangular. Assuming that each input pattern $\mathbf{x}_i$ from the set of patterns (X) is defined as a real vector $\mathbf{x}_i = [x_{i1}; x_{i2};\ldots; x_{in}]^T \in \Re^n$. Each neuron has a d-dimensional weight vector $\mathbf{w}_u = [w_{u1};w_{u2}; \ldots;w_{un}]^T \in \Re^n$, called a unit or prototype. Therefore, the proximity of points projected to the same neuron or neighboring neurons in the trained map tends to represent the similarity of the corresponding objects in the original space.

The Isomap algorithm, proposed by Tenenbaum *et al.* [6], can be seen as an extension of the MDS method in which the dissimilarity between objects is represented by the geodesic distance, in principle more appropriate than the Euclidean distance. Given an input space, *X*, the matrix of distances $d_X(i, j)$ between all pairs of points $<x_i,x_j>$ is obtained. The neighborhood of each point x_i is found using its k nearest neighbors. Then a graph *G* is designed from all points of space *X*, by interconnecting each point to its neighbors, the edges weights being $d_X(i, j)$. The algorithm estimates the geodesic distance between each pair of points, calculating the shortest path $d_G(i, j)$ between these points in the graph *G*. In order to obtain a Cartesian projection of the input space into a smaller dimension, the classical MDS cost function is minimized by the gradient method. In this case, the distance d_{ij} between two points in the original space is replaced by its geodesic distance $d_G(i, j)$.

Locally Linear Embedding method (LLE), proposed by Roweis and Saul [7], seek a global projection of the data, capturing the local characteristics of the manifold. LLE models the manifold by treating it as a union of several parts and assuming that the data stay on a manifold and that this manifold presents approximate local linearity in the proximity of each input point. Each data point $\mathbf{x}_\mathrm{i}$ Є R^n has a number of nearest neighbors indexed by the set *N*(*i*). Let $\mathbf{y}_\mathrm{i}$ Є R^m be the representation of $\mathbf{x}_\mathrm{i}$ in the low dimension. The idea is to express each $\mathbf{x}_\mathrm{i}$ as a linear combination of its neighbors and then reconstruct it as $\mathbf{y}_\mathrm{i}$ such that $\mathbf{y}_\mathrm{i}$ is expressed by the same linear combination of its corresponding neighbors, indexed by *N*(*i*).

3 Proposed Indexes for Empirical Comparison of DR Methods

This paper proposes some indexes for empirical comparison of DR methods. These indexes, by means of simple and intuitive definitions, aim to assess the degree of neighborhood relationships preservation between original and projected data on the reduced space as an approximate evaluation of topology preservation.

The unordered neighborhoods coincidence (UNC) index, c*(k)* is defined as

$$c(k) = \frac{1}{pk}\sum_{i=1}^{p} u_1(i,k) \ , \tag{1}$$

where *k* is a free parameter that defines the size of the considered neighborhood around each point, i.e., the number of nearest neighbors of each point, either in the input or output space; *p* is the number of points of the data set (cardinality); *i* is the index indicating the position of each point within that data set; $u_1(i, k)$ is a function that quantifies the index for each point $\mathbf{x}_\mathrm{i}$ in the input space (high dimension) compared to corresponding point $\mathbf{y}_\mathrm{i}$ in output (reduced) space. The function $u_1(i, k)$ depends on the size of the neighborhood and is defined for each point $\mathbf{x}_\mathrm{i}$ as $u_1(i, k) =$ # $\{\mathcal{V}_\mathrm{k}\ (\mathbf{x}_\mathrm{i}) \cap \mathcal{V}_\mathrm{k}\ (\mathbf{y}_\mathrm{i})\}$, where: #{A} is the operation that computes the cardinality of A; $\mathcal{V}_k$ ($\mathbf{x}_\mathrm{i}$) is the neighborhood with size k from point $\mathbf{x}_\mathrm{i}$, the set composed of the indexes of the points that are the k nearest neighbors of $\mathbf{x}_\mathrm{i}$ in the input space; and $\mathcal{V}_k(\mathbf{y}_\mathrm{i})$, the equivalent to the point $\mathbf{y}_\mathrm{i}$ of the output space.

The UNC index has a simple interpretation: for a neighborhood size k, previously defined, the index reflects the number of neighbors of the points of the input space that once projected, are still neighbors of the corresponding points in the output space, respectively. The index $c(k)$ can range from 0 to 1, being 1 the best case within the limits of the definition.

The ordered neighborhoods coincidence (ONC) index, $o(k)$, is defined in a similar fashion of UNC, however, with the modification in the function $u(i, k)$ to reflect the requirement of order preservation within the vicinity of the output points. Thus, we have:

$$o(k) = \frac{1}{pk}\sum_{i=1}^{p} u_2(i,k)\ , \tag{2}$$

being kept the same previous settings for the parameters p, k, i and for the sets $\mathcal{V}_k$ ($\mathbf{x}_\mathrm{i}$) and$\mathcal{V}_k(\mathbf{y}_\mathrm{i})$. Considering a certain point ordered i, taking only indexes that appear in the neighborhood $\mathcal{V}_k$ of that point in the input as well as in the output space at same time, two subsets are obtained containing the same elements (indexes), but not necessarily arranged in the same order. The value of the function $u_2(i, k)$ is then the number of perfect matches (values and position of the indexes) between these two subsets. It is noticed that this second index computes only the coincidence of the same neighbors in the same relative position.

The joint neighborhood coincidence (JNC) is a modified version of the ONC index applying a Bezdek and Pal [9] idea of using rank correlation coefficients to assess maintenance of topology. The average count of coincidences between the neighbors corresponding to each of the input and output data could be replaced by applying a ranking correlation coefficient between vector $\mathbf{r}$, grouping neighborhoods $\mathcal{V}_k(\mathbf{x}_\mathrm{i})$ of all input data points ($\mathbf{x}_\mathrm{i}$) and the corresponding vector $\mathbf{r}$*, grouping neighborhoods $\mathcal{V}_k(\mathbf{y}_\mathrm{i})$ of all output points ($\mathbf{y}_\mathrm{i}$). The desired index $g(k)$ is equivalent to ρ ($\mathbf{r}, \mathbf{r}^*$), where ρ is a rank correlation coefficient like those by Spearman or Kendall.

4 Experimental Results

The experiments were conducted in a computer with Aton 1.67 GHz processor, 2GB RAM and also using MATLAB software. Some functions of the SOM Toolbox [12] and DR toolbox [8] were used. The databases used in our experiments were Wine recognition data (Wine), Synthetic Control Chart Time Series (Control) and Quadruped Mammals (Animals). These data sets are available for research in the UCI machine learning repository [10]. These data are composed by continuous attributes and represent actual or simulated measurements of real-world objects. All experiments used unsupervised algorithms, consequently label informations were discarded in the learning process.

The methodology for evaluating the selected methods in this study followed the sequence: (1) Each selected method was applied to all databases (with proper data normalization) in order to obtain the projected data in two dimensions; (2) The indexes here presented were computed comparing input and output data sets; (3) The results are presented in plots that reflects the variations of index values as a function of the size of the neighborhood.

Hexagonal topology was used in output grid of SOM networks. Weights were initialized linearly and a limit of 1000 epochs was used in the first phase training. Initial neighborhood radius was set to 5 units and initial learning rate was 0.05. For the tuning phase the maximum number of epochs was set 10,000, an initial neighborhood radius of 5 units and the initial learning rate value was 0.01. In order to make the choice less arbitrary, the sizes of the maps used was: 11 x 6 for the Wine data base; 17 x 7 for the Control base; and 22 x 10 for the Animals data base. These sizes were obtained from estimative by using the formula $m = 5 \sqrt{} n$, where n is the number of samples in the set. For the ratio between the sizes of the map used is obtained from the two largest eigenvalues of the covariance matrix of the samples.

For LLE and Isomap methods, the only free parameter is the neighborhood size, k. In tests, the values were defined empirically by running the algorithms a number of times, using a wide range of values of k. The values chosen were the most favorable in each case. Thus, the values of k used were 12, 16 and 50 for the LLE method, and 12, 15 and 100 for Isomap, respectively to Wine, Control and Animals databases.

The architectures of AANN had 5 layers. The number of neurons of both the input and output layers as the hidden layers were determined according to the problem (dimension of input data and projected data). The architectures chosen were [13/4/2/4/13] neurons (Wine database), [60/5/2/5/60] (Control database) and [72/7/2/7/72] (Animals database). These architectures were defined empirically for each database through several tests to obtain the best relative convergence.

The Wine database is the result of a chemical analysis of samples of wine produced in the same region of Italy, but from three different cultures. The database has 178 samples with 13 continuous attributes and three classes.

The observation of results presented in Figures 1 and 2 shows the same trend for the unordered and ordered neighborhood indexes. The best performance was observed in the algorithms Sammon, Isomap and SOM, with SOM better than others. The methods PCA and autoassociative network lie in an intermediate group and the LLE method presented the worst performance. SOM and Isomap methods presented best results regarding the joint ordered coincidence index (Fig. 3), followed by PCA and Sammon methods, both presenting equivalent performance. Regarding this index, LLE and RNA presented worst index results, LLE being the less favorable.

The Control database consists of control charts synthetically generated by a simulated process control [11]. It contains 600 patterns, 100 per class, each representing a control chart with sampling at 60 regular sequential time intervals. It presents six classes for the different behaviors of the control system. Results are presented in Figures 4, 5 and 6. In these experiments, Isomap was discarded because it presented degenerated results, regarding topology preservation, when there is loss of an entire class. Considering the three indexes, the SOM presented overall superior performance. The other methods presented equivalent performance.

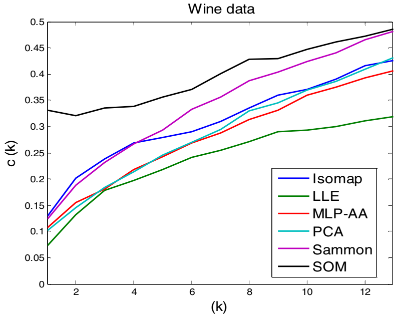


Fig. 1. Unordered coincidence index versus neighborhood size – Wine dataset

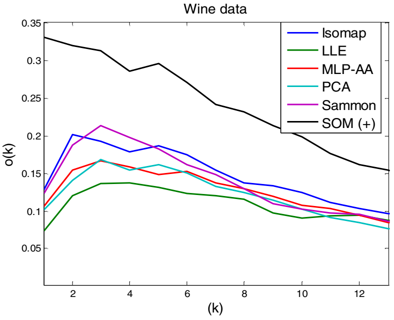


Fig. 2. Ordered coincidence index versus neighborhood size – Wine dataset

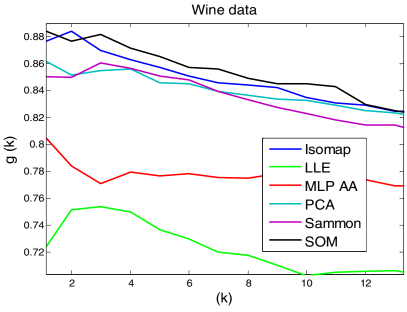


Fig. 3. Joint coincidence index versus neighborhood size – Wine dataset

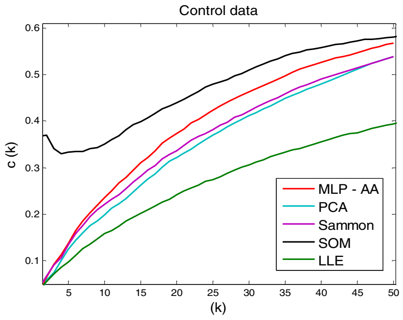


Fig. 4. Unordered coincidence index versus neighborhood size – Control dataset

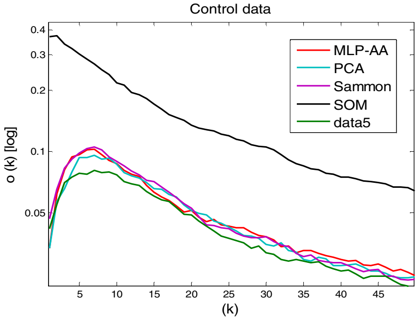


Fig. 5. Ordered coincidence index versus neighborhood size – Control dataset

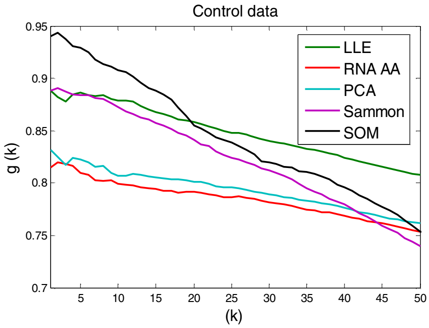


Fig. 6. Joint coincidence index versus neighborhood size – Control dataset.

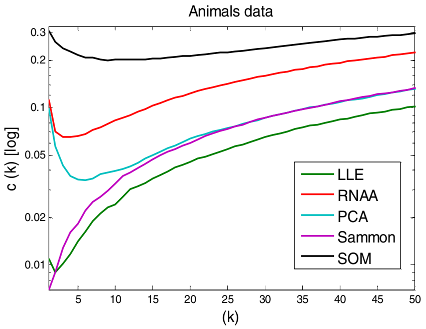


Fig. 7. Unordered coincidence index versus neighborhood size – Animals dataset

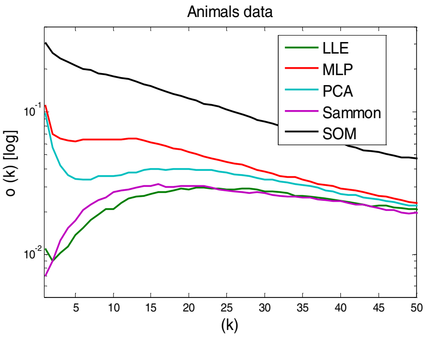


Fig. 8. Ordered coincidence index versus neighborhood size – Animals dataset

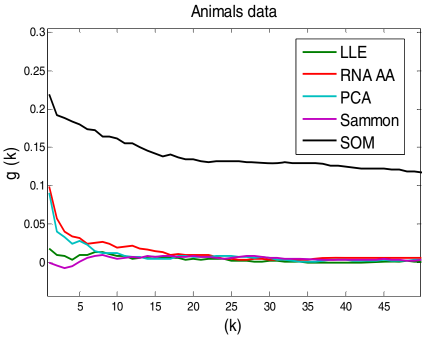


Fig. 9. Joint coincidence index versus neighborhood size – Animals dataset

The Animals database represents four classes of mammals through 72 attributes mostly of morphological measurements of animals. The Isomap projection was discarded out again due to large data losses. According to figures 7, 8 and 9, SOM performed better for both ordered, unordered and joint coincidence index. RNAA and PCA presented intermediate performance. LLE and Sammon presented lowest performance values in this test. It is noticed superior results of SOM in figure 9.

5 Conclusions

This paper presented a comparative study of six different methods of dimensionality reduction, including some classical and recent methods, applied to UCI databases. The methods were selected because of their representativeness and scientific relevance. Performances evaluation took into account approximate topology preservation measures by means of three proposed indexes based on neighborhood preservation. These indexes have proved useful comparison tools, although the absolute numerical values alone do not say much.

Considering the experiments within the three databases, the most suitable methods for data visualizing after unsupervised DR were the adaptive (connectionist) methods SOM and AANN. Isomap and LLE methods did not present good results in the experiments. PCA, although not designed to maintain topology, also presented reasonable results. Naturally, these conclusions should be placed in context, since the performance depends on the application of the methods that are used. It has to be considered that the databases, with three (Wine), six (Control) and four (Animals) classes, differs from other experiments based on a single manifold. The experiments in this work pointed out some methods that tend to present best results in reducing dimensionality toward identifying the structure of natural data, especially as in the presented cases, with presence of clusters.

Another important aspect is the difficulty of comparing elements that are not perfectly homogeneous, such as comparing SOM with other methods. Methods such as PCA perform a continuous mapping of the input data to the output space in contrast with discrete mapping performed by SOM. In the later case the output is always limited to the number of prototypes of the map. Even so, by establishing some appropriate conventions, it is possible to obtain a reasonable comparison. In this work, for example, it was agreed that the projected points on the same neuron would be considered neighbors and properly ordered. This is an optimistic approach that tends to be favorable to SOM, but is consistent with the visual perception of the projections. One alternative approach is to consider quantization error to establish an order within each unit.

Possible extensions of this work include considering other DR methods and data of different types, such as image, voice, time series, and so on. Other possibilities include the study of the influence of map size on the results of SOM and adjustments to the proposed indexes.

References

1. Yang, J., Ward, M.O., Rundensteiner, E.A., Huang, S.: Visual Hierarchical Dimension Reduction for Exploration of High Dimensional Datasets. In: Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2003)
2. Jolliffe, J.: Principal Component Analysis. Springer, Heidelberg (1986)
3. Cox, T.F., Cox, M.A.A.: Multidimensional Scaling, 2nd edn. Chapman and Hall/CRC (2001)
4. Kramer, M.A.: Nonlinear principal component analysis using Auto-associative Neural Networks. AIChE Journal 37(2) (1991)
5. Kohonen, T.: Self Organizing Maps, 3rd edn. Springer, Berlin (2003)
6. Tenenbaum, J.B., Silva, V., Langford, J.C.: A Global Geometric Framework for Nonlinear Dimensionality Reduction. Science, 290, 2319–2323 (2000)
7. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290, 2323–2326 (2000)
8. Van der Maaten, L.J.P.: An Introduction to Dimensionality Reduction Using Matlab, Report MICC 07-07, Universiteit Maastricht (2007)
9. Bezdek, J.C., Pal, N.R.: An Index of Topological Preservation for Feature Extraction. Pattern Recognition 28(3), 381–391 (1995)

10. UCI Machine Learning Repository, Irvine, CA. University of California, `http://www.ics.uci.edu/~mlearn/MLRepository.html`
11. Sammon, J.W.: A Nonlinear Mapping for Data Structure Analysis. IEEE Transactions on Computer C-18(5), 401–409 (1969)
12. Vesanto, J., et al.: Somtoolbox for Matlab, Report A 57, Helsinki University of Technology (2000)
13. Balachander, T., Kothari, R., Cualing, H.: An empirical comparison of dimensionality reduction techniques for pattern classification. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 589–594. Springer, Heidelberg (1997)
14. de Backer, S., Naud, A., Scheunders, P.: Nonlinear dimensionality reduction techniques for unsupervised feature extraction. Pattern Recognition 19(8), 711–720 (1998)
15. Yin, H.: Nonlinear Dimensionality Reduction and Data Visualization: A Review. International Journal of Automation and Computing 3(4), 294–303 (2007)

Multi Density DBSCAN

Wesam Ashour and Saad Sunoallah

Islamic University of Gaza, Gaza, Palestine
washour@iugaza.edu.ps, sad.ssa@gmail.com

Abstract. Clustering algorithms are attractive for the task of class identification in spatial databases. However, the application to large spatial databases rises the following requirements for clustering algorithms: minimal requirements of domain knowledge to determine the input parameters, discovery of clusters with arbitrary shape and good efficiency on large databases.DBSCAN clustering algorithm relying on a density-based notion of clusters which is designed to discover clusters of arbitrary shape. DBSCAN requires only one input parameter and supports the user in determining an appropriate value for it. DBSCAN cannot find clusters based on difference in densities. We extend the DBSCAN algorithm so that it can also detect clusters that differ in densities and without the need to input the value of Eps because our algorithm can find the appropriate value for each cluster individually by replacing Eps by Local cluster density.

Keywords: Clustering, Arbitrary Shape, DBSCAN, variable Densities.

1 Introduction

The density-based clustering approach is a methodology that is capable of finding arbitrarily shaped clusters, where clusters are defined as dense regions separated by low-density regions. A density-based algorithm needs only one scan of the original data set and can handle noise. The number of clusters is not required, since density-based clustering algorithms can automatically detect the clusters, along with the natural number of clusters [1].

Basic density based clustering techniques such as DBSCAN [2] and DENCLUE [3] treat clusters as regions of high densities separated by regions of no or low densities. So they are able to suitably handle clusters of different sizes and shapes besides effectively separating noise (outliers). But they fail to identify clusters with differing densities unless the clusters are separated by sparse regions.

We extend DBSCAN algorithm to discover clusters with different densities even if there is no low-density region separates them. Two adjacent spatial regions are separated into two clusters when the density difference violates a threshold. The proposed algorithm can automatically find Eps value for each cluster.

1.1 Density Based Clustering (DBSCAN)

DBSCAN is based on the concept of dense areas to form data clustering. The distribution of points in the cluster should be denser than that outside of the cluster. It defines a cluster as a maximal set of density-connected points.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 446–453, 2011.

The basic ideas of density-based clustering involve a number of new definition:

- The neighborhood within a radius Eps of a given object is called the ε -*neighborhood* of the object.
- If the ε-neighborhood of an object contains at least a minimum number, *MinPts*, of objects, then the object is called a *core object.*
- Given a set of objects, D, we say that an object p is *directly density-reachable* from object q if p is within the ε-neighborhood of q, and q is a core object.
- An object p is density-reachable from object q with respect to Eps and *MinPts* in a set of objects, D, if there is a chain of objects $p_1, \dots, p_n$, where $p_1 = q$ and $p_n = p$ such that p_{i+1} is directly density-reachable from p_i with respect to ε and *MinPts*, for $1 \le i \le n$, $p_i \in D$.
- An object p is density-connected to object q with respect to Eps and *MinPts* in a set of objects, D, if there is an object $O \in D$ such that both p and q are density-reachable from O with respect to ε and *MinPts*.

Density reachability is the transitive closure of direct density reachability, and this relationship is asymmetric. Only core objects are mutually density reachable [2].

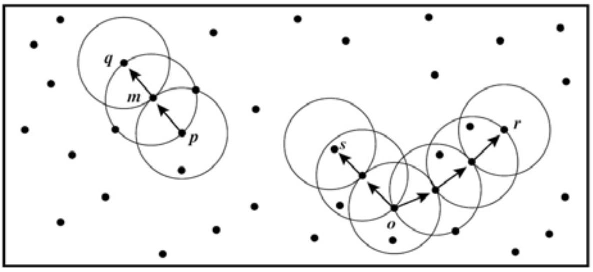


Fig. 1. Density reachability and density connectivity in density-based clustering (Source: Data Mining Concepts and Tecniques, Secnd edition, p419)

DBSCAN searches for clusters by checking the ε-neighborhood of each point in the database. If the ε-neighborhood of a point P contains more than MinPts, a new cluster with P as core point is created. DBSCAN then iteratively collects directly density-reachable points from these core points, which may involve the merge of a few density-reachable clusters.

2 Related Work

Early efforts like GAM [7] and CLARANS [8] detect clusters within huge data sets but demand high CPU effort. Later, BIRCH [9] was able to find clusters in the presence of noise. DBSCAN [2] detect clusters of arbitrary shapes. Recently, CHAMELEON [10] report clusters of different densities when data sets exhibit bridges and high discrepancy in density and noise [6].

Another algorithm [4] has introduced a simple idea to improve the results of DBSCAN algorithm by detecting clusters with large variance in density without requiring the separation between clusters.This algorithm adds a new point to the cluster depending on its core point density (the most dense point in the cluster) which will cause a problem in nearly similar density neighbor clusters with large internal density variance, which is solved in our algorithm by calculating the average density of the cluster in a slow learning process.

Another method [5], DDSC(A Density Differentiated Spatial Clustering Technique) detects clusters, which are having non-overlapped spatial regions with reasonable homogeneous density variations within them. If there is significant change in densities of adjacent regions then all are separated into different clusters.

2.1 DBSCAN Limitation: Sparse Clusters Adjacent to High-Density Clusters

For the data set in Fig. 2, DBSCAN will fall in a trap since it depends on a single Eps value so it will extract just one cluster from any adjacent clusters.

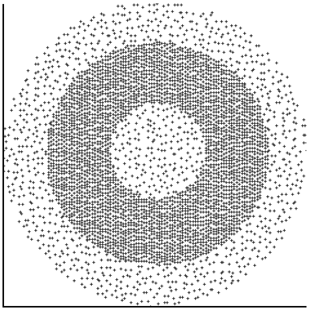
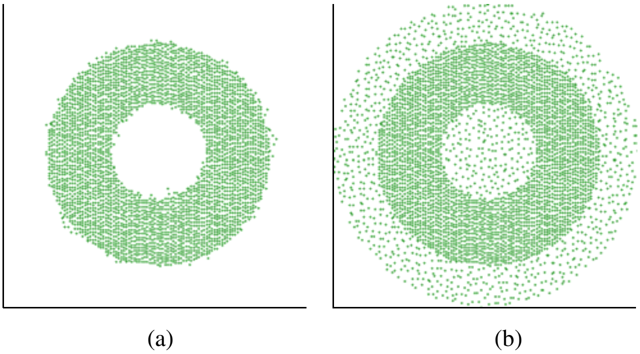


Fig. 2. Three adjacent clusters with different densities for 4700 point dataset



(a) (b)

Fig. 3. DBSCAN Result using MinPts=4 and: (a) small Eps value, (b) large Eps value

As we mentioned above DBSCAN depends on single Eps, for small Eps (high density) it will extract only the dense class (Fig.3.a) and supposes that other classes are noise, but for large enough Eps (Fig.3.b) all clusters realize the density conditions and they are connected so it assumes that they are a single cluster.

3 The Proposed Algorithm

As DBSCAN is sensitive to Eps we will use another variable: ***AVGDST*** which will represent the average distances between any point in the cluster and its kth-neighbors, that variable will be cluster dependent which means it will be varied from a cluster to another and it will be calculated automatically for each cluster.

In the algorithm we use another concept, since DBSCAN collects the nearest neighbors using the predefined Eps value, here we calculate $\boldsymbol{DST_p}$ value by taking the average distance of the nearest kth-neighbors of any point $\boldsymbol{p}$. The *AVGDST* of the cluster and the DST_p will decide if this point belong to that cluster or not.

The algorithm will follow these steps:

1. Calculate the Euclidean distance between each two points of the dataset.
2. For each point $\boldsymbol{p}$, find the average distance between it and its k^{th} nearest neighbors.

$$DST_p = \sum_{q \in N_p} \frac{dist(p,q)}{k}$$

Where N_p is the group of k^{th} nearest neighbors of p.

3. Insert all points in the queue list.
4. Starting from the most dense point in the queue which have the smallest *DST* value and do the following:
   (a) Assign the point $\boldsymbol{p}$ to new cluster (C_i).
   (b) Remove $\boldsymbol{p}$ from the queue list.
   (c) The initial average distances of the cluster will be:

$$\mathrm{AVGDST}_{\mathrm{Ci}} = \mathrm{DST}_\mathrm{p}$$

   (d) Call: *gather(C_i, p)*
5. *gather (C_i, p):* For each point $\boldsymbol{q}$ in the nearest k^{th}-neighbors of $\boldsymbol{p}$ do the following:
   (a) if ($\boldsymbol{q}$ in queue list) **and** ($DST_q \le var*AVGDST_{Ci}$) then:
      (i) Add q to C_i members
      (ii) Remove q from the queue list.
      (iii) Calculate $AVGDST_{Ci}$ from the equation:

$$AVGDST_i = \frac{AVGDST_i * (N_i - 1) + DST_q}{N_i}$$

      (iv) Call: *gather(C_i, q)*
      (v) End if

Where *var* is the distances variance in the class and its value must be ≥ 1, since we start with the most dense point and the new points added to the cluster will be the same *DST* or little larger, this variable will select the density variance in the same cluster, in other words the new added point to a cluster must has *DST* value the same as cluster *AVGDST* value or with some acceptable difference, this acceptable difference defined by the variable *var*. $(N_i - 1)$ represent the number of C_i members before adding the new point $\boldsymbol{q}$.

6. Eliminate very small clusters (noise).
7. Finish.

3.1 K-Value Problems

Since the algorithm depends on k value to calculate the *DST* for each point, selecting this value must avoid two problems depending on the dataset we use.

Bridges between two clusters

With huge dataset, some noise points may create bridges between clusters, Fig.4.a, these bridges lead our algorithm to merge bridged clusters if we use small k value (4 or 5 points), Fig. 4.b, but when increasing the value of k, Fig.4.c, the effect of the bridge will be eliminated since increasing the k value will increase the Eps value of the noise faster than the normal points and then the noise will be recognized. In the other side increasing the value of k leads to high processing time and complexity and will cause another problem, which is discussed in Fig.5.

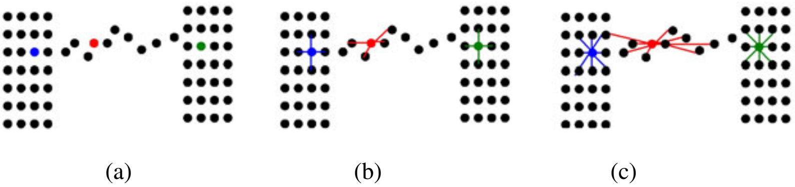


Fig. 4. (a) shows two clusters connected by a noise bridge, (b) using k=4, the noise point Eps value similar to the core points Eps, (c) increasing the value of k, will increase the Eps value of the noise

Cracks within a cluster

With huge non uniform datasets, cracks within clusters are a normal result, so fixing the k value to 4 or 5 will cause large clusters with cracks to be separated into two or more clusters, Fig.5.b, so as the dataset becomes huge the value of k must be increased to avoid the cracks problem, but a large value of k may cause to merge a well separated clusters. So the value of k must be large enough to eliminate noise bridges and connects cluster parts but not too large to merge well separated clusters.

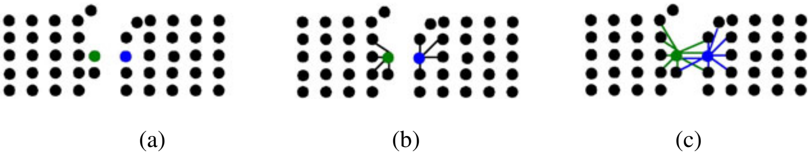
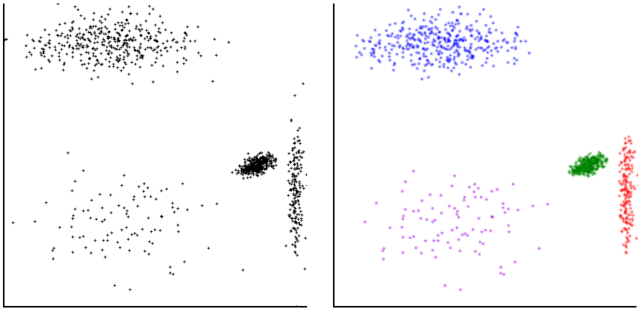


Fig. 5. (a) cracked cluster, (b) choosing k=4 will separate this cluster into two clusters, (c) increasing the value of k will merge cluster's parts

4 Experimental Results

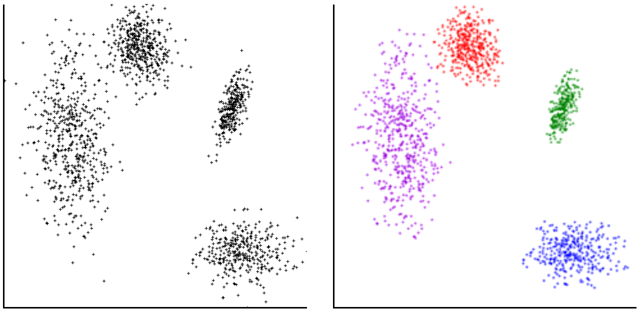
Here we evaluate the performance of the proposed algorithm. We implemented this algorithm in C#. We have used two dimensional datasets to test the algorithm.



(a) (b)

Fig. 6. (a) 1028 points dataset with four clusters, (b) the result using 2.5 variance, k=15

Fig.6.a shows four clusters of different size, shape, and density. The algorithm starts from the most dense point (lies in the green cluster) and collects the similar points around it until there is no more similar density around, then it starts from the most dense point in the remaining points (lies in the red cluster) and starts over again. Note that choosing a large enough k value protects from separating the red cluster by its south crack. The variance value 2.5 chosen by experiments outputs best results.



(a) (b)

Fig. 7. (a) 1572 points, with 4 clusters and (b) the result using 2.5 variance, k=15

In Fig.7.b we see a slightly different density clusters (red and violet) attached each other but since the algorithm has no unique Eps value so it separates them easily depending on the density of each cluster. This result may appear using DBSCAN but we need to select a critical value of Eps to ensure that neighbor clusters (Red and Violet) will not merge together.

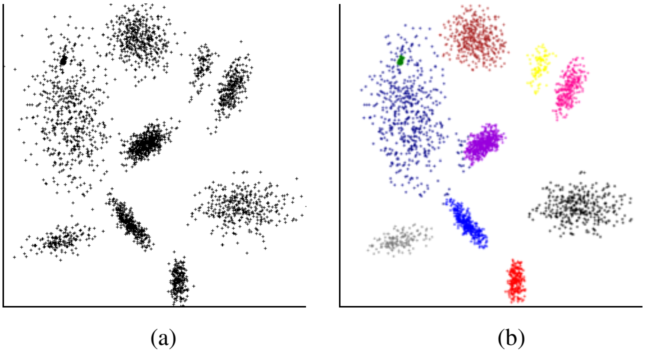


Fig. 8. (a) 2972 points, with 10 clusters and (b) the result using 2.5 variance, k=15

In Fig.8.a, we have an artificial data set with 10 clusters differ in shape, size and density. While there is no Eps value that can help the DBSCAN to find the clusters due to the variety in densities, our proposed algorithm finds all the clusters successfully as shown in Fig.8.b

In Fig.9 we have three clusters inside each other. DBSCAN will not be able to separate the clusters successfully due the variety in density. If DBSCAN finds the middle cluster, it will assume that the inner and outer clusters are noise. If we modify Eps in an attempt to allow DBSCAN to detect the inner and outer clusters successfully, then DBSCAN will merge all the clusters into one big cluster. Thus there is no value for Eps that allows DBSCAN to solve this data set. While the correct result will not appear in any Eps value using DBSCAN, the proposed algorithm can find the three clusters easily as shown in Fig. 9.b. These results illuminate the purpose of this algorithm, where the DBSCAN failed to find the correct clusters.

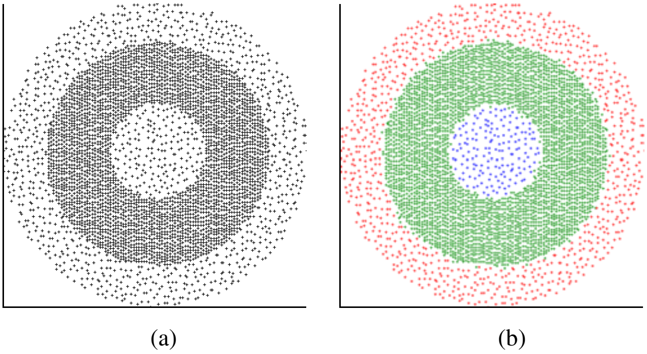


Fig. 9. (a) 4600 points with three clusters and (b) the result using 1.5 variance, k=15

5 Conclusion

In this paper, we have introduced an idea to improve DBSCAN algorithm to solve two main problems in it, the derivation of Eps value and the problem of connected clusters with different densities since DBSCAN depends on single Eps (threshold) value to all clusters.

Our algorithm starts from the most dense point as the core of a cluster, that cluster will grow by gathering similar density neighbor points, when it finishes that cluster it starts from the most remaining dense point as the core of another cluster an so on. The algorithm proves itself in many different densities, shapes, sizes, and clusters datasets with a very good result. In this paper, we show the problems of selecting small value of k in the large dataset, the noise bridge and the cracked clusters. We show that these problems can be solved by increasing the value of k.

References

1. Gan, G., Ma, C., Wu, J.: Data Clustering: Theory, Algorithms, and Applications, pp. 6–7. SIAM, Philadelphia (2007)
2. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density based algorithm for discovering clusters in large spatial data sets with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
3. Hinneburg, A., Keim, D.: An efficient approach to clustering in large multimedia data sets with noise. In: 4th International Conference on Knowledge Discovery and Data Mining, pp. 58–65 (1998)
4. Fahim, A.M., Saake, G., Salem, A.M., Torkey, F.A., Ramadan, M.A.: Improved DBSCAN for Spatial Databases with Noise and Different Densities. Georgian Electronic Scientific Journal: Computer Science and Telecommunications, 53–60 (2009)
5. Borah, B., Bhattacharyya, D.K.: DDSC: A Density Differentiated Spatial Clustering Technique. Journal of Computers 3(2), 72–79 (2008)
6. Estivill-Castro, V., Lee, I.: AUTOCLUST: Automatic Clustering via Boundary Extraction for Mining Massive Point-Dta Sets. In: Abrahart, J., Carlisle, B.H. (eds.) Proc. Of the 5th Int. Conf. on Geocomputation (2000)
7. Openshaw, S.: A Mark 1 Geographical Analysis Machine for the automated analysis of point data sets. International Journal of GIS 1(4), 335–358 (1987)
8. Ng, R.T., Han, J.: Efficient and Effective Clustering Method for Spatial Data Mining. In: Proceedings of the 20th International Conference on Very Large Data Bases (VLDB), pp. 144–155 (1994)
9. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 103–114 (1996)
10. Karypis, G., Han, E., Kumar, V.: CHAMELEON: A Hierarchical Clustering Algorithm Using Dynamic Modeling. IEEE Computer: Special Issue on Data Analysis and Mining 32(8), 68–75 (1999)

Genetic Algorithms to Simplify Prognosis of Endocarditis

Leticia Curiel[1], Bruno Baruque[1], Carlos Dueñas[2], Emilio Corchado[3], and Cristina Pérez-Tárrago[2]

[1] Department of Civil Engineering, University of Burgos, Burgos, Spain
[2] Complejo Hospitalario Asistencial Universitario de Burgos (SACYL), Servicio de Medicina Interna, Burgos, Spain
[3] Departamento de Informática y Automática, Universidad de Salamanca, Salamanca, Spain
lcuriel@ubu.es, bbaruque@ubu.es, cjdg@hgy.es, escorchado@usal.es

Abstract. This ongoing interdisciplinary research is based on the application of genetic algorithms to simplify the process of predicting the mortality of a critical illness called endocarditis. The goal is to determine the most relevant features (symptoms) of patients (samples) observed by doctors to predict the possible mortality once the patient is in treatment of bacterial endocarditis. This can help doctors to prognose the illness in early stages; by helping them to identify in advance possible solutions in order to aid the patient recover faster. The results obtained using a real data set, show that using only the features selected by employing a genetic algorithm from each patient's case can predict with a quite high accuracy the most probable evolution of the patient.

1 Introduction

Dimensionality reduction methods [1] involve processes such as feature construction, space dimensionality reduction, and sparse representations among others, which are achieved by using a wide array of techniques such as genetic algorithms [2], fuzzy systems [3] and others that investigate complex real problems in fields as medicine [4], ecology [5], engineering [6] and so on.

Infective endocarditis is a serious infection and its morbidity and mortality rate is still high, with a reported overall mortality rate ranging from 16 to 37.1% The risk of acquiring infective endocarditis is higher among patients with underlying heart diseases including valvular heart disease and congenital heart disease, among those with prosthetic cardiac valves, and among intravenous drug abusers. Substantial questions remain regarding the risk factors for infective endocarditis in bacterial infection. The changing profile of Infective Endocarditis requires continuous epidemiological updating associated infection.Usually, the illness is caused by a growth of bacteria on the edges of a defected heart or on the surface of an abnormal valve; after the bacteria enter the blood stream most commonly from dental procedures, tonsillectomy or adenoidectomy, certain types of surgery on the respiratory passageways, but also from procedures involving the gastrointestinal or urinary tract.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 454–462, 2011.

The endocarditis can be diagnosed by many procedures [7, 8] such as transthoracic echocardiography, transesophageal echocardiography, Duke criteria, magnetic resonance, tomography miltislide and by embolisms, etc.

Once the illness has been diagnosed, a rapid initiation of an adequate therapeutic regimen is important to prevent complications such as arrhythmias, brain abscess, brain or nervous system changes, congestive heart failure, glomerulonephritis, jaundice, severe heart damage, stroke,.., and death.

Patients with this condition usually need to be hospitalized to begin an aggressive treatment [7, 8] based on intravenously antibiotics. Initially, the treatment is empirical and the ideal situation is encountering the specific antibiotic for the organism causing the condition. This is determined by the blood culture and the sensitivity tests, which is not an immediate process.

For all these reasons, the correct treatment of the patient in the earliest stages as possible, is considered as an interesting objective. To help achieving this objective, this research proposes the use of genetic algorithms [9] techniques to select the most important features of this illness once the patient is in treatment, helping to predict the mortality risk.

The remaining of this paper is organised as follows. Section 2 introduces the decision genetic algorithms techniques used to realize feature selection. Section 3 describes classification models; in section 4 the dataset is explained; Section 5 shows the experiments and results obtained. Finally, in Section 6, the conclusions are set out and comments are made on future lines of work.

2 Feature Selection

The objective of this study is the identification of the most important patient's characteristics or symptoms in order to determine the future evaluation of their illness. As explained in previous sections, some of those are obtained from medical tests that can take a relatively long time, so it is important to know in advance which of them must be given higher priority. This is therefore, a clear case where the application of feature selection algorithms can be of much use.

In the case of this study, a Genetic Algorithm is employed as a mean for feature selection, enabling to guide the search among the most interesting combination of attributes (or dimensions) to obtain similar results of the ones obtained by using the whole set of attributes or characteristics for each patient.

2.1 Genetic Algorithms (GAs)

These kinds of algorithms are devised to solve search and optimization problems. They were originally proposed in [9] and are based in the evolution process of the biological species in nature. By imitating this behaviour, this family of algorithms is able to “evolve” a population of different solutions to the problem presented, until one of the generated solutions is fit enough to be considered as the final one [10].

The power of GAs comes from the fact that the basic technique is robust and can deal with a wide array of different problem statements. They are not guaranteed to find the global optimum solution for the given problem, but can achieve an “acceptably good” solution in a relatively low time [11].

In the case of the present work, this algorithm has been used as a way of performing a guided search among the different attributes that could be used to classify future evolution of the patients. This is usually known in literature as a wrapper method [12]. Each individual represents a different subset of the features chosen among the whole of them; while the fitness of each individual is the classification rate obtained by a regular machine learning classifier. In order to test the method in combination with a wider array of models, tests have been performed with three different classifiers: Support Vector Machines, ID3 Decision Trees and Naïve Bayes with Kernel Density Estimation.

3 Data Classification

3.1 Support Vector Machines

The Support Vector Machines (SVM) are supervised algorithms for the classification of multi-dimensional data samples or regression analysis. The most well-known version of the algorithm is the one proposed in [13].

It is based in the concept of hyper-planes used as decision boundaries. The algorithm is devised to find a high-dimensional plane that divides the data samples used as a training set into different classes, according to the labels provided. One of their main characteristics is that it will find the hyper-plane that accounts for the largest distance to the nearest training data points of any class, obtaining therefore the best possible generalization [14].

Mathematically expressed: if we consider the data samples $x_i \in \Re^d$ with their corresponding class labels $y_i \in \{\pm 1\}$; the SVM performs a mapping to a higher dimensional Hilbert space $\Phi : \Re^d \to H$. In that space (*H*) the decision rule is governed by a simple hyperplane that separates x_i into two different classes:

$$\overline{\psi} \cdot \overline{x}_i + b \geq k_0 - \xi_i, y_i = +1 \tag{1}$$

$$\overline{\psi} \cdot \overline{x}_i + b \leq k_1 + \xi_i, y_i = -1 \tag{2}$$

where ξ_i are positive slack variables introduced to handle the non-separable case and where k_0 and k_1 are typically defined to be +1 and -1 respectively.

In those cases, the Ψ is calculated by minimizing the objective function:

$$\frac{\overline{\psi} \cdot \overline{\psi}}{2} + C\left(\sum_{i=1}^{\ell} \xi_i\right)^p \tag{3}$$

subject to Eqs. (1) and (2), where C is a constant and p is usually chosen to be 2. A test vector (x_i) is then assigned a class label depending on whether $\overline{\psi} \cdot \overline{\Phi}(x) + b$ is greater or less than $(k_0 + k_1)/2$.

3.2 The Iterative Dichotomiser 3

The Iterative Dichotomiser 3 (ID3) [15] is a mathematical algorithm used to generate decision trees. This algorithm consists of constructing a tree from a random subset of the training set. The process must be repeated with the incorrect classifications values while the tree does not classify correctly the remaining cases of the training set.

To achieve this, the algorithm extracts the attribute that best separates the given cases into targeted classes. The algorithm uses the statistical property called "information gain" to choose which attribute is the best to separate training examples. This gain of set S on attribute A is defined as follows:

$$G(S,A) = E(S) - \sum_{v=1}^{t} \frac{|S_v|}{|S|} E(S_v) \qquad (4)$$

Where $\sum$ is each value v of all possible values of attribute A; S_v represents a subset of S which attribute A has value v; $|S_v|$ and $|S|$ are the number of elements in S_v and in S, respectively; and $E(S)$ is the information entropy of the subset S expressed by:

$$E(S) = -\sum p(I) \log_2 p(I) \qquad (5)$$

Where $p(I)$ is the collection of S belonging to class I.

3.3 Naïve Bayes with Kernel Density Estimation

The naïve Bayes classifier is also a very widespread supervised classifier, known for its simplicity and relatively good performance [16]. It is based in the probability theory, more precisely in Bayes theorem [17]. This method has the particularity that it will assume that the probability of each of the different attributes, to determine the final class of the sample, can be considered independent of the rest. That is, are conditionally independent given the class label. Although this does not always happen to be true, it is a good way to simplify the calculations. It performs its classification by calculating the *a priory* probability and the Likelihood of a sample belonging to a class by using a set of previously labeled training data.

Among many of the modifications that have been introduced to the basic algorithm, one of the most used is the inclusion of a kernel density estimator to calculate the true density of the continuous variables using kernels in the computation of the Likelihood of samples [18].

4 Data Description

The data set contains 50 cases of bacterial endocarditis extracted from the evolution of different patients that were admitted into the Complejo Hospitalario Asistencial Universitario de Burgos (Spain).

The following 14 input variables have been collected:

- Diagnostic Tool: Shows how the endocarditis has been diagnosed. The variables considered are: transesophageal echocardiography, transthoracic echocardiography, Duke criteria or autopsy.

- Clinical Time: It is the number of days passed from the appearance of first symptoms to endocarditis diagnosis.
- Patient's age: Contains the patient's age, where there are cases ranging from 15 to 89 years old.
- Patient's sex: Male or female.
- Complications: Resulting from infection during the treatment. It has been considered that the patient may suffer from heart failure, cardiogenic shock which is worse than heart failure, septic emboli and uncomplicated.
- Septic shock: It is life-threatening low blood pressure due to the introduction of bacteria into the blood stream.
- Catheter Sepsis: Indicates if sepsis is associated with intravascular catheters.
- Appropriate treatment: Once the illness is diagnosed, a rapid initiation of an adequate therapeutic regimen is required to prevent severe complications. The main treatment [7, 8] is through aggressive antibiotics. The problem is that the diagnosis of what kind of bacteria originated the infection is based on positive blood culture results with identical micro-organisms, which is not an immediate process. So, doctors in many cases have to begin the treatment before knowing the specific bacteria the patient is infected with. For this reason, sometimes, the treatment has to be changed once the blood culture results have been obtained. Then, this variable indicates whether the initial treatment is correct according to the bacteria.
- Change Time: The number of hours that the patient has been with an incorrect treatment.
- Previous valve: Indicates whether the affected patient's heart valve was working properly before being infected
- Valve type: Indicates the type of the infected heart valve. It is discriminated between native valve, prosthetic valve, pacemaker or prosthetic valve with pacemaker.
- Infected valve: Indicates the valve or valves affected. Organism: Bacteria that causes the infection. Contains more than 10 different types and its variants; such us enterococcus faecalis, enterococcus faecium, Haemophilus parainfluenzae, staphylococcus Lugdunens, staphylococcus parasanguis,…
- ICU: indicates whether the patient has been admitted to the intensive care unit.

The output to be predicted is the patient's condition 30 days after being admitted to the hospital. To simplify the problem, only the differentiation between "Alive" and "Dead" has been considered.

5 Experiments and Results

The aim of the experiments is to determine the most interesting set of features to determine the future evolution of the patient. In order to validate and test the use of GAs in this study, a classification comparison is proposed. A classification has been performed only with the variables identified as most relevant by the GA, after performing a wrapped search among all the features available on the dataset; then the results are compared with a classification performed using all the variables of the dataset.

For all experiments, the initial dataset is the one described in Section 4. It is therefore, composed of 50 different cases, each corresponding to a different patient; and 14 possible variables or features. As the dataset is relatively small, all experiments have been performed using the standard 10-fold cross-validation, in order to obtain statistically significant measures.

Table 1. Parameters used in the training of the models

SVM	**ID3**	**Naïve Bayes**
Kernel type: Dot.	Criterion: gain ratio	Estimation mode: Greedy.
Kernel cache: 200	Minimal size for split: 4	Bandwith: 0.1
Convergence epsilon: 0.001	Minimal leaf size: 2	Number of kernels: 10
Maximun iterations: 1000	Number of threads: 2	
Complexity constant: 0		
Genetic Algorithm parameters		
Selection mode: Roulette wheel	Prob. initalization: 0.5	
Population size: 5	Prob. mutation: -1	
Selection scheme: Tournament	Prob. crossover: 0.5	
Tournament size: 0.25	Crossover type: Uniform	

Table 2. Classification results with the different algorithms

		Feature Selection Classification			**Classification with all features**		
		SVM	ID3	Naïve Bayes	SVM	ID3	Naïve Bayes
Class recall	Alive	94.87%	97.44%	**97.14%**	94.87%	76.92%	94.44%
	Dead	33.33%	22.22%	**42.86%**	11.11%	44.44%	12.50%
Class precision	Alive	86.05%	84.44%	**89.47%**	82.22%	85.71%	82.93%
	Dead	60%	66.67%	**75.00%**	33.33%	30.70%	33.33%
Accuracy		83%	83.50%	**88.50%**	79.50%	71%	80.50%

For the sake of comparison, three different classification algorithms have been applied both to the complete dataset and inside the GA wrapper to select the subset of features for classification. The results of the three have been compared and shown in Table 2; while Table 3 shows the variables discriminated in each of the tests.Table 1 shows the parameters used for the training of each of the classifiers and for the Genetic Algorithm wrapper.

Looking at Table 2, it seems clear that the model that best classifies the future state of the patients, when is only trained with the features extracted using the Genetic Algorithm, is the Naïve Bayes. In the experiments performed, this combination accounts for the highest classification accuracy and best recall and precision in all but one class precision. It is able to prognosticate future cases close to 89% with the features selected where other models achieve values close to 83%.

Table 3. Features selected by the wrapped search depending on the model

	SVM	ID3	Naïve Bayes
Features selected	Age ICU Septic shock Complications Diagnostic Tool Previous valve Infected valve	Sex Complications Previous valve Organism Septic shock Catheter Sepsis Diagnostic Tool Infected valve Valve type	Sex AgeICU Septic shock Complications Appropriate treatment Change time Previous valve Valve type Infected valve Organism

Analysing Table 3, in this case according with the doctors' expertise and previous medical publications [19]; the most interesting set of features is the one included in the third column.

In the model adjusted for clinically important variables (age, sex, health care–associated acquisition of infection, diabetes, cancer, long-term immunosuppressive therapy, Organism *(S. aureus* infection), Previous valve (paravalvular abscess, cardiac surgery, Complications (stroke, heart failure, and new conduction abnormality)), variables independently associated with higher mortality among patients with native valve endocarditis were age 60 years or older, health care–associated acquisition of infection, diabetes, *S. aureus* infection, paravalvular abscess, stroke, heart failure, and new conduction abnormality. In agreement with previous studies, the results point to the fact that advanced age and endocarditis complications (stroke, heart failure, and septic embolism) were associated with greater mortality in patients with native valve endocarditis. Independent predictors of in-hospital mortality among patients with endocarditis in the present study included increasing age, systemic embolism, heart failure, prosthetic valvular endocardidtis and clinical delay. So, it can be concluded that the mortality rate may be increased by patient factors such as age and comorbid conditions, rather than by intrinsic qualities of the organism.

It is interesting to note that precisely the model combination that obtains best classification accuracy is the one proposed as having selected the most relevant features for the problem. This even outperforms the same classification algorithm when trained with all available features. Although this situation does not really add knowledge to what doctors already know, it proves that these models can successfully discard additional unimportant information and help to the prognosis of an illness using the patients' symptoms as they were obtained for the presented tests.

6 Conclusions and Future Research

The present study describes an ongoing multidisciplinary research in which an application of classical models by means of genetic algorithms to a medical problem has been presented. The features selected through the genetic algorithms presented are consistent with the medical literature found [19] and the models tested have been able to predict the mortality risk with a reasonable degree of accuracy using a relative small amount of samples.

This work proves that is possible to identify and discard the most uninteresting features for this analysis using automated learning algorithms, enabling doctors to concentrate in the remaining –most interesting– ones in the specific case of the endocarditis. In this application field, using this small amount of patients and reducing the features needed for each of them, seems as an advantageous feature; as such kind of real data are so costly to acquire.

Future work will be focused on the collection and storage of more specific attributes for each patient. Results seem to point to the fact that with more detailed data the medical condition of each patient alongside enough amount of different patients better results could be obtained. These results may include better prediction of mortality risk based on detailed data obtained from simple tests performed as close to the admission time of the patient as possible.

Another research line is the use of the information and experience gathered in these experiments for the development of a Case Base Reasoning system [20] to solve tasks related to the ones presented above. These would be able to handle the incorporation of new information with the treatment and monitoring the evolution of more patients.

Acknowledgments. We would like to extend our thanks to Complejo Hospitalario Asistencial Universitario de Burgos (SACYL). This research has been partially supported through projects TIN2010-21272-C02-01 from the Spanish Ministry of Science and Innovation and Grupo Antolin Ingenieria, S.A., within the framework of project MAGNO2008 - 1028.- CENIT.

References

[1] Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. IEEE Educational Activities Department 17(4), 491–502 (2005)

[2] Lorena, A.C., Ponce, A.C.: Evolutionary design of code-matrices for multiclass problems. In: Soft Computing for Knowledge Discovery and Data Mining, pp. 153–184. Springer, Heidelberg (2008)

[3] Berlanga, F.J., Rivera, A.J., Jesus, M.J., Herrera, F.: GP-COACH: Genetic Programming-based learning of Compact and Accurate fuzzy rule-based classification systems for High-dimensional problems. Information Science 180(8), 1183–1200 (2010)

[4] Chang, C.-D., Wang, C.-C., Jiang, B.C.: Using data mining techniques for multi-diseases prediction modeling of hypertension and hyperlipidemia by common risk factors. Expert Systems with Applications 38(5), 5507–5513 (2011)

[5] Baruque, B., Corchado, E., Mata, A., Corchado, J.M.: A forecasting solution to the oil spill problem based on a hybrid intelligent system. Information Sciences, Special Issue on Intelligent Distributed Information Systems 180(10), 2029–2043 (2010)

[6] Sedano, J., Curiel, L., Corchado, E., de la Cal, E., Villar, J.R.: A Soft Computing Based Method for Detecting Lifetime Building Thermal Insulation Failures. Integrated Computer-Aided Engineering 17(2), 103–115 (2010)

[7] Plicht, B., Erbel, R.: Diagnosis and treatment of infective endocarditis. Current ESC guidelines. HERZ 35(8), 542–548 (2010)

[8] Plicht, B., Janosi, R.A., Buck, T., Erbel, R.: Infective endocarditis as cardiovascular emergency. HERZ 51(8), 987–994 (2010)

[9] Holland, J.H.: Adaptation in natural and artificial systems. MIT Press, Cambridge (1992)

[10] Goldberg, D.E.: An Introduction to Genetic Algorithms. MIT Press, Cambridge (1996)

[11] Niknam, T., Fard, E.T., Pourjafarian, N., Rousta, A.: An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering. In: Engineering Applications of Artificial Intelligence, vol. 24, pp. 306–317. Pergamon-Elsevier Science Ltd. (2011)

[12] Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97, 273–324 (1997)

[13] Vapnik, V.: Statistical Learning Theory. Springer, New York (1998)

[14] Burges, C.J.C.: A tutorial on Support Vector Machines for pattern recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)

[15] Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)

[16] Rish, I.: An empirical study of the naive Bayes classifier. In: Proceedings of IJCAI-2001 Workshop on Empirical Methods in AI In International Joint Conference on Artificial Intelligence, pp. 41–46 (2001)

[17] Bayes, T.: An Essay towards solving a Problem in the Doctrine of Chances. Philosophical Transactions of the Royal Society of London 53(2), 370–418 (1763)

[18] Larrañaga, P., Inza, I., Martinez, A.P.: Bayesian classifiers based on kernel density estimation. International Journal of Approximate Reasoning 50(2), 341–362 (2009)

[19] Benito, N., Miro, J.M., Lazzari, E., Cabell, C.H., Rio, A., Altclas, J., Commerford, P., Delahaye, F., Dragulescu, S., Giamarellou, H., Habib, G., Kamarulzaman, A., Sampath, A., Nacinovich, F.M., Suter, F., Tribouilloy, C., Venugopal, K., Moreno, A., Fowler, V.G.: The ICE-PCS (International Collaboration on Endocarditis Prospective Cohort Study) Investigators. Health Care Associated Native Valve Endocarditis: Importance of Non-nosocomial Acquisition. Annals of Internal Medicine 150, 586–594 (2009)

[20] Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. Artificial Intelligence Communications-AICom 7(1), 39–59 (1994)

Analyzing Key Factors of Human Resources Management

Lourdes Sáiz[1], Arturo Pérez[2], Álvaro Herrero[1], and Emilio Corchado[3]

[1] Department of Civil Engineering, University of Burgos
C/ Francisco de Vitoria s/n, 09006 Burgos, Spain
[2] Investigador Programa Torres Quevedo, TTT Diseño, Comunicación y Contenidos, S.L.
Reyes Católicos, 41, 1, 09005 Burgos, Spain
[3] Departamento de Informática y Automática, Universidad de Salamanca
Plaza de la Merced, s/n, 37008 Salamanca, Spain
{lsaiz,ahcosio}@ubu.es, arrturo@hotmail.com, escorchado@usal.es

Abstract. This study presents the application of an unsupervised neural projection model for the analysis of Human Resources (HR) from a Knowledge Management (KM) standpoint. This work examines the critical role that the acquisition and retention of specialized employees play in Hi-tech companies, particularly following the configuration approach of Strategic HR Management. From the projections obtained through the connectionist models, experts in the field may extract conclusions related to some key factors of the HR Management. One of the main goals is to deploy improvement and efficiency actions in the implantation and execution of the HR practices in firms. The proposal is validated by means of an empirical study on a real case study related to the Spanish Hi-tech sector.

Keywords: Unsupervised Neural Networks, Knowledge Management, Human Resources Management, Acquisition & Retention.

1 Introduction

Knowledge Management (KM) enables organizations to capture, share, and apply the collective experience and the know-how (knowledge) of their staff. For KM to be successfully applied in organizations, it is necessary to develop and implement knowledge infrastructures [1]. These knowledge infrastructures consist of three main dimensions: people, organizational and technological systems.

In recent years, the deployment of information technology has become a crucial tool for enterprises to achieve a competitive advantage and organizational innovation [2]. In keeping with this idea, Artificial Intelligence (AI) [3] can be applied in KM systems in order to speed up processes, classify unstructured data formats that KM is unable to organize, visualize the intrinsic structure of data sets, and select employee-related knowledge from large amounts of data, among other processes.

In keeping with this idea, the present study deals with Human Resource (HR) Management in Hi-tech companies that allows improving the interpretation and processing of the information related to HR practices in firms. In a more precise way,

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 463–473, 2011.

this work is intended for detecting situations of effective and ineffective management and diagnosing the most advisable actions for every case.

To identify the position of a company according to its HR Management, neural projection models are applied in this research. The main goal is to identify the status of a company to perform subsequent corrective actions under the frame of a Hybrid Artificial Intelligence System, as proposed in [4]. This is an ongoing research that started with a wider analysis of HR features [5] and is now focused on the acquisition and retention settings related with R+D employees of high-tech firms. These are key issues as there is a lack of this kind of employees, according to the data provided by the Spanish National Institute of Statistics [6]. Decisions in this field are strongly related with rivalry. In this sense, KM provides us with valuable information to determine the most effective politics on HR acquisition and retention. This allows proposing to each firm the decision to be taken in order to improve an unsatisfactory or inefficient HR situation.

The study is structured in the following way. Section 2 introduces the field of HR and KM, while section 3 describes the unsupervised neural projection model applied in this research. Section 4 covers the application to a real-world problem and the experimental study that has been carried out. Finally, Section 5 presents the conclusions and some proposals for future work.

2 Knowledge Management and Human Resources

In the present economic context, similar changes have occurred in almost every economic sector, cutting down the strategic importance of tangible resources. The more knowledge demanding a sector is (as the case of high technology), the more important knowledge-based resources are as value generators. Organizations investing in innovation obtain more benefits than those who do not [7].

Innovation is an important result for firms, mainly in technological industries [8]. Furthermore, in global and dynamic environments, the successful firms do need an explicit strategy involving innovation and KM for fast adaptation [9]. The main reason for this is that the introduction of new products and services depends on the capability to generate, combine and exchange the new knowledge [10]. Thus, the source of innovation is the integration of new knowledge with the knowledge previously stored by individuals in a specialized form [11].

Organizations may achieve good performance results if they are able to both take advantage of the knowledge held by a group of people and to effectively organize them [12]. The processes of knowledge creation, transformation, integration and influence are related with the internal collaboration of employees. HR Management promotes and eases knowledge gathering, and at the same time it leads to implement the previously mentioned processes. It is especially critical when HR are strongly linked to technological resources.

One of the ways to achieve the required level of HR is by employing those people who have previous experience and specialized knowledge. In keeping with this idea, HR planning, extensive recruiting, and selection practices are key factors to achieve a high level of HR. However, there is a lack of such employees and, as a result, there is a serious rivalry between firms. Thus, the firms that will get the employees with high knowledge level would be those applying exhaustive recruiting policies such as multiple selection sources, a wide group of candidates, high salaries, among others.

Employing specialized staff will only partially solve the associated problem because, at the same time, these HR may move away to a different firm. As a firm generates value through their employees, success partially relies on its ability to keep them [12]. As knowledge and knowledge creation take place in the individuals, it is a menace that abilities and specific knowledge may disappear, reducing the firm capability to generate new knowledge.

It is important for knowledge acquisition that employees know who do they have to contact in order to find the required knowledge [13]. Hence, personnel stability is essential to perfect development of knowledge resources [14]. The stay of such employees may be guaranteed through HR management systems including compensations and rewards higher than average, comfortable working places, pleased and committed employees, social training, employee-organization connection, welcoming politics, labour safety, autonomy, participating culture, etc.

Taking into account the above mentioned ideas, a wide research on analysis factor was carried out [5] ranging from starting advanced HR practices to explaining firm results. To do so, it focused on intermediate related indicators such as employee characteristics, organizational capabilities and some other internal features. The studied factors were: five settings for HR practices (acquisition, development, compromise, retention and flexibility), five features of employees (human, social and organizational capital, motivation and turn over), four organization capabilities (knowledge creation and application, organizational flexibility and information technologies) and some other internal features (strategic vision, HR emphasis, heterogeneity, and task-associated technology). From the above mentioned, the present study has only taken into account those decisions related to the acquisition of specialized personnel, employees retention and the configuration of these two ones (Strategic HR Management [15]). The remaining factors will be covered in future work. Some HR practices are coherent with the strategic objectives and greatly contribute to the achievement of firm results. Some of the features related with these practices have been considered in the present study to visually identify the situation of hi-tech firms according to their HR situation.

3 Neural Projection Model

Projection models perform changes to the spatial coordinates of high-dimensional data in order to project them onto a lower dimensional space. The main goal is to identify the patterns that exist across dimensional boundaries by identifying “interesting” directions in terms of any specific index or projection. Such indexes or projections are, for example, based on the identification of directions that account for the largest variance of a data set -as is the case of Principal Component Analysis (PCA) [16], [17]- or the identification of higher order statistics such as the skew or kurtosis index -as is the case of Exploratory Projection Pursuit (EPP) [18]. Having identified the most interesting projections, the data is then projected onto a lower dimensional subspace plotted in two or three dimensions, which makes it possible to examine its structure with the naked eye.

The combination of projection techniques together with the use of scatter plot matrices is a very useful visualization tool to investigate the intrinsic structure of multidimensional data sets, allowing experts to study the relations between different components, factors or projections, depending on the applied technique.

The solution proposed in this research applies an unsupervised neural model called Cooperative Maximum Likelihood Hebbian Learning (CMLHL) [19]. It is based on Maximum Likelihood Hebbian Learning (MLHL) [19], and introduces the application of lateral connections [20], [21] derived from the Rectified Gaussian Distribution [22]. This connectionist model has been chosen because it reduces the data dimensionality while preserving the topology in the original data set. Considering an N-dimensional input vector (x), and an M-dimensional output vector (y), with W_{ij} being the weight (linking input j to output i), then CMLHL can be expressed as:

1. Feed-forward step:

$$y_i = \sum_{j=1}^{\mathbf{N}} W_{ij} x_j, \forall i \ . \tag{1}$$

2. Lateral activation passing:

$$y_i(t+1) = [y_i(t) + \tau(b - Ay)]^+ \ . \tag{2}$$

3. Feedback step:

$$e_j = x_j - \sum_{i=1}^{M} W_{ij} y_i, \forall j \ . \tag{3}$$

4. Weight change:

$$\Delta W_{ij} = \eta . y_i . sign(e_j) | e_j |^{p-1} \ . \tag{4}$$

Where: η is the learning rate, τ is the "strength" of the lateral connections, b the bias parameter, p a parameter related to the energy function [19], [21] and A a symmetric matrix used to modify the response to the data [21]. The effect of this matrix is based on the relation between the distances separating the output neurons.

4 Experimental Study

To empirically validate the importance of the acquisition and retention factors, this study has covered 126 Spanish organizations related with high-technology. 267 R+D employees from these firms were surveyed in order to analyze the HR strategies and subsequently improve the status of the analyzed firms.

The average profile of these hi-tech firms is that of an organization with 266 employees, manufacturing products and services (111 out of the 126 studied organizations). 47% of the analyzed firms claim they innovate in both products and services, running 124 R+D annual programs. 44% of the analyzed firms are members of a corporate group, 16% of them being international.

As only the HR acquisition and retention factors are considered in the present study, features related with these two factors were analyzed from the surveyed data. These features are described in the following table.

Table 1. Analyzed features for both acquisition and retention factors

Acquisition factor	Retention factor
1.- There is a plan to find the required HR abilities.	1.- Candidates are selected according to their fitting with the firm.
2.- A wide group of candidates is considered for vacancies.	2.- Employees match with the organization culture.
3.- Initial salaries are higher than competitors to attract candidates.	3.- New employees are supported.
4.- A high amount of money is spent to contract the right person.	4.- Social and outdoor activities are sponsored by the firm for employees to know each other.
5.- Trainers of new employees are carefully chosen and prepared.	5.- Higher salaries than competitors are offered to keep employees.

Five features have been used to define each factor. The values for all these features are discrete ones and range from 1 (strongly low) to 5 (strongly high). As a result, five features from each firm (126) have been gathered in both dataset (acquisition and retention).

4.1 Results

This section comprises an analysis of the best projections obtained in the above-described experimental study by applying the CMLHL model to the data related with the two HR factors. The visualized groups in each one of the projections has been labelled (1.1, 1.2 …) in Fig. 1 and Fig. 2 for easy referencing.

Acquisition Factor

Firstly, CMLHL was applied to the acquisition features, generating the projection in Fig. 1. An in-deep analysis of this projection allows us to define the common characteristics for each one of the clearly identified groups (labelled in Fig. 1) according to the dataset features.

Considering the first feature for the acquisition factor ("There is a plan to find the required HR abilities"), the firms are decreasingly ordered from left to right in Fig. 1. Thus, firms in the left third of the projection (groups 1.1, 1.2, 2.1, 2.2, 2.3, and 2.4) are those with the highest values for this feature, which means that these firms have designed and applied the required abilities for the economic activity. Those firms in the middle of the projection identified this ability plan as incomplete and it is even worse in the case of companies in the right third of Fig. 1 (groups 4.1, 4.2, 4.3, 4.4, 4.5, 5.1, 5.2, and 5.3). According to this result, it has been checked that most of the firms have not completed the staff ability plan that is required and critical in order to fulfil the criteria for the acquisition factor.

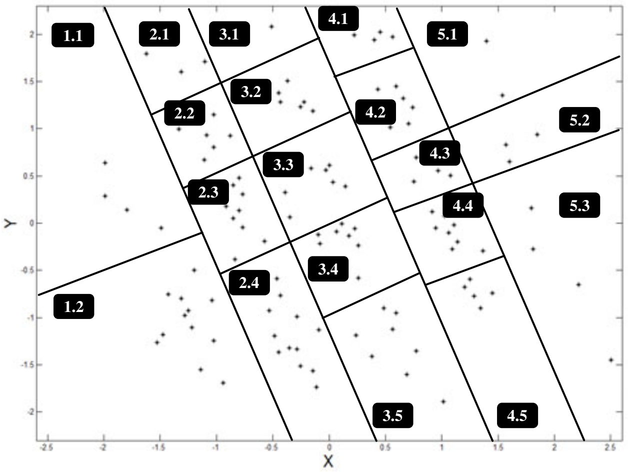


Fig. 1. CMLHL projection of the Acquisition Factor dataset

Regarding the second feature for the acquisition factor, the CMLHL projection provides a clearer clustering of firms. The firms are horizontally ordered, taking the highest values for this feature in the left side of the projection and the lowest values in the right side of the projection. Most of the companies are in an intermediate location, twelve of the companies take the highest value for this feature (Groups 1.1 and 1.2 in Fig. 1) while nine of them take the lowest value (Groups 5.1, 5.2 and 5.3 in Fig. 1). The more precisely the ability requirements are defined, more candidates will apply for the vacancies.

The third feature of the acquisition factor reveals a vertical ordering of the data in Fig. 1. The firms at the bottom of the CMLHL projection are those taking the highest values for this feature, while the ones at the top of the projection take the lowest value. After the visual analysis it can be concluded that many of the firms take high values regarding this feature, so they have the best employees from the market.

The fourth feature shows a vertical ordering of the data in Fig. 1, as in the case of the previous one. It is coherent with the previous feature as those firms that care about the knowledge of beginner trainers do also pay higher initial salaries than competitors.

The fifth feature complements the previous ones in the study of acquisition factor. The grouping obtained through CMLHL (Fig. 1) is not as precise as previously indicated. However, there is a kind of ordering, as those firms in a better situation are in the left side of the projection and those in the worse situation are in the right side.

From a general perspective, the projection organizes the firms in nineteen different groups (from 1.1 to 5.3). This global order leads to gather in group 1.2 those firms with the best policies regarding HR. On the other hand worst HR policies cause firms to be located in groups 5.1 and 5.2. Firms in an intermediate situation are gathered in groups 2.2, 2.3, 3.2, 3.3, 3.4, 4.2, 4.3 and 4.4.

Retention Factor

Fig. 2 shows the projection of firms by means of CMLHL and according to the retention factor.

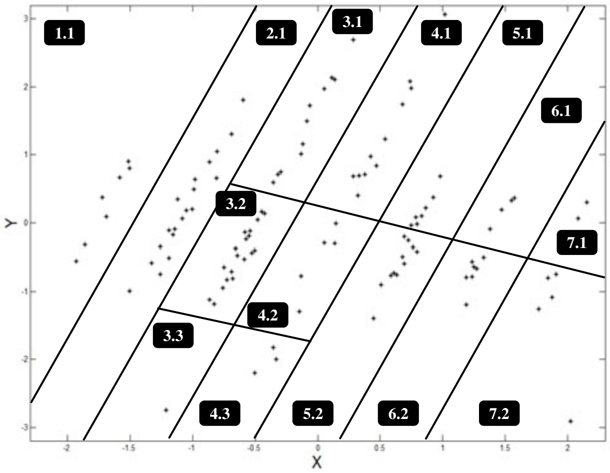


Fig. 2. CMLHL projection of the Retention Factor dataset

The first feature to be considered for this factor (Candidates are selected according to their fitting with the firm) is very important, and causes a vertical order of the data. Those firms at the top of Fig. 2 are the ones with the highest values for this feature, while the ones at the bottom are those whose fitting to new employees is almost inexistent. Two of the companies do not take this issue into account at all and are depicted at the very bottom of the projection.

Regarding the second feature, the projection shows a horizontal distribution of the data. Most of the companies are in an intermediate situation, as only six of them take the highest value (in the right side of Fig. 2) and seven of them take the lowest value (in the left side of Fig. 2). This feature is related with the previous one as there is a coincidence between the global fitting and the firm culture.

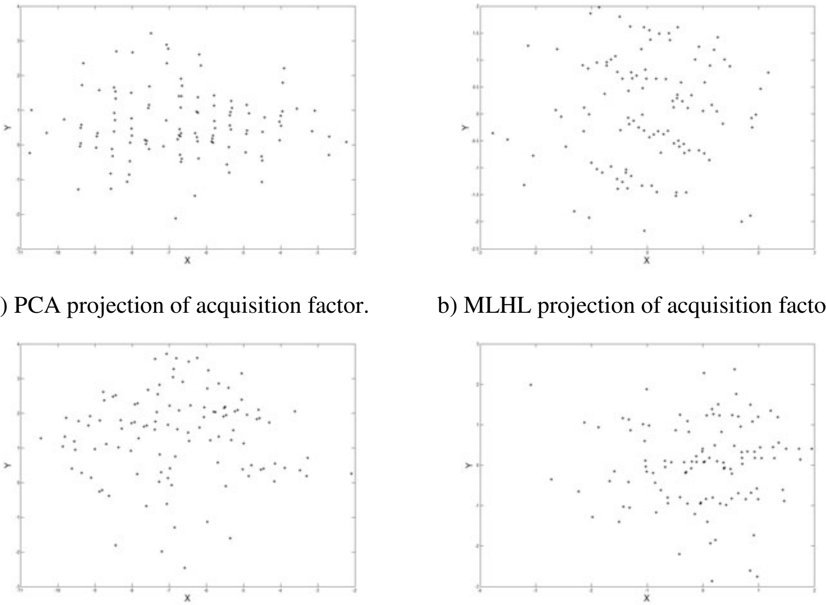
The third one of the retention features reveals a precise tendency in the projection. Firms at the bottom of Fig. 2 are the ones with highest values for this feature. The lowest values place firms at the top of the projection. There is not a big amount of firms taking the maximum value for this feature. On the contrary, most of the firms take medium/minimum values. This can be identified as a deficiency in the important task of new employee integration. Considering the previous features, those firms that get the highest value in the employee selection according to the global fitting with the company do not care about the successful integration of new employees.

According to the fourth feature, the firms are ordered in a similar way as for the second one (highest scores in the right side and the lowest ones in the left side). This is coherent as the sponsoring of social activities for employees is part of the culture values of the company and leads to a high level of commitment that may be extended to other cultural components. The fifth variable shows a vertical tendency in the data ordering as in the case of the first feature related with the acquisition.

From a general perspective, the whole dataset is split in fourteen different groups according to the retention factor. The best practices are performed by the firms in groups 6.1 and 7.1 (Fig. 2). On the other hand, firms in group 1.1 are those in the worst situation regarding the analyzed factor. Some other firms are in an intermediate situation and are located in groups 3.1, 3.2, 4.1, 4.2, 5.1 and 5.2 (Fig. 2).

4.2 Comparative Study

To compare the CMLHL projections with those obtained from some other unsupervised techniques, Principal Component Analysis (PCA) [17], MLHL and Self-Organizing Map (SOM) [23] have been also applied to the HR dataset. The obtained projections are shown in Fig. 3. The PCA neural version [24], MLHL and CMLHL are compared below as the three of them are projection models based on unsupervised learning, aimed to provide a visual analysis of the internal structure of a data set.



a) PCA projection of acquisition factor.

b) MLHL projection of acquisition factor.

c) PCA projection of retention factor.

d) MLHL projection of retention factor.

Fig. 3. PCA and MLHL projections of HR data (acquisition and retention datasets)

The sum of the data variance accumulated by the two first principal components is 67.8% for the acquisition factor (Fig. 3.a) and 66.9% for the retention factor (Fig. 3.c). As can be seen in Fig. 3 none of them reveals the inner structure of the data for any of the analyzed factors. On the other hand, the MLHL projections of both acquisition (Fig. 3.b) and retention (Fig. 3.d) factors allow a deeper analysis of the data.

Regarding the acquisition factor, MLHL provides a less precise visualization of the data structure than CMLHL. However, a certain organization of the data can be identified. Firms in the left side of the projection (Fig. 3.b) take high values concerning the second, third and fifth features. Firms in the right side of the projection are in the opposite situation regarding the mentioned features. It is the other way round considering the first and fourth features. Firms in the centre of the projection (Fig. 3.b) take intermediate values, whose situation improve if they are at the bottom of the projection.

An in-deep analysis of the retention factor and the MLHL projection (Fig. 3.d), shows that firms in the left side of the projection take the highest values regarding the second, third and fourth features. Firms taking the highest values for the first and fifth features are located in the right side of the projection.

On the other hand, SOM has also been applied to the above described dataset. For the sake of brevity, graphical results of the SOM mapping have not been included in this study. It can be said that the SOM does not provides with a clear clustering of the data, as neurons do not collectively respond to a certain kind of input data.

While previous research [25] [26] [27] outlines the effectiveness of HRM practices (such as selective hiring, performance appraisal, self-managed team, extensive training and some others) to increase the firm's levels of knowledge acquisition, sharing and application, some other studies [5] [28] [29] reach evidence that KM capabilities play a mediating role between HRM practices and overall firm performance.

5 Conclusions and Future Work

The main target of the present study is to provide a powerful and robust (to be applied to disparate data) intelligent tool for researching on KM and HR, from purely theoretical formulations to applications allowing a diagnosis based on the captured data.

This objective has been fulfilled as CMLHL has been able to visualize the data in an ordered way, leading to a fruitful analysis of the firms' situation. The clear-cut grouping in the projections allows the definition of firms according to the acquisition and retention settings. Subsequent corrective actions can be proposed according to the firm strategy.

In conclusion, the application of unsupervised neural techniques has contributed to successfully detect the HR practices in firms and how to evolve and progress toward a more efficient deployment and execution of them. This will greatly contribute to improve the efficiency and competitiveness of firms. Thus, CMLHL provides with a strategic viewpoint that allows us to identify the specific key characteristics for companies to improve their knowledge management status.

Future work will focus on extending this study with some other factors that are important in the HR and KM fields (described in Section 2), and in the application of other unsupervised models such as the ViSOM [30].

Acknowledgments. This research has been partially supported through the Spanish Ministry of Science and Innovation (MICINN) under projects DPI2010-16920, Consolider CSD2010-00034 and CIT-020000-2009-12 (funded by the European Regional Development Fund); project of the Spanish Ministry of Science and Innovation TIN2010-21272-C02-01 (funded by the European Regional Development Fund). The authors would also like to thank the vehicle interior manufacturer, Grupo Antolin Ingenieria S.A., within the framework of the MAGNO2008 - 1028.- CENIT Project also funded by the MICINN.

References

1. Sivan, Y.Y.: Nine Keys to a Knowledge Infrastructure: A Proposed Analytic Framework for Organizational Knowledge Management. In: WebNet 2000 - World Conference on the WWW and Internet, pp. 495–500. AACE (2000)
2. Shu-Mei, T.: The Effects of Information Technology on Knowledge Management Systems. Expert Systems with Applications: An International Journal 35(1-2), 150–160 (2008)
3. Russell, S.J., Norvig, P.: Artificial Intelligence: a Modern Approach. Prentice Hall, Englewood Cliffs (1995)
4. Herrero, Á., Corchado, E., Sáiz, L., Abraham, A.: DIPKIP: A Connectionist Knowledge Management System to Identify Knowledge Deficits in Practical Cases. Computational Intelligence 26(1), 26–56 (2010)
5. Sáiz, L., Pérez, A.: Formación de las Capacidades de Creación de Conocimiento y Flexibilidad Organizativa en Empresas de Alta Tecnología. In: 4th International Conference on Industrial Engineering and Industrial Management (2010)
6. Spanish National Institute of Statistics, `http://www.ine.es/`
7. Soo, C.W., Devinney, T.M., Midgley, D.F.: External Knowledge Acquisition, Creativity and Learning in Organisational Problem Solving. International Journal of Technology Management 38(1), 137–159 (2007)
8. Akgün, A.E., Keskin, H., Byrne, J.C., Aren, S.: Emotional and Learning Capability and their Impact on Product Innovativeness and Firm Performance. Technovation 27(9), 501–513 (2007)
9. Maranto-Vargas, D., Rangel, R.G.T.: Development of Internal Resources and Capabilities as Sources of Differentiation of SME under Increased Global Competition: A Field Study in Mexico. Technol. Forecast. Soc. Chang. 74(1), 90–99 (2007)
10. Smith, K.G., Collins, C.J., Clark, K.D.: Existing Knowledge, Knowledge Creation Capability, and the Rate of New Product Introduction in High-Technology Firms. Academy of Management Journal 48, 346–357 (2005)
11. Grant, R.M.: Toward a Knowledge-based Theory of the Firm. Strategic Management Journal 17(10), 109–122 (1996)
12. Kamoche, K.N.: Managing People in Turbulent Economic Times: A Knowledge-creation and Appropriation Perspective. Asia Pacific Journal of Human Resources 44(1), 25 (2006)
13. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company. Oxford Press, Oxford (1995)
14. Collins, C.J.: Strategic Human Resource Management and Knowledge-creation Capability: Examining the Black Box between HR and Firm Performance. PhD Thesis. Robert H. Smith School of Business. University of Maryland (2000)
15. Dolan, S.L., Mach, M., Olivera, V.S.: HR Contribution to a Firm's Success Examined from a Configurational Perspective: An Exploratory Study Based on the Spanish CRANET Data. Management Review 16(2), 272 (2005)
16. Hotelling, H.: Analysis of a Complex of Statistical Variables into Principal Components. Journal of Education Psychology 24, 417–444 (1933)
17. Pearson, K.: On Lines and Planes of Closest Fit to Systems of Points in Space. Philosophical Magazine 2(6), 559–572 (1901)
18. Friedman, J.H., Tukey, J.W.: A Projection Pursuit Algorithm for Exploratory Data-Analysis. IEEE Transactions on Computers 23(9), 881–890 (1974)

19. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Mining and Knowledge Discovery 8(3), 203–225 (2004)
20. Corchado, E., Han, Y., Fyfe, C.: Structuring Global Responses of Local Filters Using Lateral Connections. Journal of Experimental & Theoretical Artificial Intelligence 15(4), 473–487 (2003)
21. Corchado, E., Fyfe, C.: Connectionist Techniques for the Identification and Suppression of Interfering Underlying Factors. International Journal of Pattern Recognition and Artificial Intelligence 17(8), 1447–1466 (2003)
22. Seung, H.S., Socci, N.D., Lee, D.: The Rectified Gaussian Distribution. In: Advances in Neural Information Processing Systems, vol. 10, pp. 350–356 (1998)
23. Kohonen, T.: The Self-Organizing Map. Proceedings of the IEEE 78(9), 1464–1480 (1990)
24. Fyfe, C.: A Neural Network for PCA and Beyond. Neural Processing Letters 6(1-2), 33–41 (1997)
25. Har, W.C., In, T.B., Phaik, L.S., Hsien, L.V.: The Impact of HRM Practices on KM: A Conceptual Model. Journal of Applied Sciences Research 6(12), 6281–6291 (2010)
26. Zheng, W., Yang, B., McLean, G.N.: Linking Organizational Culture, Structure, Strategy, and Organizational Effectiveness: Mediating Role of Knowledge Management. Journal of Business Research 63(7), 763–771 (2010)
27. Liao, H., Toya, K., Lepak, D.P., Hong, Y.: Do They See Eye to Eye? Management and Employee Perspectives of High-Performance Work Systems and Influence Processes on Service Quality. Journal of Applied Psychology 94(2), 371–391 (2009)
28. Chen, C.-J., Huang, J.-W.: Strategic Human Resource Practices and Innovation Performance - The Mediating Role of Knowledge Management Capacity. Journal of Business Research 62(1), 104–114 (2009)
29. Theriou, G.N., Chatzoglou, P.D.: Enhancing Performance through Best HRM Practices, Organizational Learning and Knowledge Management: A Conceptual Framework. European Business Review 20(3), 185–207 (2008)
30. Yin, H.: ViSOM - a Novel Method for Multivariate Data Projection and Structure Visualization. IEEE Transactions on Neural Networks 13(1), 237–243 (2002)

A Principled Approach to the Analysis of Process Mining Algorithms

Phil Weber, Behzad Bordbar, and Peter Tiňo

School of Computer Science, University of Birmingham, B15 2TT, UK
{p.weber,b.bordbar,p.tino}@cs.bham.ac.uk

Abstract. Process mining uses event logs to learn and reason about business process models. Existing algorithms for mining the control-flow of processes in general do not take into account the probabilistic nature of the underlying process, which affects the behaviour of algorithms and the amount of data needed for confidence in mining. We contribute a first step towards a novel probabilistic framework within which to talk about approaches to process mining, and apply it to the well-known Alpha Algorithm. We show that knowledge of model structures and algorithm behaviour can be used to predict the number of traces needed for mining.

Keywords: Business process mining, probabilistic automata, Petri nets.

1 Introduction

Business processes describe sets of related activities which are carried out to solve a business problem, or produce a service or product. As a process is executed, the systems involved will record information in log files. Process mining [7] uses these logs to discover and analyse models of business processes.

As a simple example, consider the process in Fig.1. An order is received, stock checked, and either the item picked from the warehouse, or the order rejected. Despatch and billing take place in parallel, then payment may be chased repeatedly, before the order is closed. Abstracting from detail, the 'trace' of a single enactment of the process may be recorded as a string *abdefggh*. Process mining algorithms use logs of traces to produce models such as this Petri net.

Various techniques exist, reviewed in [7]. Other than [3,1], non-probabilistic languages (e.g. Petri nets, BPMN) are usually used to represent processes. The aim is usually to represent the control-flow structure in a model that is visually understandable, using heuristics [10] or clustering [6,2] to abstract from excessive detail or noise. Probabilities are generally not represented, and algorithms assume a 'complete' log, for some notion of completeness. Comparison of models is by syntactic methods such as replaying logs or Petri net token behaviour [5].

Little work has been done on systematically analysing process mining algorithms to discover their fundamental properties, or analysing the completeness of logs. Yet these aspects are of critical importance to enable confidence that the log is an adequate sample of the underlying behaviour, and thus in the accuracy of the mined model. While the core interest is in the control-flow of a process,

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 474–481, 2011.

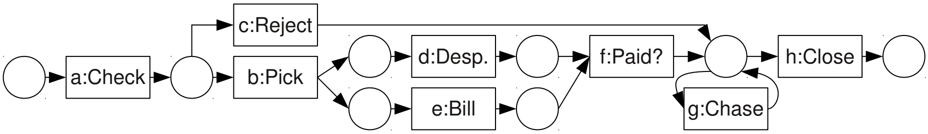


Fig. 1. Simplified Business Process for fulfilling an order

it must be appreciated that traces are generated randomly according to an underlying probability distribution unknown to the process mining algorithm. Not all activity sequences or decisions are equally likely, and their probabilities may have a dramatic effect on the amount of data needed for mining.

This paper contributes a first step towards a novel probabilistic framework within which to talk about approaches to process mining (section 2). We suggest a radically new view on process mining algorithms, in which a process is viewed as a distribution over traces of activities, and mining algorithms in terms of their ability to learn such distributions. We use probabilistic automata as a unifying representation, and compare models using distances between the probability distributions which they generate.

As an illustrative example, we apply this framework in section 3 to the foundational algorithm 'Alpha' [8]. Unlike previous methods, the framework allows us to answer in a principled manner the question of the probability of identifying the correct process from a given log of data. We show that a process model can be broken into structures and the probability estimated of correct mining of those structures, and thus of the original process model. Some experimental results are presented in section 4, and section 5 concludes the paper.

2 Processes as Distributions over Strings of Symbols

Similar to the approach in [1], we view processes as probability distributions over strings of symbols. Activities occur according to a "ground truth" process model $\mathcal{M}$ which may be unknown. We consider only acyclic process models, and place restrictions on processes equivalent to those used elsewhere, e.g. [8]: A process has a single start task s and end task e; the events of activities' occurrence are atomic (take no time) and are recorded as they occur in a workflow log W; and the underlying process model is fixed. A sequence of activities from start to end task is called a process trace. The log is therefore a multiset of traces.

Let Σ be an alphabet of symbols representing business activities. Process traces are represented by strings $\{x \in \Sigma^+\}$. $\mathcal{M}$ is therefore a stochastic regular language, describing a probability distribution $P_\mathcal{M}$ over Σ^+. The probability of trace x occurring is $P_\mathcal{M}(x)$, such that $\sum_{x \in \Sigma^+} P_\mathcal{M}(x) = 1$. The set of valid process traces is given by the finite support of $P_\mathcal{M}$. A process mining algorithm can therefore be viewed as learning a probability distribution $P_{\mathcal{M}'}$ over strings, to approximate $P_\mathcal{M}$, i.e. $P_{\mathcal{M}'}(x) \approx P_\mathcal{M}(x), \forall x \in \Sigma^+$. Learning is from the finite sample W drawn *i.i.d.* from the distribution to be learnt, $P_\mathcal{M}$.

For the purposes of analysis we use probabilistic deterministic finite automata (PDFA) [9], which have the bare minimum needed to represent distributions generated by business processes. A PDFA is a five-tuple $A = (Q_A, \Sigma, \delta_A, q_0, q_F)$, where Q_A is finite set of states including single start and end states q_0, q_F; Σ is an alphabet of symbols; and $\delta_A : Q_A \times \Sigma \times Q_A \to [0, 1]$ defines the probability function governing transition between states. The probabilities on transitions from a state sum to 1, and the transition function is deterministic: given a current state and symbol, the next state is certain, and there is a unique state path through A for any string x that it can parse. All states are accessible from the initial state, and from any state, it is possible to reach the final state.

PDFA A generates a probability distribution P_A on Σ^+:

$$P_A(x) = \delta_A(q_0, s_0, q_{s_0}) \times \left(\prod_{i=1}^{n-2} \delta_A(q_{s_{i-1}}, s_i, q_{s_i}) \right) \times \delta_A(q_{s_{n-2}}, s_{n-1}, q_F), \qquad (1)$$

where x is a string of symbols $s_0 s_1 \ldots s_{n-1}$ which can be parsed by the automaton to the unique final state q_F, and q_{s_i} denotes the state reached after symbol s_i is parsed. $P_A(x) = 0$ for strings which cannot be parsed.

Processes characterised in this way are equivalent to those represented by *sound* Workflow Nets [8], with the addition of probabilities on transitions. The restricted Hidden Markov Models used in [3] are similar in behaviour to our PDFA, since each state is restricted to a single output activity.

3 An Illustrative Example — The Alpha Algorithm

In this section we use this framework to analyse the behaviour of the process mining algorithm 'Alpha' with regard to the probability of it correctly re-discovering the process structures in a known ground truth, from a given log file.

The Alpha algorithm [8] makes a single pass through a workflow log to identify which tasks directly follow each other. This information is used to infer three basic relations between task pairs, which are used to construct a Petri Net:

- $a \to b$ (task b always follows a, never vice-versa),
- $a \# b$ (a and b never follow each other), and
- $a \parallel b$ (both ab and ba occur in the log).

A single start and end place are assumed, and the remaining places inferred using these relations. Two tasks are always related by $\to$, $\to^{-1}$, $\#$ or $\parallel$, and these relations partition the set of tasks [8, Property 3.1]. Acting on a pair of tasks, these relations also partition the set of all logs of n traces. Alpha is proven to mine processes representable by a sub-class of Petri nets, from noise-free logs.

Business processes are composed of structures (Fig.2,3). For acyclic processes, Alpha can discover sequences of tasks, exclusive (XOR) and parallel (AND) splits and joins. We give examples of these structures and how they may be represented by PDFA, and state the formulae for the probability of Alpha discovering them from a log of n process traces.

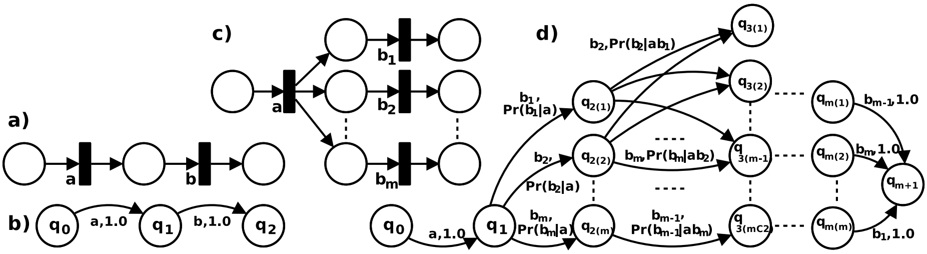


Fig. 2. Petri Net and PDFA fragments for Sequence a),b) and Parallel Split c),d)

We first state formulae for the probability of Alpha discovering each of the basic relations, when acting on a log of n traces, based on the probabilities of strings in the log. Let $\pi(ab)$ be shorthand for $P_{\mathcal{M}}(s\Sigma^* ab \Sigma^* e)$, the probability of ab occurring in a trace. We define $\pi_n(E)$ as "the probability of complex event E holding true in a log of n traces", and $P_\alpha(a \rightarrow_n b)$ as "the probability that Alpha infers the relation $a \rightarrow b$ over n traces", similarly for the other Alpha relations. Alpha will then discover the basic relations with the following probabilities:

$$\begin{aligned} P_\alpha(a \rightarrow_n b) =& \big(1 - \pi(ba)\big)^n - \big(1 - \pi(ab) - \pi(ba) + \pi(ab \wedge ba)\big)^n, && (2)\\ P_\alpha(a \,\#_n\, b) =& \big(1 - \pi(ab) - \pi(ba) + \pi(ab \wedge ba)\big)^n, \text{ and} && (3)\\ P_\alpha(a \,\|_n\, b) =& 1 - \big(1 - \pi(ab)\big)^n - \big(1 - \pi(ba)\big)^n \\ &+ \big(1 - \pi(ab) - \pi(ba) + \pi(ab \wedge ba)\big)^n. && (4) \end{aligned}$$

We next give exact formulae for the discovery of basic structures, and show how these may be usefully simplified without loss of accuracy. Due to space restrictions we do not provide full derivations; these will be published elsewhere.

3.1 Control-Flow Structures

Here we consider sequential activities, and exclusive and parallel splits and joins.

Sequential Activities: If a occurs, it is immediately followed *in the model* by b (Fig.2). In the log, other parallel tasks may 'interfere', so the following will hold: if a occurs in a trace, b will occur before the end of the trace. Discovery simply requires discovery of the causal relationship $a \rightarrow_n b$ (equation 2).

Splits and Joins: Alpha uses the relations →, # and || between pairs of tasks to locate places in the net, which characterises splits and joins as XOR or AND. As the discovery of a relation is a complex event arising from Alpha's interpretation of a log of n traces, they are not independent: multiple relations may be inferred, or not, from the log. Therefore to obtain exact probabilities for discovery of splits and joins it is necessary to use the probabilities of sub-strings which 'must' and "must not" be seen in the log, to build the formulae for larger structures.

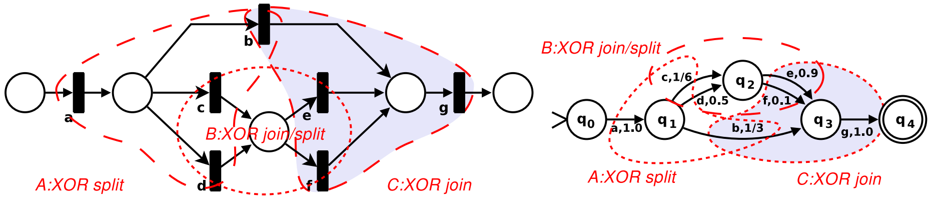


Fig. 3. Example Model as Petri Net and PDFA, Highlighting Structures

Exclusive Choice: XOR Split: An m-way XOR split (e.g. Fig.3 structure A) occurs where there is a choice between m exclusive paths through the model after task a, each path starting with a task $\{b_1 \dots b_m\}$. If a occurs in a trace, then exactly one $b_i \in \{b_1 \dots b_m\}$ will be included in the remainder of the trace.

To discover this split, Alpha must infer each $a \rightarrow_n b_i$ and each $b_i \#_n b_j$. So the log must contain at least one of each of m sub-strings ab_i, none of the m 'reverse' strings $b_i a$, and none of $\frac{m!}{(m-2)!}$ pairs of 'post-split' tasks. Let $N, Y \subset \Sigma \times \Sigma$ be the set of task pairs which *must not* (resp. *must*) be seen in the log. $S_n(X) \rightarrow [0,1]$, where $X \subseteq \Sigma \times \Sigma$, is the probability of not seeing any of the $|X|$ task pairs in n traces, and for $X_i = (t_i, t'_i) \in X, \pi(X_i) = \pi(t_i t'_i)$. Using the "inclusion-exclusion" principle for calculating the probability of intersecting events, applied to both strings within a trace, and traces within a log:

$$P_\alpha\big(a \rightarrow_n b_1 \# \dots \# b_m\big) = S_n(N) - \sum_{i=1}^{i=m} S_n(N \cup \{Y_i\}) +$$

$$\sum_{i,j=1:i<j}^{i,j=m} S_n(N \cup \{Y_i, Y_j\}) - \dots + (-1)^m S_n(N \cup Y), \text{ where} \tag{5}$$

$$S_n(X) = \Big(1 - \sum_{i=1}^{i=|X|} \pi(X_i) + \sum_{i,j=1:i<j}^{i,j=|X|} \pi(X_i \wedge X_j) \dots + (-1)^{|X|} \pi(X_1 \wedge \dots \wedge X_{|X|})\Big)^n .$$

Parallel Split: In an m-way AND split (Fig.2), after task a, m paths may proceed in parallel. Each path starts with a task $\{b_1 \dots b_m\}$. If a occurs in a trace, then the remainder of the trace will contain each $b_i \in \{b_1 \dots b_m\}$ before the end of the trace, in one of $m!$ permutations. Since PDFA do not explicitly represent parallelism, the fragment using XOR splits is more complex than the Petri net equivalent (Fig.2). After the first parallel task there are $\binom{m}{1}$ possible states, $\binom{m}{2}$ after the second, and so on to $\binom{m}{m-1}$ before the last parallel task.

The equations given for XOR splits can be modified to give the probability of discovery of XOR joins, and AND splits and joins, with sets Y and N populated with the required "must see" and "must not see" pairs of tasks.

3.2 Simplifying the Formulae

Given knowledge of the ground truth, many terms in these equations may be zero. Nevertheless, they can become cumbersome to work with, requiring knowledge of many probabilities. Nor do they relate intuitively to the working of the algorithm. Next we discuss how these formulae can be effectively simplified without loss of accuracy to give formulae which intuitively follow from the working of the Alpha algorithm, and are simpler to calculate. We denote the probability of discovery of structure S by Alpha, as $P_\alpha(S)$.

Lemma 1. *The probability of discovery of splits and joins may be usefully approximated by treating the probabilities of discovery of the Alpha relations over n traces as independent, and multiplying. The probability is over-stated but the error rate decreases exponentially with increasing n. For a general split/join structure (B in Fig.3) where m paths of which p are XOR (the remainder parallel) join and then split to n paths of which q are XOR (the remainder parallel):*

$$P_\alpha(S) \leq \prod_{i,j=1}^{i=m,j=n} P_\alpha(a_i \rightarrow_n b_j) \times \prod_{i,j=1:i<j}^{i,j=p} P_\alpha(a_i \#_n a_j) \times \prod_{i,j=1:i<j}^{i,j=q} P_\alpha(b_i \#_n b_j) \times$$
$$\prod_{i,j=(m-p):i<j}^{i,j=m} P_\alpha(a_i \parallel_n a_j) \times \prod_{i,j=(n-q):i<j}^{i,j=n} P_\alpha(b_i \parallel_n b_j) \qquad (6)$$

Due to space restrictions the proof will be published elsewhere. In summary, the error in the approximation is the difference between equations 5 and 6. Using $p_i \in [0,1]$ as shorthand for $\pi(ab_i)$, etc., this error is bounded by the sum of terms of the form $(1-p_i)^n(1-p_j)^n - (1-p_i-p_j)^n = (1-p_i-p_j+p_ip_j)^n - (1-p_i-p_j)^n$, which decay exponentially in n, after a maximum at relatively low n.

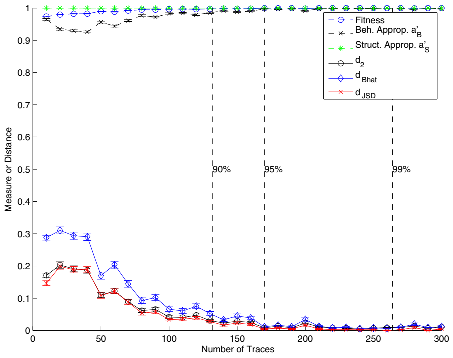
These probabilities for discovery of structures in a model can be combined to give the probability of successful mining by Alpha of a whole model.

4 Experiments

We designed a simple artificial process model (Fig.3) as the ground truth, and used the original (section 3.1) and simplified (3.2) formulae to predict the number of traces needed to mine a correct model (table 1). There is very little difference between the predictions, the simplification giving a slight underestimate. Logs in the MXML format were simulated from the automaton; 100 samples of logs from 10 to 300 traces in increments of 10, with a ground truth log of 10000 traces, assumed to be complete and distributed approximately according to the ground truth. Alpha in ProM (www.processmining.org) was used to mine these logs. Since Alpha produces non-probabilistic Petri nets, to compare the structure of the mined models with the ground truth structure, the nets were converted to probabilistic automata using their Reachability Graphs and labelling splits with maximum likelihood probabilities from the ground truth log.

Table 1. Predicted *vs.* Actual Number of Traces for Probability of Successful Mining

Probability	Exact Prediction	Simplified	Actual Traces
90%	132	131	90–100
95%	170	170	130–140
99%	263	262	280–290



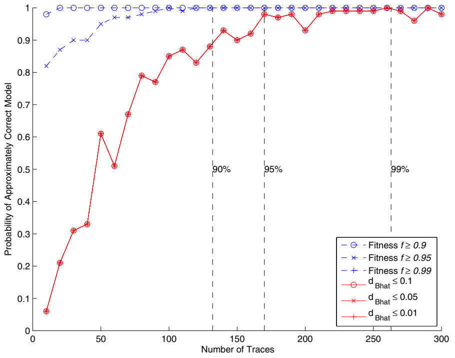


Fig. 4. Average Metrics Against Number of Traces

Fig. 5. Probability of 95% Approximately Correct Model

The d_2 and Bhattacharyya [4] distances, and the Jensen-Shannon Divergence (based on Kullback-Leibler) were used to calculate the average difference between the ground truth and automata mined from each log size. These and the 'Fitness' (recall) and Behavioural Appropriateness (precision) metrics [5] were plotted against number of traces (Fig.4). The graph shows that approximate correctness of the mined models converges at approximately the predicted points.

The distance measures seem more discriminating, being distributed over a clearer scale, compared with 'Fitness'. This is more apparent in Fig.5, which shows the *probability* of mining an approximately correct model, as measured by 'Fitness' (f) and the Bhattacharyya distance (D_{Bhat}), for various thresholds of approximate correctness. This is only a rough measure, as a single data point is calculated for each log file size; a count of the number of experiments from the 100 carried out for each size, for which the distance was below the threshold. The probability distance measure seems less sensitive to the threshold used, whereas 'Fitness' indicates convergence too soon, except for the 99% threshold.

5 Conclusion

Most process mining algorithms attempt to mine structural models of activities and relations between them, from a log that is assumed to be complete, and do not model probabilities. This does not provide for a way to know how much data is needed to be confident in mining results.

We suggest a novel probabilistic framework for considering business processes and process mining algorithms. The underlying business process is a distribution over strings of activities, and the primary task of mining the control-flow of the process is to learn this "ground truth" distribution, from a finite random sample of process traces which are drawn *i.i.d.* from the ground truth. Process mining algorithms then secondarily address additional requirements such as the representation language to use, or display detail or abstraction. Within this framework, process models may be compared using distances between the distributions which they generate, rather than ad-hoc or syntactic methods, and the behaviour of algorithms in terms of their convergence to the ground truth.

Applying this framework to the Alpha algorithm [8] we showed that using the structures in a model it is possible to accurately predict how much data will be needed to, with a given level of confidence, mine a model that is correct to a specified accuracy. We plan to apply this framework to other process mining algorithms and develop deeper learning theory relating to process mining.

Acknowledgments. P. Weber is supported by a Doctoral Training Grant funded by EPSRC and the School of Computer Science, University of Birmingham.

References

1. Ferreira, D.R., Gillblad, D.: Discovering Process Models from Unlabelled Event Logs. In: Dayal, U., Eder, J., Koehler, J., Reijers, H.A. (eds.) BPM 2009. LNCS, vol. 5701, pp. 143–158. Springer, Heidelberg (2009)
2. Günther, C.W., van der Aalst, W.M.P.: Fuzzy Mining – Adaptive Process Simplification Based on Multi-perspective Metrics. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 328–343. Springer, Heidelberg (2007)
3. Herbst, J., Karagiannis, D.: Integrating Machine Learning and Workflow Management to Support Acquisition and Adaption of Workflow Models. Int. J. Intell. Syst. Account. Financ. Manage. 9(2), 67–92 (2000)
4. Kailath, T.: Divergence and Bhattacharyya Distance Measures in Signal Selection. IEEE Trans Communication Technology CM-15(1), 52–60 (1967)
5. Rozinat, A., van der Aalst, W.M.P.: Conformance Checking of Processes Based on Monitoring Real Behavior. Information Systems 33(1), 64–95 (2008)
6. Song, M., Günther, C.W., van der Aalst, W.M.P.: Trace Clustering in Process Mining. In: Ardagna, D., Mecella, M., Yang, J. (eds.) Business Process Management Workshops. LNBIP, vol. 17, pp. 109–120. Springer, Heidelberg (2009)
7. Tiwari, A., Turner, C.J., Majeed, B.: A Review of Business Process Mining: State-of-the-Art and Future Trends. Bus. Process Manage. J. 14(1), 5–22 (2008)
8. van der Aalst, W.M.P., Weijters, T., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Trans. Knowl. Data Eng. 16(9), 1128–1142 (2004)
9. Vidal, E., Thollard, F., de la Higuera, F., Casacuberta, F., Carrasco, R.C.: Probabilistic Finite-State Machines - Part I. IEEE Trans. Pattern Anal. 27(7), 1013–1025 (2005)
10. Weijters, T., van der Aalst, W.M.P., Alves de Medeiros, A.K.: Process Mining with the Heuristics Miner Algorithm. BETA Working Paper Series, vol. 166. Eindhoven University of Technology, Edinhoven (2006)

Soft Computing Decision Support for a Steel Sheet Incremental Cold Shaping Process[*]

José Ramón Villar[1,**], Javier Sedano[2], Emilio Corchado[3], and Laura Puigpinós[4]

[1] University of Oviedo, Gijón Asturias 33204, Spain
villarjose@uniovi.es

[2] Instituto Tecnoló de Castilla y León, Poligono Industrial de Villalonquejar, Burgos, Spain
javier.sedano@itcl.es

[3] Computer Science and Automatica Department with the University of Salamanca,
Plaza de la Merced s/n 37008 Salamanca, Spain
escorchado@usal.es

[4] Fundación Privada Ascamm, Avda. Universitat Autònoma,
23 08290 Cerdanyola del Vallés, Spain
lpuigpinos@ascamm.com

Abstract. It is known that the complexity inherited in most of the new real world problems, for example, the cold rolled steel industrial process, increases as the computer capacity does. Higher performance requirements with a lower amount of data samples are needed due to the costs of generating new instances, specially in those processes where new technologies arise. This study is focused on the analysis and design of a novel decision support system for an incremental steel cold shaping process, where there is a lack of knowledge of which operating conditions are suitable for obtaining high quality results. The most suitable features have been found using a wrapper feature selection method, in which genetic algorithms and neural networks are hybridized. Some facts concerning the enhanced experimentation needed and the improvements in the algorithm are drawn.

Keywords: Wrapper Feature Selection, Genetic Algorithms, Neural Networks, Support Vector Machines, Incremental Cold Shaping.

1 Introduction

Over recent years there has been a high increase in the use of artificial intelligence and Soft Computing methods to solve real world problems. Many different applications have been reported: the use of Exploratory Projection Pursuit (EPS) [3] and ARMAX for modelling the manufacture of steel components [4], EPS and neural networks for determining the operation conditions in face milling operations and in pneumatic drilling process [10], genetic algorithms and programming for trading rule extraction [5] and

[*] This research has been partially funded by the projects TIN2008-06681-C06-04, TIN2010-21272-C02-01 and PID 560300-2009-11 of the Spanish Ministry of Science and Innovation, by the Junta de Castilla y León CCTT/10/BU/0002 project, by the FICYT project CN-08-028-IE07-60 and by the ITCL project CONSOCO.

[**] Corresponding author.

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 482–489, 2011.

low quality data in lighting control systems [14], feature selection and association rule discovery in high dimensional spaces [13] or neural networks (NN) and EPS in building energy efficiency [11,12].

It is known that the complexity inherited in most of the new real world problems, as for example, the steel cold shaping industrial process, is increasing as the computer capacity does. Higher performance requirements with a lower amount of data samples is needed due to the costs of generating new instances, specially in those processes where new technologies arise.

In this sense, the steel cold shaping is a relatively new technology in the production of low quantity lots steel pieces, which represents an effervescent area. Neural networks have been used to find relationships between the mechanical properties of the cold-rolled sheets of interstitial free and the chemical composition of the steel and the rolling and the batch annealing parameters [9]. Neural networks have been applied for identification of the parameters for operating conditions [18,19]. Up to our knowledge, no specific study has been published in steel iterative cold shaping.

This study focuses on determining the main parameters in an steel sheet incremental cold shaping. The main objective is to find the most relevant feature subset; the second objective is to obtain a decision support system in the operating conditions design, so the costs of producing such lots with a low amount of pieces is reduced. The next Section is concerned with the problem description. In Section 3 the algorithm used is detailed, while Sect. 4 deals with the experiments carried out and the results obtained. Finally, conclusions and future work are drawn.

2 Steel Incremental Cold Shaping

The metal incremental cold shaping is based on the concept of incremental deformation. This technology allows the manufacturing of pieces of metal sheet through the iteration of small sequential deformation stages until the desired shape is achieved and avoiding the axis-symmetric restrictions due to incremental rotatory deformation. Comparing incremental cold shaping with traditional deformation technologies it can be said that the former reduces the cost of specific machine tools and the manufacturing costs dramatically.

This type of technology has evolved from the well-known Rapid Manufacturing, allowing to generate pieces with complex geometries in a wide spread of materials without the need of frameworks or specific tools.

The main part of cold shaping has been controlled using numerical controlled tools in order to reduce as much as possible the fast, reliable, and costless manufacturing of lots with and small amount of metal pieces and prototypes.

The process of cold shaping starts with the design of a geometric shape in a 3D CAD file. This file should include as many layers as desired, each layer represents the bounds to be reached in each deforming step and are piled vertically. Consequently, the piece should be generated using sequential and incremental layers, each one at a different depth and constraint within the defined bounds.

Plenty of parameters have to be fixed to manufacture a metal piece, the force in each of the three dimensions to be develop by the deforming head, the speed change, the trajectory of the head, the surface roughness, the sheet pressure stress, the incremental step between layers, the number of steps or stages, the attack angle, the angle variation, the depth variation, etc. In Table 1, the range of the values of several of the input features is shown.

Table 1. Typical values of several of the variables involved in steel sheet incremental cold shaping

Variable	Units	Range of values
Sheet pressure stress	MPa	[75, 180]
Surface roughness about the tool	mm	[0.1, 0.5]
Speed change	mm/m	[800, 6000]
Force to be applied in the x-axis for shaping	N	[250, 500]
Force to be applied in the y-axis for shaping	N	[250, 500]
Force to be applied in the z-axis for shaping	N	[500, 1000]
Depth variation	mm	[0, 9]
Angle variation	degrees	[3, 15]
Number of incremental layers		[3, 6]
Incremental step between layers	mm	[0.3, 0.8]
Minimum thickness	mm	[0.28, 0.40]
Maximum depth achieved	mm	[-16.4, -39]

2.1 The Problem Definition

The first aim of this study is to evaluate if it is possible to model the operating conditions so the suitability of the experiment could be established, in other words, to analyse whether the operating conditions would generate a faulty piece or not while the most relevant features involved are to be selected.

The second aim is to model the maximum suitable depth that can be achieved with the given operating conditions. As in the former problem, the best feature subset is also required.

Therefore, there are two problems to solve, both including a feature selection process and a modelling process. While the former is a two-class problem, the second is a regression problem.

3 Feature Selection and Neural Networks

In order to obtain a suitable feature subset some requirements are needed. As there are integer features, nominal features and real valued features, the algorithm should deal with any kind of data. Therefore, the same approach should be valid for the both subproblems, the two-class problem and the maximum depth. Besides, not only the best feature subset for each problem but also the best model are desired, a classifier in the former case and a regression model in he latter.

It is known that for this kind of problems the wrapper approach for feature selection performs better than filter solutions [2,16]. These studies proposed wrapper feature selection methods using genetic algorithms (GA) [2] for dealing with the feature subset selection, that is, each individual is a feature subset. To evaluate individuals a modeling technique has been applied: the former proposed a lazy learning model as the K-Nearest Neighbour (KNN), the latter made use of a neural network (NN) method that iteratively fix the number of hidden neurons.

Different approaches as to how the NN is learnt have been studied. In [1] a GA approach to fingerprint feature selection is proposed and selected features are supplied as input to NN for fingerprint recognition, while in [15] a similar approach has been applied to automatic digital modulation recognition. Moreover, this type of approach has been reported to perform bettern that using statistical models [17]. Despite this, Support Vector Machines (SVM) have been also used in conjunction with evolutionary feature selection to reduce the input space dimensionality [6,7].

Algorithm 1. IND_EVALUATION: Evaluate an individual

```
**Require:** $I$ the input variables data set
**Require:** $O$ the output variable data set
**Require:** $ind$ the individual to evaluate, with its feature subset
  $model$ {the best model learned for $ind$}
  $mse = 0$ {the associated mean of Mean Square Error for $ind$}
  $indMSE = 0$ {best MSE found in the cross validation}
  **for** $k = 1$ to 10 **do**
    {run the $k$ fold in the cross validation scheme}
    generate the train and test reduced feature data set
    initialize the model $indModel$
    train indModel with the train data set
    $indKMSE \leftarrow$ calculate the MSE for $indModel$ with the test data set
    $mse+ = indKMSE$
    **if** $k == 1$ **or** $indMSE > indKMSE$ **then**
      $indMSE = indKMSE$
      $model = indModel$
    **end if**
  **end for**
  $mse = mse/10$
  **return** $[model, mse]$
```

In this study we adopt two different solutions depending whether we are dealing with the two-class or the maximum depth problem. An hybridized method of GA evolving the feature subsets and a SVM classifier is chosen in the former case, while in the latter an hybridized method of GA evolving the feature subsets and a NN for modeling the desired output is used. TIn both modelling and feature selection problems the GA is an steady state approach with the percentage of elite individuals to be defined as a parameter. The algorithm is outlined in Algorithms [1,2].

The typical steady state GA parameters, like the crossover and mutation probabilities, the number of generations, the population size and the elite population size, are all

Algorithm 2. GA$^+$ Feature Selection

```
Require: $I$ the input variables data set
Require: $O$ the output variable data set
Require: $N$ the feature subset size
  $FS \leftarrow \{ \emptyset\}$ {the best feature subset}
  $model$ {the model learned for $FS$}
  $mse = 0$ {the associated mean of Mean Square Error for $FS$}
  Generate the initial population, $Pop$
  for all individual $ind$ in $Pop$ do
    $[ind.model, ind.mse] = IND_E VALUATION(I, O, ind)$
  end for
  $g \leftarrow 0$
  while $g < G$ do
    while $size(Pop') < (popSize - |E|)$ do
      Generate new individuals through selection, crossover and mutation
      add valid individuals to $Pop'$
    end while
    extract the elite subpopulation $E \in Pop$
    for all individual $ind$ in $Pop'$ do
      $[ind.model, ind.mse] = IND_E VALUATION(I, O, ind)$
    end for
    $Pop = \{E \cup Pop'\}$
    sort $Pop$
    g++
  end while
  $FS \leftarrow Pop[0]$
  $[model, mse] \leftarrow$ corresponding model and MSE
  return $[FS, model, mse]$
```

of them given for each experiment. The individual representation is the string of indexes of the chosen feature subset. The tournament selection is implemented and one point crossover is used. After each genetic operation the validity of the off-prints is analysis: repeated features indexes are erased and random indexes are introduced to fill the individual feature subset.

Third order polynomials are used as kernel functions for the SVM. The number of hidden nodes in the NN is set as a parameter. The NN models are generated randomly and trained. In all cases, 10-fold cross validation is used, and the mean value of the mean squared error in each fold is the fitness of an individual.

4 Experiments and Results

Generating data set samples is costly as each one of the samples needs a real case to be carried out, that is, a sheet of steel has to be cold shaped; consequently, the smaller the number of experiments, the lower the cost and the smaller the data set size.

The data set comprises 19 samples, each one with the whole set of parameters values. Once the piece is processed as the corresponding sample establishes, then it is manually classified as {GOOD, BAD} according to the deformation or the quality faults that could appear in the piece. Besides, the maximum depth in each case is also measured. These two latter values are appended to each sample as the output variables.

As SVM and NN are to be used in the modelling part of the feature selection GA method, then the data set is normalized with means 0 and deviations 1.

In the experimentation, the GA parameters have been fixed as follows: 50 individuals in the population, 100 of generations, the probability of crossover equals to 0.75, while the mutation probability is 0.25. An steady state GA evolutionary scheme is used, with a number of 5 elite individuals that will be kept in the next generation.

The size of the feature subset has been fixed to three. The SVM kernel function is fixed as third order polynomials and the feed forward back-propagation NNs includes 6 neurons in the hidden layer. The parameters of the SVM and the NN have been kept constant during the feature selection and model learning.

As stated in the previous section, the 10-fold cross validation schema is carried out. Only the validation results are used to compare individuals. For the two-class problem the mean of the classification errors among the folds is used to evaluate the models and the feature subsets. For the maximum depth estimation, each feature subset and its corresponding model are evaluated with the mean of the mean squared error on each fold.

In the case of the two-class problem, the best feature subset found includes the *step increment*, the *angle variation* and the *variation in depth*, with a mean classification error of 0.1. For the second problem, the best feature subset found includes the variables *step increment*, the *number of stages* and *variation in depth*, with a mean error of 0.0096.

It is worth mentioning that the reduced number of data samples induces relatively high error values as the test data set includes only one or two examples. More accuracy should be obtained if a bigger number of samples is given. However, the cost of the data gathering increases considerably; this dilemma should be evaluated.

Moreover, the algorithms do not include local optimization of the models parameters. So it is possible that better performance of the models and the feature selection process could be more affordable if such local optimization were implemented.

Finally, the maximum depth have been found regardless of the two-class problem, which was not the expected result in the expert opinion. It is though that the above mentioned local optimization of the models parameters should improve the performance and the experts confidence in the results.

5 Conclusions and Future Work

This study introduces a feature selection method for choosing the best feature subset in a steel sheets cold shaping process divided in a two-class problem and a maximum depth estimation problem. Moreover, a genetic algorithm is hybridized, on the one hand, for the first case, with a support vector machine model to choose the best feature subset and on the other hand, for the second case, with a feed foward back-propagation neural network .

From the experimentation the best feature subset has been found for both problems, and some relevant facts have arisen. Firstly, the data set size should be increased in order to obtain better models fitness values. Secondly, local optimization for the models parameters should improve the obtained results. Finally, it could be desirable that the optimum number of features should be dynamically fixed, which represents an improvement in the individual representation and is left for future work. Future work also includes evaluating the approach in more detail and including a comparison to the related work.

References

1. Altun, A.A., Allahverdi, N.: Neural network based recognition by using genetic algorithm for feature selection of enhanced fingerprints. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007, Part II. LNCS, vol. 4432, pp. 467–476. Springer, Heidelberg (2007)
2. Casillas, J., Cordón, O., del Jesus, M.J., Herrera, F.: Genetic Feature Selection in a Fuzzy Rule-Based Classification System Learning Process. Information Sciences 136(1-4), 135–157 (2001)
3. Corchado, E., MacDonald, D., Fyfe, C.: Maximum and Minimum Likelihood Hebbian Learning for Exploratory Projection Pursuit. Data Min. Knowl. Discov. 8(3), 203–225 (2004)
4. Corchado, E., Sedano, J., Curiel, L., Villar, J.R.: Optimizing the operating conditions in a high precision industrial process using soft computing techniques. Expert Systems (2011) (in press)
5. de la Cal, E., Fernández, E.M., Quiroga, R., Villar, J., Sedano, J.: Scalability of a Methodology for Generating Technical Trading Rules with GAPs Based on Risk-Return Adjustment and Incremental Training. In: Corchado, E., Graña Romay, M., Manhaes Savio, A. (eds.) HAIS 2010. LNCS, vol. 6077, pp. 143–150. Springer, Heidelberg (2010)
6. Fung, G.M., Mangasarian, O.L.: A Feature Selection Newton Method for Support Vector Machine Classification. Computational Optimization and Applications 28(2), 185–202 (2004)
7. Huanga, C.-L., Wang, C.-J.: A GA-based feature selection and parameters optimizationfor support vector machines. Expert Systems with Applications 31(2), 231–240 (2006)
8. The MathWorks, MATLAB - The Language Of Technical Computing (2011), http://www.mathworks.com/products/matlab/
9. Mohanty, I., Datta, S., Bhattacharjeeb, D.: Composition-Processing-Property Correlation of Cold-Rolled IF Steel Sheets Using Neural Network. Materials and Manufacturing Processes 24(1), 100–105 (2009)
10. Sedano, J., Corchado, E., Curiel, L., Villar, J., Bravo, P.: The Application of a Two-Step AI Model to an Automated Pneumatic Drilling Process. Int. J. of Comp. Mat. 86(10-11), 1769–1777 (2008)
11. Sedano, J., Curiel, L., Corchado, E., de la Cal, E., Villar, J.R.: A Soft Computing Based Method for Detecting Lifetime Building Thermal Insulation Failures. Int. Comp.-Aided Eng. 17(2), 103–115 (2009)
12. Sedano, J., Villar, J.R., Curiel, L., de la Cal, E., Corchado, E.: Improving Energy Efficiency in Buildings Using Machine Intelligence. In: Corchado, E., Yin, H. (eds.) IDEAL 2009. LNCS, vol. 5788, pp. 773–782. Springer, Heidelberg (2009)
13. Villar, J.R., Suárez, M.R., Sedano, J., Mateos, F.: Unsupervised Feature Selection in High Dimensional Spaces and Uncertainty. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 565–572. Springer, Heidelberg (2009)
14. Villar, J.R., Berzosa, A., de la Cal, E., Sedano, J., García-Tamargo, M.: Multi-objective Simulated Annealing in Genetic Algorithm and Programming learning with low quality data. In: Publication for Neural Computing (2011)

15. Wong, M.L.D., Nandi, A.K.: Automatic digital modulation recognition using artificial neural network and genetic algorithm. Signal Proc. 84(2), 351–365 (2004)
16. Yang, J., Honavar, V.: Feature Subset Selection Using a Genetic Algorithm. IEEE Intelligent Systems 13(2), 44–49 (1998)
17. Zhang, P., Verma, B., Kumar, K.: Neural vs. statistical classifier in conjunction with genetic algorithm based feature selection. Pat. Recog. Letters 28(7), 909–919 (2005)
18. Zhao, J., Cao, H.Q., Ma, L.X., Wang, F.Q., Li, S.B.: Study on intelligent control technology for the deep drawing of an axi-symmetric shell part. J. of Materials Processing Tech. 151(1-3), 98–104 (2005)
19. Zhao, J., Wang, F.: Parameter identification by neural network for intelligent deep drawing of axisymmetric workpieces. J. of Materials Processing Tech. 166(3), 387–391 (2005)

Eigenlights: Recovering Illumination from Face Images

James Burnstone and Hujun Yin

School of Electrical and Electronic Engineering
The University of Manchester
Manchester, M13 9PL, UK
james.burnstone-2@postgrad.manchester.ac.uk,
h.yin@manchester.ac.uk

Abstract. In this paper, we present the use of subspace modelling to find the basis features of illumination across human face images. Instead of using a real image data set we use computer-generated 3D models, which we have built. Using these models we can better investigate the effect of the recognition of faces under illumination not confined within a particular trained subset. With this we have designed a recognition system where we investigate how many training images are necessary to build an illumination subspace that gives robust recognition. We aim to apply this technique to deal with the lighting problem in face recognition on mobile devices where some current methods are simply too complex to use.

1 Introduction

Face recognition has become a popular and preferred biometric means amid increased security applications. However, with over 25 years of research, illumination is still a major problem in face recognition standing in the way of a robust self-sufficient system. At present 3D models and advanced representations of lighting conditions by spherical harmonics is being used [1, 2]. The initial work into illumination compensation began as a search to find all the images of a face under all types of illumination [3]. With the success of appearance-based methods such as PCA and LDA, there became a need to find all the images of a face under all possible illumination conditions to be used for training [5]. Although promising results were obtained using linear subspaces and manifolds, there was too much offline processing and curse of dimensionality remained as the number and size of these subspaces increased.

With the increase of handheld devices such as mobile phones and tablet PCs, there is more focus than ever to apply these practices to smaller devices, which, although increasing in their complexity rapidly, are still a challenge for real-time image processing. At present some of the most powerful devices contain 1000MHz processing with 750MB of RAM, large image processing tasks can simply breaks down and crashes. The emphasis is therefore placed on dimensionality reduction and efficient processing.

Many literature studies of face recognition show that the problem is almost solved (under fixed lighting conditions). A recent Face Recognition Vendor Test demonstrated many systems with error rates of only 0.01. Although this seems promising many of these advanced methods cannot be applied to real-world systems where images are

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 490–497, 2011.

of low-quality and often scaled or orientated in a way difficult to suit recognition. Several face recognition systems under test have further exposed the challenges in practical applications.

We aim to deal with illumination problem by finding the basis vectors of a collection of images of single subjects taken under a wide variety of illumination conditions. Using PCA on data gives a set of basis features that represent the variances in the data set. If a data set of face images varies by illumination conditions, PCA gives a set of principal lighting conditions from which all other lighting conditions can be reconstructed. This approach is well known and is used in a method known as subspace modeling [4, 6, 7]. The method of using PCA to create a low-dimensional collection of images is combined with spherical harmonic representation and Morphable models to build recognition systems [7]. In most cases the subspaces are built using defined image sets, such as the PIE or YALE database. This approach involves synthesis of new images of faces artificially under new or unseen illumination conditions. In this paper the aim was to investigate the ability of this subspace modeling using CGI 3D models. The 3D models can be illuminated under a combination of ambient lighting, diffuse spot lights, area lighting and representation of sun light. By building subsets of different combinations of these illumination conditions different sets can be used for training of the subspace and testing. The images taken using the 3D models represent illumination in an uncontrolled environment. While there are many subspace modeling algorithms that give high recognition results, they are tested on images from controlled environment and do not represent real world stimuli.

For recognition, we use these basis features, which we term as Eigenlights, to decompose a novel image into the contribution of each basis feature to the unknown illumination in the image. We then apply these contributions to each subject in the gallery by building an image of that subject under the novel images illumination conditions. This means we have an image of each subject in the gallery under the same illumination conditions as the novel image. Classification is conducted using the Nearest-Neighbour algorithm between the novel image and the constructed images from the gallery. For each experiment the type of illumination conditions are changed, as are the number of images used to construct the subspaces.

2 Finding the Eigenlights

Whereas previous methods have used selections of available image databases, in this approach we use rendered images of 3D models. At present these 3D models are considered life-like and robust enough for face recognition experiments where images of real individuals are usually processed and scaled into slightly less life-like appearances. The true potential of using 3D models is that we have access to unlimited number of individuals, for which we can place them in numerous poses and expressions, and can realistically account for changes in subjects' age, weight and skin tone. This gives us a freedom not available when using any image database and is more efficient than taking our own images of real subjects under these varying conditions. 3D models allow us to simulate a large number of lighting conditions using realistic ray-tracing and Lambertian and Phong reflectance function models which are used intensively in the literature [8].

In our initial experiments we have a set of eight 3D models, which account for a wide selection of face shape, age and skin tone, all in a frontal pose with neural expression. For each subject, around 100 images were taken across a wide variety of lighting conditions by moving a single point light source across a region around the front hemisphere of the face. Once we had gathered enough images to cover a reasonable amount of the entire possible variation of lighting, we can use PCA to find the basis features for all possible illumination conditions on a face.

PCA in face recognition is widely used for dimensionality reduction [9]. PCA creates a set of basis vectors for which all the original data vectors can be recovered by a weighted linear combination of these basis vectors.

$$\Gamma_n = [\omega_n^1 \phi_1 + \omega_n^2 \phi_2 + ... + \omega_n^N \phi_N] + \psi \tag{1}$$

where $\{\phi_1, \phi_2, ..., \phi_N\}$ are the Eigenvectors calculated from the covariance matrix of the data, $\{\omega^1{}_n, \omega^2{}_n, ..., \omega^N{}_n\}$ corresponds to the weights or coefficients of an original vector projected onto the Eigenvector space, and ψ is the mean image.

The Eigenvectors correspond to the directions of variances in the data set. There are as many Eigenvectors as dimensions of the data, ordered by their scalar equivalent, known as the Eigenvalues. These scalar values show the amount of influence of a particular direction of variance on the data set. It is possible to only use those corresponding to the largest Eigenvalues for reconstruction as data can be faithfully restored with these most important Eigenvectors. This means we only need to store the most important Eigenvectors, whose number is usually much less than the dimensions of the original data, and yet we can still recover each vector in the original data set with minimal loss of information. Fig. 1 shows the first five eigenvectors of an example.

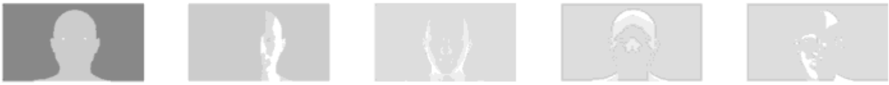


Fig. 1. The basis features of lighting

Once these Eigenvectors, the basis features or Eigenlights as we call them, are established, a new vector, not in the original data set, can be projected on the space to create its own weights. This means we are able to recover a linear combination of the basis features for any novel vector. When a vector is projected onto the eigen space, its length along each dimension is calculated. In the face space each dimension is an Eigenface this case can be thought of as finding the contribution of each Eigenface to the novel image. Mathematically this is achieved by taking the dot product of the novel vector and all Eigenfaces. Once we have these contributions or weights, we can build an image using Eq. (1).

Previous work has shown that subspaces can be formed for the collection of images of a face under varying lighting conditions [4]. These subspaces are calculated analytically and contain assumptions due to shadowing. We hope this approach, though simpler, will be more suited to being applied to real-time face recognition systems. With the image sets of eight different 3D face models, we calculated the basis images using PCA. The eigenvalues show how much of the variance of the data set is contained in the corresponding basis images. By ordering the Eigenvectors in descending Eigenvalues one can see how many of the most important vectors are needed to represent enough of the variance.

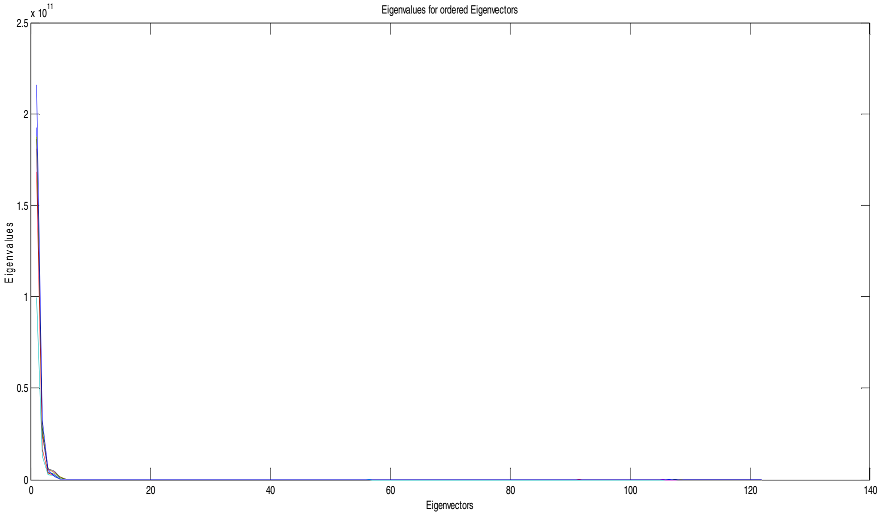
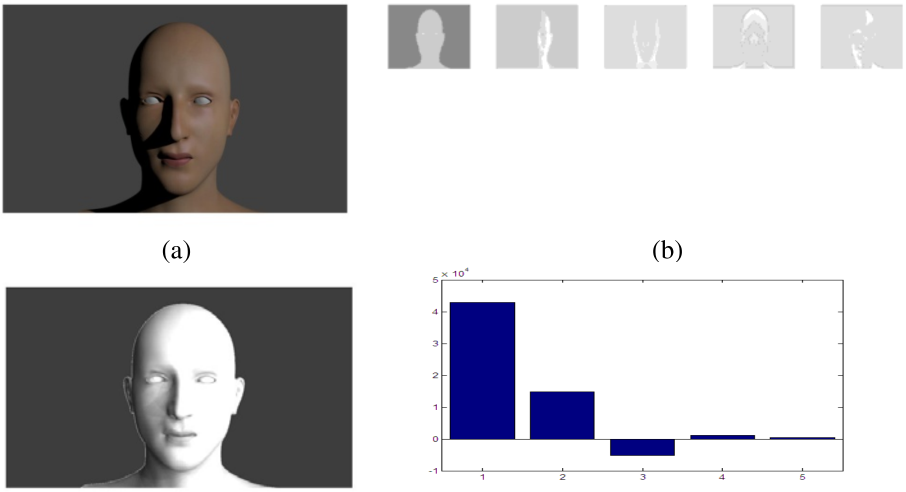


Fig. 2. Eigenvalues of corresponding Eigenvector, showing the amount of the variance of the data set they contain

Fig. 2 shows a plot of the Eigenvalues for all the subjects. The values have been sorted with the highest values at the first and the least important at end, 120-140, depending on the number of images of that subject. The *y* axis shows the scales, i.e. the amount of variance for each Eigenvector. As it can be seen most of the variance of all the subjects is contained within the first five Eigenvectors, consistent with previous findings [10]. Therefore these first five Eigenlights describe a subspace in the face space that contains nearly all of the possible images of a face under all lighting directions used in our tests.

Therefore lighting on any face can be broken down to five main directions. This is addressed in many previous studies, where it is shown that the lighting conditions for any object should lie on a low-dimensional manifold in the vector space, [3]. It has also been noted that as there is an infinite number of lighting conditions, to calculate these by real images was too big a task. In our approach we believe that by using a large enough set of deformable 3D models we may be able find some patterns that cover enough variance of the lighting. To test how well these basis light images, or Eigenlights, perform we use some novel images of each subject under random single or multiple light sources and then projected these on these Eigenlights vector space. This creates weights for the novel image, corresponding to the contribution of each Eigenlight to the image. We then reconstruct the novel image from only these weights and the Eigenlight and compare with the original, as shown in Fig. 3.



(a) (b)

(c) (d)

Fig. 3. (a) novel face image, (b) the Eigenlights, (c) recovered face, and (d) weights

The recovered image shows that the illumination effect has been removed or significantly reduced. However, one cannot use a subject's Eigenlights on another as they depend on the geometry of the face. For each subject the Eigenlights found so far are similar in terms of the lighting directions. Therefore the Eigenlights form a 5D shape in the face space. However, due to the complexity and differences in shape and features in human faces, one is unable to directly predict this subspace for a new subject. It has been described [4] that it would be impossible to predict illumination subspaces from real images. However recent papers in subspace modeling including [6], use statistically models to predict the lighting conditions of an unknown face. The initial subspace is contrasted with a bootstrapped set of cropped and aligned images. It is then possible to predict the lighting conditions of an unseen face by finding the best match in the subspace.

3 Synthesis under Different Lighting Conditions

It is intuitive that if a subspace has been built with a range of illumination conditions, any unseen image that comprises of lighting conditions contained within the scope of the training set can be recovered. It is more informative to test the subspace with images of lighting conditions far different to that observed in the training set. Using the 3D models gives a freedom to invite any sort of lighting conditions, including realistic representations of natural sunlight, or normal lighting found in buildings with multiple sources.

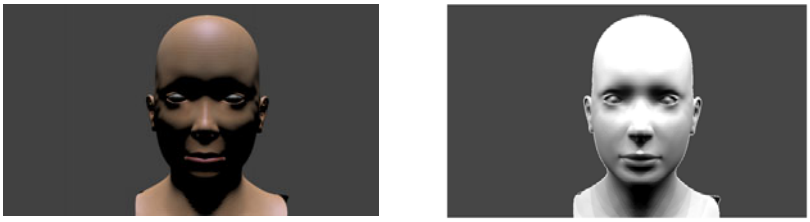


Fig. 4. Novel face image under sun and point source illumination and recovered image

In Fig.4 a test image comprises of an overhead sun light source and a point light source below. This is far from any image taken in the training set. However, the recovered image clearly shows the correct identity of the subject with facial features enacted. There are shadows in the correct regions, esp. above the eyebrows however the lit areas in the test image appear as specular highlights in the recovered image.

4 Recognition Experiment

If the illumination condition of an image is known before recognition is performed, one can eliminate the variation in the image set due to the lighting. Therefore recognition can be purely based on the geometry of the face and hence the identity of the subject. At this stage we assume a standard pose and expression for all images.

To remove variation due to lighting, we find the illumination conditions of the novel image, and then apply them to the images in the gallery set. For each subject in the set we have the 5D subspace. When a novel (test) image is presented, it is projected onto the Eigenlights of each subject. This gives the image's weights for all basis light features. Then these weights are used to build a similar image of each subject in the gallery from their own Eigenlights. A nearest-neighbour classier is then used on the new images measured against the novel image. The novel image and each subject's image all share the same illumination conditions and so the variation due to lighting is removed. The images of each subject are split into subsets, grouped by the amount of illumination ranging from normal frontal in subset 1, mild shadowing in subset 2, extreme shadowing in subset 3, etc. An example of the first three subsets is shown in Fig. 4.

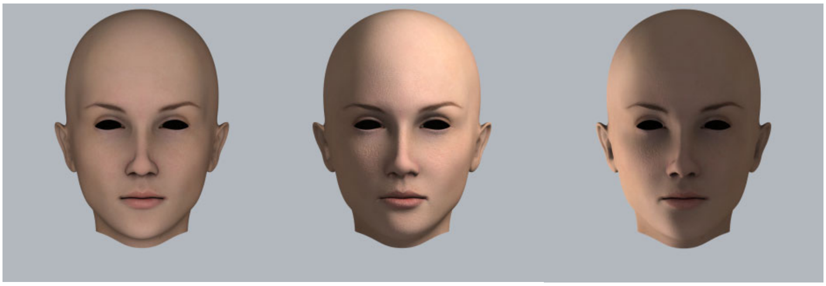


Fig. 5. Face images from subsets 1, 2 and 3

For evaluation of the method, the entire data set was randomly split into training and test sets. The Eignelights of each subject were subsequently learnt from the training set. Then the performance, or recognition rate, on the test set was measured. For a comparison we have also tested recognition using a standard Eigenface method, with Homomorphic filtering [11], using the same training and test sets. The experiment was repeated independently for ten times. The averaged results are listed in Table 1. Please note that Eigenface method is rarely directly applied to images of variable lighting conditions. The comparison here only serves as a reference.

Table 1. Recognition performance

Training/Testing Images %	EigenLights (Recognition Rate)	EigenFace (Recognition Rate)
30:70	98.44%	30.5%
10:90	87.9%	24.77%

In these test only eight subjects were used. It is fair to say that a much larger set would be necessary to give more faithfully results, though as long as we have each subject Eigenlights we are able to recover a near perfect representation of an image of the subject under any lighting, so yielding good recognition rates.

5 Conclusion and Future Work

In this paper we investigate the method of subspace modeling to represent changes in illumination in face images as a low-dimensional manifold in the face space. Computer-generated 3D models are used to test the robustness of the method by using a wide variety of different lighting conditions not usually found in controlled environments of any face image databases. The PCA methods in the literature have previously shown promising results when systems are trained with these face image database of little lighting variations and then tested with them. Extreme lighting and lighting from different or multiple sources is less well recovered. Our investigation has shown there are still a wide variety of illuminations conditions which are not confined within a simple 5-dimensional subspace, yet they are apparent to any real world facial recognition system.

In the experiment we have shown that promising results are achieved by relighting the face images with the illumination conditions recovered from a test image, and that we can build a basis feature subspace without prior images of all possible illumination conditions. The method requires a hundred or so images of a subject under uniformed, variable lighting to build the 5D subspace which gives over 98% recognition. In a real world deployed system it would be expected to have even higher results. At present these images are unique to each subject. This may prove difficult if one wants to use these findings in a large recognition system as one would need to find a way to recover this subspace without so many images. Therefore instead of building a unique subspace for each subject we could design a normalized set of Eigenlights projected on a mean shape. Previous work has shown that faces can be deformed to the mean shape in a data set, and recovered by adjusted the variance [12]. If all faces are plotted in a pdf curve, by adjusted the standard deviation you can move along the set of faces. Then, each novel face can be adjusted into this mean shape, projected onto the normalized Eigenlights, and then the recovered illumination conditions can be applied to each subject in the gallery.

References

1. Blanz, V., Romdhani, S., Vetter, T.: Face identification across different poses and illumination with a 3D morphable model. In: Proc. 5th Int'l Conf. Automatic Face and Gesture Recognition, pp. 192–197 (2002)
2. Zhang, L., Samaras, D.: Face recognition from a single image under arbitrary unknown lighting using spherical harmonics. IEEE Trans. Pattern Analysis and Machine Intelligence 28, 209 (2006)
3. Belhumeur, P.N., Kriegman, D.J.: What is the set of images of an objects under all possible illumination conditions? Internal Journal of Computer Vision 28(3), 245–260 (1998)
4. Basri, R., Jacobs, D.W.: Lambertian reflectance and linear subspaces. IEEE Trans. Pattern Analysis and Machine Intelligence 25(2), 218–233 (2001)
5. Lee, K., Ho, J., Kriegman, D.: Acquiring linear subspaces for face recognition under variable lighting. IEEE Trans. Pattern Analysis and Machine Intelligence 27(5), 684–695 (2005)
6. Shim, H., Luo, J., Chen, T.: A subspace model-based approach to face relighting under unknown lighting and pose. IEEE Trans. on Image Processing 17(8), 1331–1341 (2008)
7. Wang, Y., Liu, G., Wen, Z., Zhang, Z., Samaras, D.: Face re-lighting from a single image under harsh lighting conditions. IEEE Trans. Pattern Analysis and Machine Intelligence 31(11), 1968–1984 (2009)
8. Ramamoorthi, R., Hanrahan, P.: On the relationship between radiance and irradiance: determining the illumination from images of a convex Lambertian object. J. Optical Soc. Am, A. 18(10), 2448–2459 (2001)
9. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. J. Cognitive Neuroscience 3(1), 71–86 (1991)
10. Ramamoorthi, R.: Analytic PCA construction for theoretical analysis of lighting variability in images of a Lambertian object. IEEE Trans. Pattern Analysis and Machine Intelligence 24(10), 1322 (2002)
11. Delac, K., Grgic, M., Kos, T.: Sub-image homomorphic filtering for improving facial identification under difficult illumination conditions. In: Int'l Conf. on Systems, Signals and Image Processing, Budapest (2006)
12. Edwards, G.J., Lanitis, A., Taylor, C., Cootes, T.: Statistical models of face images – improving specificity. Image and Vision Computing 16, 203–211 (1998)

Novel Data Mining Approaches for Detecting Quantitative Trait Loci of Bone Mineral Density in Genome-Wide Linkage Analysis

Qiao Li[1], Alexander J. MacGregor[2], and Wenjia Wang[3]

[1] School of Public Health, Imperial College London, London, W2 1PG, UK
[2] School of Medicine, University of East Anglia, Norwich, NR4 7TJ, UK
[3] School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK

Abstract. Haseman-Elston (H-E) regression is a commonly used conventional approach for detecting quantitative trait loci (QTLs), which regulate the quantitative phenotype based on the Identical-By-Descent (IBD) information between twins in Genome-wide scan. However, this approach only considers genetic effect at individual loci, but not any interaction between genes. A Pair-Wise H-E regression (PWH-E) and a Feature Screening Approach (FSA) are proposed in this paper to take gene-gene interaction into account when detecting QTLs. After testing these approaches with several series of simulation studies, they are applied to a real-world bone mineral density (BMD) dataset, and find three site specific sets of potential QTLs. Further comparison analyses show that our results not only corroborate the 14 findings from previous published studies, but also suggest 22 new QTLs of BMD.

Keywords: Quantitative trait loci, Haseman-Elston regression, bone mineral density.

1 Introduction

The DNA regions that contain or link to genes influencing a quantitative phenotypic trait are known as quantitative trait loci (QTLs) and finding QTLs will help to understand the genetic structure and variations of a trait. Linkage analysis is an important analytical approach for identifying quantitative trait loci (QTLs). This paper presents two new approaches of linkage analysis to uncover QTLs that influence Bone Mineral Density (BMD). A real-world BMD dataset studied in this paper includes quantitative BMD phenotypes at three body sites (spine, hip and forearm) and genetic information relating to Identical-By-Descent (IBD) alleles in twins.

Complex phenotypes are usually influenced simultaneously by multiple QTLs individually and/or their interactions, as illustrated in Fig. 1. Genetic influences from individual QTLs alone are usually referred as main genetic effects. Gene-gene interactions are increasingly recognized as important phenomena in the field of genetic epidemiological studies [1]. The presence of gene-gene interaction presents a

H. Yin, W. Wang, and V. Rayward-Smith (Eds.): IDEAL 2011, LNCS 6936, pp. 498–509, 2011.

particular challenge in the search for QTLs, because if the effect from one QTL is altered or masked by effect from another QTL, the power to detect these two QTLs is likely to be reduced and unveiling the joint effects at the two loci will be hindered by their interaction [2]. The form of biological interactions between genes is complex, and there is often little prior information from biological research to inform the development of hypothetical models. Various statistical models can be hypothesized to represent the genetic interaction effect on quantitative phenotypes. A commonly used statistical interaction model is the multiplicative model [3], which is applied here in simulation studies. The conventional non-parametric linkage method, H-E regression [4, 5] is based on the process of estimating the main effects of individual genetic markers one by one in a genome-wide scan. It is not designed for detecting QTLs that may have gene-gene interactions.

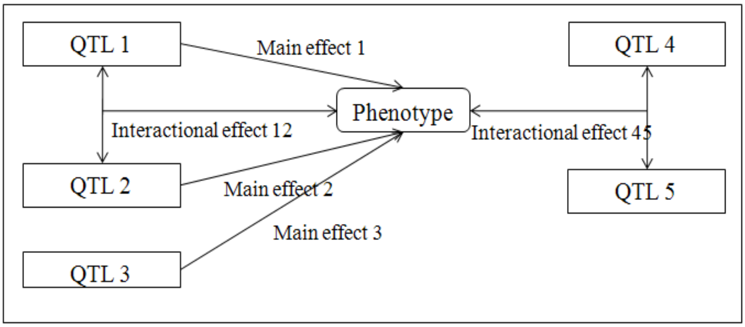


Fig. 1. Three typical situations of QTLs influencing a phenotype: 1) QTL1 and QTL2 have both main and interactional effects on the phenotype; 2) QTL3 has main effect on the phenotype; 3) QTL4 and QTL5 have interactional effect on the phenotype.

This paper presents two new approaches, a Pair-Wise H-E regression (PWH-E) and a Feature Screening Approach (FSA), to detect QTLs allowing interactions. These two new approaches and the conventional H-E method are tested on simulated data generated under different models to verify their ability to detect QTLs that accommodate genetic interaction. These approaches are then applied to a real-world bone mineral density dataset of twins. The outputs are compared with the results of previous linkage and association analyses on BMD.

The remainder of this paper is organized as following: Section 2 introduces the methods used in this research, including H-E, PWH-E and FSA. Section 3 presents simulation studies for verifying and comparing these three methods. In Section 4, a real-world data is analysed for uncovering the QTLs of BMD.

2 Methods

As the aim of this study is to use data of genome-wide microsatellite markers to find the chromosomal position of QTLs associated with BMD, we devised a methodological framework as shown in Fig. 2. including two new methods (PWH-E and FSA) and one conventional method (H-E) as the comparison baseline. The PWH-E and FSA were designed for detecting QTLs, especially when gene-gene interactions may dominate the genetic influences. They both involve a feature selection step.

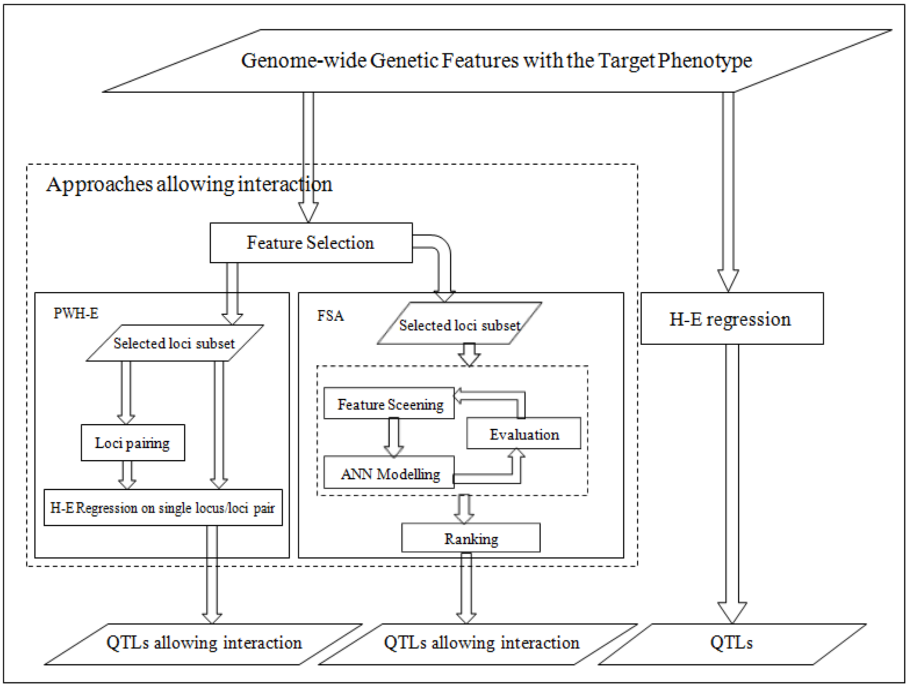

Genome-wide Genetic Features with the Target Phenotype
Approaches allowing interaction
Feature Selection
PWH-E
Selected loci subset
Loci pairing
H-E Regression on single locus/loci pair
FSA
Selected loci subset
Feature Sceening
Evaluation
ANN Modelling
Ranking
H-E regression
QTLs allowing interaction
QTLs allowing interaction
QTLs

Fig. 2. Methodology framework for genome-wide linkage analysis, including three methods: H-E regression, PWH-E (Pair-Wise H-E regression), and FSA (Feature Screening Approach)

2.1 Haseman-Elston Regression

Haseman-Elston regression (H-E) [4, 6] is the conventional non-parametric linkage analysis method to detect the QTLs based on the genetic IBD information between siblings/twins. This study uses an updated H-E regression method [5], and implements it in the package Stata (StataCorp, version 10).

H-E regression involves an evaluation of change in phenotypic intra-class correlation between the phenotype values of the first sibling X, and the second sibling, Y, given a change in genetic covariance, which is equivalently observed as the proportion of alleles shared IBD, π [6]. Given the population intra-class correlation, *r*, and the proportion of phenotypic variance explained by a QTL, Q, a linear relationship between phenotypic covariance and genetic covariance exists [7] (Var[*S*] = Var[*X*] = Var[*Y*]):

$$E[(X-Y)^2] = 2Var[S]\left(1-\left(r+Q(\pi-E[\pi])\right)\right)$$
$$= 2Var[S](1-r) - 2Var[S]Q(\pi-E[\pi]) \qquad (1)$$

The response of $(X\text{-}Y)^2$ can be used to estimate Q. An estimate of Q and assessments of its significance can thus be generated by maximizing likelihood. In linkage analysis, Logarithm Of Odds (LOD) score is usually used to indicate the significance of the linkage of a locus. LOD score is equivalent to significance of Generalized Linear Model (GLM) regression test statistic $\frac{\hat{Q}}{\sqrt{Var(\hat{Q})}}$.

2.2 New Approaches Considering Interaction

We proposed two other approaches, PWH-E and FSA, to take gene-gene interactions into account. These approaches consider not only the individual main genetic effect of a single locus, but also the influence from the interaction between loci. Both the approaches include the feature selection to pre-select loci for further analysis.

1) Feature Selection

The feature selection method devised in this research consists of a searcher to generate feature subsets from the original data and an evaluator to evaluate the strength of the relationship between the feature subset and the target phenotype. Correlation-based Feature Subset Selection (CfsSelection) [8] based on the Subset Forward search (SFS) method [9] is used in this study.

The Subset Forward Search (SFS) method introduced in [9] is a subset search strategy. The process of the SFS includes three major steps: (1) to produce a ranking by using an evaluator (i.e CfsSelection); (2) to carry out an interior k-fold cross-validation. A Linear Forward Selection on each fold to determine the optimal feature subset-size by using the given evaluator; (3) to apply the Linear Forward Selection with the optimal subset size on the whole data.

CfsSelection evaluates the worthiness of a subset of genetic features by considering the individual predictive ability of each locus along with the degree of redundancy between them. Subsets of genetic features that are highly correlated with the target phenotype while having low inter-correlation are preferred.

After completing the selection, the selected loci will be presented to the two approaches to detect the significance of the main genetic effect of individual loci and the interactive genetic effect from all selected loci pairs.

2) Pair-wise H-E regression (PWH-E)

The PWH-E is designed based on H-E regression, and its functional diagram is shown in Fig. 2. The difference between this method and the conventional H-E regression is that it includes the stage of loci pairing before running H-E regression. All possible loci pairs are generated by multiplying the IBD values of two loci to represent the possible statistical interaction effect. Both individual loci and loci pairs are tested by H-E regression to calculate LOD scores which indicate the significance of linkage.

This approach is proposed to detect both main and interactional effects by taking advantage of the strength of H-E regression.

3) Feature Screening Approach (FSA)

The proposed FSA (Fig. 2) uses a feature screening based on an artificial neural network (ANN). ANN is used for its high flexibility in modelling non-linear functions between input and output variables and its pattern recognition capability. The proposed procedure relies on the predictive importance of loci (or loci pairs) on the phenotypic outcome and does not explicitly incorporate any specific type of gene-gene interaction. It is based on the ANN input sensitivity analysis using the input screening technique [10, 11]. It compares the error made by the network with the original pattern to the error made when the selected inputs are blocked for all patterns. The greater increases in the error corresponds to the greater importance of a tested input [12].

In the process of this approach (FSA in Fig. 2), all the pre-selected genetic features are firstly presented to the modelling algorithm to generate a model to predict the target phenotypic value. The mean absolute error (MAE) between the predicted value and the true phenotypic value is calculated to be the performance benchmark (MAE_a).

Then a locus or loci pair is blocked from the modelling. ANN uses the remaining genetic features to predict the target phenotype. The performance of the modelling is evaluated by MAE (when the ith genetic feature is removed, then the output is recorded as MAE_i). The change of the MAE is calculated by equation (2).

$$\Delta MAE_i = MAE_i - MAE_a \tag{2}$$

ΔMAE_i is recorded as the score for the blocked locus or loci pair. A larger ΔMAE_i indicates the greater importance of the locus or loci pair. Each of loci and loci pairs is tested by the above process, until all of them have been tested, respectively.

As presented in Fig. 2., all individual loci and loci pairs are then ranked based on the generated ΔMAE. The top ranked loci or loci pairs suggest the potential QTLs of the target phenotype. The selected individual loci suggest they have strong main genetic effects, and the selected loci pairs reflect the strong genetic effect from the involved loci including the possible interaction effects between loci. This approach thus considers not only main effects of individual loci but also interaction effects between loci, without incorporating explicitly any specific type of gene-gene interaction.

3 Simulation Studies

3.1 Simulation Designs

Simulation studies are designed to test the performance of the conventional and proposed new approaches, before applying them to real-world BMD data from the twins. The simulation is based on real-world genotypic data. Quantitative phenotypic data are simulated according to different genetic models. The genotypic information of IBD alleles was measured at 756 microsatellite markers across 23 chromosomes of twins participating in the TwinsUK Registry. The values are the estimated proportion of identity by descent (IBD) alleles at each genetic marker of twin members. The number of IBD alleles is the number of identical copies of the same ancestral alleles at a marker locus that are shared by the members in a twin pair. It is the most commonly used measure of concordance of two individuals at a marker locus. Twin pairs sharing more alleles identical by descent (IBD) tend to have greater similarity in their phenotypes than pairs that share less [13].

A simulated phenotypic variable *P* is generated based on the values of genetic effects and the individual environmental effect. The values of *P* are then appended to the genetic IBD data to form a complete dataset for further simulation tests. The model for simulating phenotype *P*, which is assumed to be influenced by two QTLs, is given below.

$$P = f(v_x, v_y) = aHv_x + bHv_y + cHv_xv_y + dE$$

$$\text{(Subject to: } H + d = 1 \text{ and } a + b + c = 1) \qquad (3)$$

where v_x, v_y are the genetic variants at QTLs *X* and *Y*, their values are different for twin members in a twin pair according to genetic IBD values. a and b are the proportions of the heritability (*H*) for main genetic effects of *X* and *Y* respectively; c the proportion of the heritability (*H*) for genetic interaction effect between *X* and *Y*; *E* is the random individual environmental variant, and *d* is the coefficient of environmental effect *E*.

```
/*For a given genetic dataset of n pairs of twins, produce the simulated phenotypic data for twins*/
1. Set the parameters a, b, c, H (subject to the conditions for Equation (3));
2. Generate random genetic variants ($v_x$ and $v_y$) for twin 1 and twin2;
3. Generate random variants ($r_1$ and $r_2$) for unshared genetic variants between twin 1 and twin 2;
4. Generate genetic interaction variation ($z_1$ and $z_2$) between QTL X and QTL Y for twin1 and twin 2, using the corresponding shared IBD values x and y from the genetic data set.
        $z_1 = v_xv_y$
        $z_2 = (xv_x + (1-x)r_1)(yv_y + (1-y)r_2)$
5. Generate random variants ($e_1$ and $e_2$) for environmental variants of twin1 and twin2;
6. Generate phenotypic values for twin 1 and twin2 ($P_1$ and $P_2$) respectively.
        $P_1 = aHv_x + bHv_y + cHz_1 + (1-H)e_1$
        $P_2 = aH(xv_x + (1-x)r_1) + bH(yv_y + (1-y)r_2) + cHz_2 + (1-H)e_2$
7. Append the phenotypic values $P_1$ and $P_2$ to the original twins' genetic data;
8. Repeat steps from 2 to 7 for n times.
```

Fig. 3. Process of simulating phenotypic values

The process for generating the simulated phenotypic values is presented in Fig. 3., where x and y represent the proportions of shared IBD alleles at QTL X and Y, respectively; r_1 and r_2 are simulated random variation for unshared genetic variation between twin 1 and twin2; e_1 and e_2 are simulated random environmental variations for twin1 and twin 2.

Five simulations were designed (D1-D5) with different combinations of values for coefficients *a, b* and *c*, as shown in Table 1. To generate a dataset, a simulated phenotypic variable is generated according to the process of phenotypic data generation (Fig. 3.) based on one of the simulation designs (Table 1) and then appended to the genetic IBD data.

To test the performance consistency of these 3 methods, thirty datasets are generated for each of these five simulation designs (D1 - D5) and each of 3 particular heritability values (0.6, 0.8 or 1.0). The average power of each method for detecting both QTLs successfully is calculated.

Table 1. Five simulated data designs (D1-D5) considering main and interaction genetic effects from two QTLs (X and Y). *a*, *b* and *c* indicate the proportions of the heritability for main effects (of X and Y), and interaction effects (between X and Y), respectively.

D	***a***	***b***	***C***	**Description**
D	0%	0%	100	Only interaction effect
D	10	10	80	Strong interaction effect with weaker main
D	33.3	33.3	33.3	Equal main and interaction effects
D	40	40	20	Weaker interaction effect with stronger main
D	50	50	0%	Only main effects

3.2 Performance Evaluation and Comparison

H-E, PWH-E and FSA are applied to the five simulated designs. Thirty datasets are generated based on each design, and the power (i.e. the ratio of detecting both hypothetical QTLs successfully) of all three approaches is calculated and presented in Fig. 4.

According to Fig. 4, the results of simulations and evaluation, therefore, are clear enough to suggest that the performance of PWH-E is the best of the three approaches. For detecting both QTLs, PWH-E performs better than the H-E regression method especially in cases where only interaction effect or a strong interaction effect with weaker main effect dominates the genetic influence. For instance, the power of PWH-E is higher (p-value = 0.03, using paired t-test) compared to H-E regression in D1 with different heritability. When the heritability is fixed at 0.6, the power of PWH-E is also higher (p-value = 0.005) considering all five situations (D1-D5). The power of the FSA is calculated based on the ratio of QTLs appearing in the top 10 in the experiments. Fig. 4 shows the performance of FSA does not change much in the five simulation designs. For instance, the power is in the range [0.77-0.83] when heritability equals 1.0. Though its performance does not increase significantly with the increasing main genetic effect, and is slightly worse than the conventional H-E especially at D5 (only main effects), its performance is better than H-E at D1 and D2 (i.e., when interaction effect between two QTLs dominating the genetic influences in the phenotype). The average power of H-E is 0.38 at D1, while using FSA can achieves 0.68. FSA is not designed for considering any specific type of gene-gene interaction, but can still cope with multiplicative interactions. It means that FSA can be extended to reflect the genetic influence from other unknown types of interaction which can be validated in the further work.

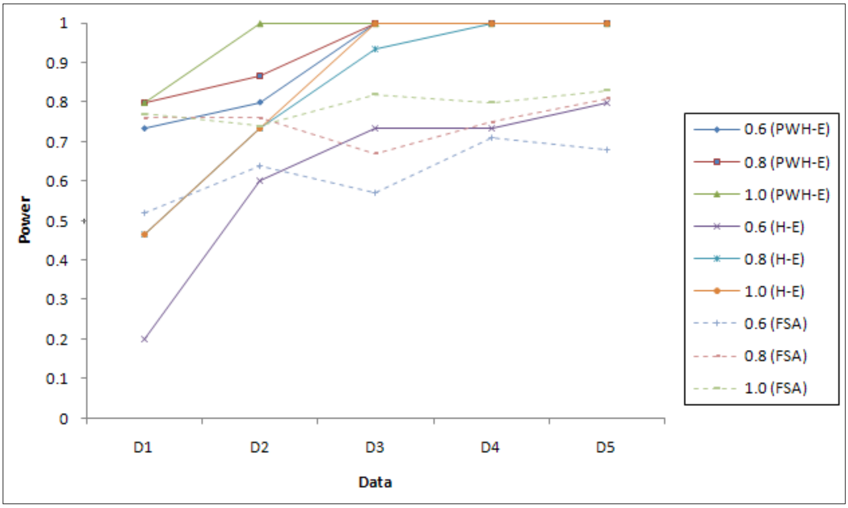


Fig. 4. The performance comparison between FSA, H-E and PWH-E. Different heritability values (H = 0.6, 0.8 or 1.0) are set to simulate the phenotype. The performance of H-E and PWH-E are based on the threshold of suggestive linkage (LOD>2).

4 Application to Real-World Bone Mineral Density Data

Bone mineral density (BMD) is a measure that indicates the amount of calcium per square centimetre of bones. It is a heritable complex phenotype often used to diagnose osteoporosis and assess the fracture risk. The conclusions generated by previous studies on BMD including genome-wide linkage and association studies are often inconsistent [14, 15]. Further genome-wide studies on BMD are required to produce more convincing evidences and conclusions to uncover the locations of genes regulating BMD phenotypes.

Data used in this research were collected by the Department of Twin Research (DTR) of King’s College, London. It contains both phenotypic BMD values and genetic values of DZ (non-identical) twins. The BMD information (Table 2) was measured by DEXA at three body sites (spine, hip, and forearm) from DZ female twin pairs.

In order to verify our findings, the outputs of PWH-E, and FSA are compared with approximate chromosomal locations suggested by six previous other studies

([14, 16-20]) on BMD. The consistent outputs including 14 chromosomal locations are presented in Table 3. Among them, six chromosomal locations are suggested by both PWH-E and FSA, and at least one previous study. They are in chromosomes 2, 3, 6, 12, 13 and 23 respectively.

The chromosomal location at 0-65.94 cM in chromosome 13 has the strongest evidence for containing the true QTLs of BMD. This is suggested by the results from all 3 methods in this study and also in 3 previous studies ([16], [18], and [14]). The chromosomal location at 50.7-85.36 cM in chromosome 3 also has strong evidence of containing QTLs suggested by the results from PWH-E and FSA in this study and 2 previous studies ([16] and [20]).

The good consistency of our results with the published research is a clear evidence to indicate that our methods are capable of detecting the true QTLs of BMD. Thus it

Table 2. Statistical summary of the square difference (SD) values of BMD in twin pairs

Variables	Min	Max	Mean	Std. Dev	Valid records
Spine (SD)	0	0.314	0.026	0.043	363
Forearm (SD)	0	0.034	0.003	0.004	338
Hip (SD)	0	0.194	0.018	0.028	363

Table 3. Findings of conventional H-E, PWH-E and FSA confirmed by previous studies, Chromosomal locations suggested by both PWH-E and FSA, and at least one previous study are highlighted

Chromosome	Position (cM)	Previous studies	H-E	PWH-E	FSA
1	0-125.39	[16],[18]			Hip, Forearm
2	82.95-93.56	[16]		Spine	Spine
	127.4-135	[19]			Hip
3	50.7-85.36	[16],[20]		Hip	Hip
4	83-115	[20]			Hip
6	55.38-89.66	[18]		Forearm	Forearm
9	69	[20]		Hip	
11	73.2-81.22	[16]			Hip
12	120-213.84	[14]		Forearm	Forearm
13	0-65.94	[16],[18],[14]	Spine	Spine, Hip, Forearm	Spine, Forearm
	110	[20]		Forearm	
14	112.47-118.70	[17]		Forearm	
17	73.1-81.92	[16]		Spine	
23	140-167.84	[19]		Hip	Hip, Spine

is reasonable to infer that 22 chromosomal locations (listed in Table 4), which are identified by our methods but not any of the known previous studies, are more likely new findings. The locations at 225cM in chromosome 1 and 50cM in chromosome 4 are suggested containing QTLs of BMD at hip by both PWH-E and FSA. 10cM in chromosome 10, 85-90cM in chromosome 13, and 0cM in chromosome 23 are suggested influencing BMD at spine by both of our methods. The outputs of both PWH-E and FSA also suggest 0-5cM in chromosome 13 determining BMD at forearm.

Table 4. New findings suggested by PWH-E and FSA (22 chromosomal locations are involved)

Chromosome	Position (cM)	PWH-E	FSA
1	225	Hip	Hip
	270		Spine
2	145		Forearm
	45	Forearm	
3	205-215	Hip	
	150		Spine
4	50	Hip	Hip
6	130		Hip
	160		Forearm
7	0		Spine
8	110	Forearm	Hip
9	30		Forearm
10	10	Spine	Spine
12	15		Spine
13	85-90	Spine	Spine
	0-5	Forearm	Forearm
15	50		Forearm
16	70		Spine
19	85		Spine
20	0		Forearm
22	55		Hip
23	0	Spine	Spine

5 Summary

Haseman-Elston (H-E) regression is a conventional approach for detecting QTLs that regulate the target quantitative phenotype based on IBD information between twins. However, this approach only considers the genetic effect from individual loci, but not gene-gene interaction. We proposed two new approaches: PWH-E and FSA, to detect the QTLs allowing interaction influence between different loci in genome-wide linage analysis.

A series of simulation studies were carried out to test the abilities of the new approaches as well as the H-E method, to uncover QTLs, particularly the QTLs that have interaction effects on the phenotype. The result showed that, when significant main genetic effects regulate the phenotype, both the conventional H-E regression and the PWH-E method can detect QTLs with substantial power (which can reach 100% power when the heritability of the phenotype is larger than or equals 0.8). When gene-gene interactions dominate the genetic influences on the phenotype, both new proposed methods PWH-E and FSA, performed better than the H-E. The performance of PWH-E is also better than FSA. The reasons may include that PWH-E was designed to consider multiplicative interaction relationships, and the gene-gene interaction was represented by multiplicative relationship in the simulation. FSA using ANN is not designed for considering any specific type of gene-gene interaction, but it can still cope with the multiplicative interaction. It can be extended to be used in more general situations to detect a genetic influence from other unknown types of interaction relationships.

The new methods are then applied for a genome-wide linkage analysis on a real world dataset relating to bone mineral density at the spine, hip and forearm. Their results are compared with the findings of six previous studies on BMD. Apart from confirming that the previous findings, including 6 chromosomal locations (in chromosomes 2, 3, 6, 12, 13 and 23) that have been implicated as having strong influences on BMD, and further highlighting the importance of chromosome 3 cM 50.7-85.36 and chromosome 13 cM 0-65.94, our methods also produced strong evidences to suggest 22 chromosomal locations as new findings of QTLs associated with BMD. Of course, the true influence of these QTLs on BMD needs to be studied in further genetic epidemiological and biological research and this research provides a basis for doing that.

References

1. Carrasquillo, M.M., et al.: Genome-wide association study and mouse model identify interaction between TET and EDNRB pathways in Hirschsprung disease. Nat. Genet. 32, 237–244 (2002)
2. Motsinger, A.A., Ritchie, M.D., Reif, D.M.: Novel methods for detecting epistasis in pharmacogenomics studies. Pharmacogenomics 8(9), 1229–1241 (2007)
3. Musani, S.K., et al.: Detection of gene x gene interactions in genome-wide association studies of human population data. Human Heredity 63(2), 67–84 (2007)
4. Elston, R.C., et al.: Haseman and Elston revisited. Genetic Epidemiology 19(1), 1–17 (2000)
5. Barber, M.J., et al.: Gamma regression improves Haseman-Elston and variance components linkage analysis for sib-pairs. Genetic Epidemiology 26(2), 97–107 (2004)
6. Haseman, J.K., Elston, R.C.: The investigation of linkage between a quantitative trait and a marker locus. Behavior Genetics 2(1), 3–19 (1972)
7. Sham, P.C., et al.: Powerful regression-based quantitative-trait linkage analysis of general pedigrees. The American Journal of Human Genetics 71(2), 238–253 (2002)

8. Hall, M.A.: Correlation-based feature selection of discrete and numeric class machine learning (2000)
9. Gütlein, M., et al.: Large-scale attribute selection using wrappers (2009)
10. Masters, T.: Practical neural network recipes in C++. Morgan Kaufmann, San Francisco (1993)
11. Montano, J.J., Palmer, A.: Numeric sensitivity analysis applied to feedforward neural networks. Neural Computing & Applications 12(2), 119–125 (2003)
12. Matchenko-Shimko, N., Dube, M.P.: Gene-Gene Interaction Tests Using SVM and Neural Network Modeling. In: CIBCB (2007)
13. Thomas, D.C.: Statistical Methods in Genetic Epidemiology. Oxford University Press, Oxford (2004)
14. Kaufman, J.M., et al.: Genome-wide linkage screen of bone mineral density (BMD) in European pedigrees ascertained through a male relative with low BMD values: evidence for quantitative trait loci on 17q21-23, 11q12-13, 13q12-14, and 22q11. Journal of Clinical Endocrinology & Metabolism 93(10), 3755 (2008)
15. Richards, J.B., et al.: Bone mineral density, osteoporosis, and osteoporotic fractures: a genome-wide association study. The Lancet (2008)
16. Rivadeneira, F., et al.: Twenty loci associated with bone mineral density identified by large-scale meta-analysis of genome-wide association datasets. Bone 44, 230–231 (2009)
17. Koller, D.L., et al.: Genome-Wide Association Study of Bone Mineral Density in Premenopausal European-American Women and Replication in African-American Women. Journal of Clinical Endocrinology & Metabolism
18. Styrkarsdottir, U., et al.: Multiple genetic loci for bone mineral density and fractures. New England Journal of Medicine 358(22), 2355 (2008)
19. Zhang, F., et al.: A whole genome linkage scan for QTLs underlying peak bone mineral density. Osteoporosis International 19(3), 303–310 (2008)
20. Wilson, S.G., et al.: Comparison of genome screens for two independent cohorts provides replication of suggestive linkage of bone mineral density to 3p21 and 1p36. The American Journal of Human Genetics 72(1), 144–155 (2003)

Author Index