ADVANCES IN INTELLIGENT SYSTEMS AND COMPUTING 204

Pedro Latorre Carmona
J. Salvador Sánchez
Ana L.N. Fred (Eds.)

Pattern Recognition – Applications and Methods

Springer

Advances in Intelligent Systems and Computing 204

Editor-in-Chief

For further volumes:
http://www.springer.com/series/11156

Pedro Latorre Carmona, J. Salvador Sánchez,
and Ana L.N. Fred (Eds.)

Pattern Recognition – Applications and Methods

International Conference, ICPRAM 2012
Vilamoura, Algarve, Portugal,
February 6–8, 2012
Revised Selected Papers

Springer

Editors
Pedro Latorre Carmona
Department of Computer Languages and Systems
Jaume I University
Castellon de la Plana
Spain

J. Salvador Sánchez
Department of Programming Languages and Information Systems
Jaume I University
Castellon de la Plana
Spain

Ana L.N. Fred
IST - Technical University of Lisbon
Lisbon
Portugal

ISSN 2194-5357
ISBN 978-3-642-36529-4
DOI 10.1007/978-3-642-36530-0
Springer Heidelberg New York Dordrecht London

e-ISSN 2194-5365
e-ISBN 978-3-642-36530-0

Library of Congress Control Number: 2013931192

Printed on acid-free paper

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

Preface

The present book includes extended and revised versions of a set of selected papers from the First International Conference on Pattern Recognition (ICPRAM 2012), held in Vilamoura, Algarve, Portugal, from 6 to 8 February, 2012, sponsored by the Institute for Systems and Technologies of Information Control and Communication (INSTICC) and held in cooperation with the Association for the Advancement of Artificial Intelligence (AAAI) and Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL2). This conference was technically co-sponsored by IEEE Signal Processing Society, Machine Learning for Signal Processing (MLSP) Technical Committee of IEEE, AERFAI (Asociación Española de Reconocimiento de Formas y Análisis de Imagen) and APRP (Associação Portuguesa de Reconhecimento de Padrões). INSTICC is member of the Workflow Management Coalition (WfMC).

The purpose of the International Conference on Pattern Recognition Applications and Methods (ICPRAM) is to bring together researchers, engineers and practitioners interested on the areas of Pattern Recognition, both from theoretical and application perspectives.

ICPRAM received 259 paper submissions from 46 countries, in all continents. To evaluate each submission, a double blind paper review was performed by the Program Committee, whose members are highly qualified researchers in ICPRAM topic areas. Based on the classifications provided, only 115 papers were selected for oral presentation (61 full papers and 54 short papers) and 32 papers were selected for poster presentation. The full paper acceptance ratio was 24%, and the total oral acceptance ratio (including full papers and short papers) 44%. These strict acceptance ratios show the intention to preserve a high quality forum which we expect to develop further next year.

We must thank the authors, whose research and development efforts are recorded here. We also thank the keynote speakers for their invaluable contribution and for taking the time to synthesise and prepare their talks. Finally, special thanks to all the members of the INSTICC team, whose collaboration was fundamental for the success of this conference.

December 2012

Pedro Latorre Carmona
J. Salvador Sánchez
Ana L.N. Fred

Organization

Conference Chair

Ana Fred	Technical University of Lisbon / IT, Portugal

Program Co-chairs

J. Salvador Sánchez	Jaume I University, Spain
Pedro Latorre Carmona	Jaume I University, Spain

Organizing Committee

Patrícia Alves	INSTICC, Portugal
Sérgio Brissos	INSTICC, Portugal
Helder Coelhas	INSTICC, Portugal
Patrícia Duarte	INSTICC, Portugal
Liliana Medina	INSTICC, Portugal
Carla Mota	INSTICC, Portugal
Vitor Pedrosa	INSTICC, Portugal
Daniel Pereira	INSTICC, Portugal
Cláudia Pinto	INSTICC, Portugal
José Varela	INSTICC, Portugal
Pedro Varela	INSTICC, Portugal

Program Committee

Shigeo Abe, Japan
Felix Agakov, U.K.
Gady Agam, U.S.A.
Mayer Aladjem, Israel
Rocío Alaiz-Rodríguez, Spain
Guillem Alenya, Spain
Genevera Allen, U.S.A.
Ethem Alpaydin, Turkey
Jesús Angulo, France
Annalisa Appice, Italy
Cedric Archambeau, France
Antonio Artés-Rodríguez, Spain
Thierry Artières, France
Karl Aström, Sweden

Bing Bai, U.S.A.
Kevin Bailly, France
Vineeth Nallure Balasubramanian, U.S.A.
Luis Baumela, Spain
Jon Atli Benediktsson, Iceland
Charles Bergeron, U.S.A.
André Bergholz, Germany
J. Ross Beveridge, U.S.A.
Monica Bianchini, Italy
Concha Bielza, Spain
Isabelle Bloch, France
Dorothea Blostein, Canada
Anselm Blumer, U.S.A.
Liefeng Bo, U.S.A.
Joan Martí Bonmatí, Spain
Gianluca Bontempi, Belgium
Henrik Boström, Sweden
Patrick Bouthemy, France
Francesca Bovolo, Italy
Ulf Brefeld, Spain
Alexia Briassouli, Greece
Paula Brito, Portugal
Lorenzo Bruzzone, Italy
Hans du Buf, Portugal
Tiberio Caetano, Australia
Javier Calpe, Spain
Gustavo Camps-Valls, Spain
Ramón A. Mollineda Cárdenas, Spain
Pedro Latorre Carmona, Spain
Xavier Carreras, Spain
Marco La Cascia, Italy
Rui M. Castro, The Netherlands
Zehra Cataltepe, Turkey
Nicolas Cebron, Germany
Jocelyn Chanussot, France
Snigdhansu Chatterjee, U.S.A.
Frederic Chazal, France
Chi Hau Chen, U.S.A.
Seungjin Choi, Korea, Republic of
Jesús Cid-Sueiro, Spain
Juan Manuel Corchado, Spain
Antoine Cornuejols, France
Tom Croonenborghs, Belgium
Jesús Manuel de la Cruz, Spain
Justin Dauwels, Singapore

Marc Deisenroth, U.S.A.
Jeroen Deknijf, Belgium
José Bioucas Dias, Portugal
Thorsten Dickhaus, Germany
Carlos Diuk, U.S.A.
Kurt Driessens, The Netherlands
Petros Drineas, U.S.A.
Gideon Dror, Israel
Artur Dubrawski, U.S.A.
Delbert Dueck, U.S.A.
Carl Henrik Ek, Sweden
Tapio Elomaa, Finland
Francesc J. Ferri, Spain
Mario Figueiredo, Portugal
Maurizio Filippone, U.K.
Gernot A. Fink, Germany
Vojtech Franc, Czech Republic
Damien François, Belgium
Ana Fred, Portugal
Élisa Fromont, France
Sabrina Gaito, Italy
Vicente Garcia, Spain
Paulo Gotardo, U.S.A.
Giuliano Grossi, Italy
Sébastien Guérif, France
José J. Guerrero, Spain
Adolfo Guzman, Mexico
Jose Antonio Martin H., Spain
Amaury Habrard, France
Benjamin Haibe-Kains, U.S.A.
Peter Hall, Australia
Barbara Hammer, Germany
Onur Hamsici, U.S.A.
Edwin Hancock, U.K.
Mark Hasegawa-Johnson, U.S.A.
Pablo Hennings-Yeomans, U.S.A.
Francisco Herrera, Spain
Tom Heskes, The Netherlands
Laurent Heutte, France
Colin de la Higuera, France
Kouichi Hirata, Japan
Tin Kam Ho, U.S.A.
Susan Holmes, U.S.A.
Antti Honkela, Finland
Eduardo Raul Hruschka, Brazil

Bao-Gang Hu, China
Bert Huang, U.S.A
Su-Yun Huang, Taiwan
Eyke Hüllermeier, Germany
Nazli Ikizler-Cinbis, Turkey
Jose M. Iñesta, Spain
Akihiro Inokuchi, Japan
Alan J. Izenman, U.S.A.
Saketha Nath Jagarlapudi, India
Robert Jenssen, Norway
Alfons Juan, Spain
Felix Jungermann, Germany
Ata Kaban, U.K.
Yasushi Kanazawa, Japan
Kristian Kersting, Germany
Yunho Kim, U.S.A.
Koichi Kise, Japan
Walter Kosters, The Netherlands
Constantine Kotropoulos, Greece
Irene Kotsia, U.K
Sotiris Kotsiantis, Greece
Konstantinos Koutroumbas, Greece
Oleksiy Koval, Switzerland
Felix Krahmer, Germany
Eyal Krupka, Israel
Adam Krzyzak, Canada
Gautam Kunapuli, U.S.A.
Nojun Kwak, Korea, Republic of
Jaerock Kwon, U.S.A.
Michail Lagoudakis, Greece
Terran Lane, U.S.A.
Helge Langseth, Norway
Pedro Larrañaga, Spain
Nikolaos Laskaris, Greece
Mustapha Lebbah, France
Yoonkyung Lee, U.S.A.
Christian Leistner, Switzerland
Reiner Lenz, Sweden
Ales Leonardis, Slovenia
Qi Li, U.S.A.
Wenhui Liao, U.S.A.
Rainer Lienhart, Germany
Aristidis Likas, Greece
Hantao Liu, The Netherlands
Huan Liu, U.S.A.
Xiaoming Liu, U.S.A.
Gaelle Loosli, France
Michael Mahoney, U.S.A.
Jesús Malo, Spain
Stephane Marchand-Maillet, Switzerland
Elena Marchiori, The Netherlands
Jesús Picó i Marco, Spain
Peter McCullagh, U.S.A.
Yoan Miche, Finland
Luisa Micó, Spain
David J. Miller, U.S.A.
Majid Mirmehdi, U.K.
Rafael Molina, Spain
Luis Montesano, Spain
Igor Mozetic, Slovenia
Klaus-Robert Müller, Germany
Arthur Munson, U.S.A.
Laurent Najman, France
Michele Nappi, Italy
Claire Nédellec, France
Zoran Nenadic, U.S.A.
Mark Nixon, U.K.
Guillaume Obozinski, France
Erkki Oja, Finland
Ronald Ortner, Austria
Martijn van Otterlo, The Netherlands
Vicente Palazón-González, Spain
Apostolos Papadopoulos, Greece
Emilio Parrado-Hernández, Spain
Hélène Paugam-Moisy, France
Marcello Pelillo, Italy
Pedro Albertos Pérez, Spain
Petra Perner, Germany
Gabriel Peyre, France
Frederick Kin Hing Phoa, Taiwan
Sergey Plis, U.S.A.
Doina Precup, Canada
Philippe Preux, France
Lionel Prevost, France
Peihua Qiu, U.S.A.
Petia Radeva, Spain
Subramanian Ramamoorthy, U.K.
Jan Ramon, Belgium
Nicola Rebagliati, Italy
Lev Reyzin, U.S.A.

Elisa Ricci, Italy
François Rioult, France
José Cristóbal Riquelme, Spain
David Masip Rodo, Spain
Juan J. Rodríguez, Spain
Lior Rokach, Israel
Rosa María Valdovinos Rosas, Mexico
Juho Rousu, Finland
Yvan Saeys, Belgium
Lorenza Saitta, Italy
Mathieu Salzmann, U.S.A.
Luciano Sanchez, Spain
J. Salvador Sánchez, Spain
Antonio-José Sánchez-Salmerón, Spain
Michele Scarpiniti, Italy
Tanya Schmah, Canada
Thomas Schoenemann, Sweden
Friedhelm Schwenker, Germany
Pedro Garcia Sevilla, Spain
Katsunari Shibata, Japan
Yoel Shkolnisky, Israel
José Martínez Sotoca, Spain
Vassilios Stathopolous, U.K.
Stefan Steidl, Germany
Masashi Sugiyama, Japan
Shiliang Sun, China
Yajie Sun, U.S.A.
Johan Suykens, Belgium
Ichiro Takeuchi, Japan
Ameet Talwalkar, U.S.A.
Toru Tamaki, Japan
Lijun Tang, U.S.A.
Yuliya Tarabalka, U.S.A.
Nikolaj Tatti, Belgium
Bo Thiesson, U.S.A.
Michalis K. Titsias, U.K.
Michael Titterington, U.K.
Ryota Tomioka, Japan
Fabien Torre, France
Fernando De La Torre, U.S.A.
Andrea Torsello, Italy
Godfried Toussaint, U.S.A.
V. Javier Traver, Spain
Ivor Wai-Hung TSANG, Singapore
Gerhard Tutz, Germany
Eugene Tuv, U.S.A.
Raquel Urtasun, U.S.A.
Giorgio Valentini, Italy
Antanas Verikas, Sweden
Michel Verleysen, Belgium
Alessandro Verri, Italy
Cinzia Viroli, Italy
Jordi Vitrià, Spain
Sviatoslav Voloshynovskiy, Switzerland
Jilles Vreeken, Belgium
Thomas Walsh, U.S.A.
Joost van de Weijer, Spain
Kilian Weinberger, U.S.A.
Pierre Armand Weiss, France
David Windridge, U.K.
Ole Winther, Denmark
Stephen Wright, U.S.A.
Jianxin Wu, Singapore
Xin-Shun Xu, China
Filip Zelezny, Czech Republic
Josiane Zerubia, France
Arthur Zimek, Germany
Albrecht Zimmermann, Belgium
Indre Žliobaite, The Netherlands
Reyer Zwiggelaar, U.K.

Auxiliary Reviewers

Alessandro Adamo, Italy
Michael AUPETIT, France
Daniel Bartz, Germany
Guoqing Chao, China
Pierre Charbonnier, France
John Chiverton, Thailand
Thiago Covoes, Brazil
Kris Cuppens, Belgium
Glen Debard, Belgium
Jose Augusto Andrade Filho, U.S.A.
Qingbin Gao, China
Jorge Garcia, Spain
Patrick Heas, France
Evan Herbst, U.S.A.

Ernesto Di Iorio, Italy
Prashanth Marpu, Iceland
Daniel Mateos, Spain
Stefano Melacci, Italy
Luis Puig Morales, Spain
Jan Richarz, Germany
Alejandro Rituerto, Spain
Szilard Vadja, Germany
Jing Zhao, China

Invited Speakers

Tiberio Caetano	NICTA, Australia
Francis Bach	INRIA, France
José C. Príncipe	University of Florida, U.S.A.
JoachimM. Buhmann	ETH Zurich, Switzerland
Kostas Triantafyllopoulos	University of Sheffield, U.K.

Contents

Pattern Recognition as an Indicator of Diagnostic Expertise

Thomas Loveday[1], Mark Wiggins[1], Marino Festa[2], David Schell[2], and Dan Twigg[3]

[1] Department of Psychology, Macquarie University, Australia
{thomas.loveday,mark.wiggins}@mq.edu.au
[2] Childrens Hospital at Westmead, Sydney, Australia
{marino.festa,david.schell}@health.nsw.gov.au
[3] Transpower, Wellington, New Zealand
dan.twigg@transpower.co.nz

Abstract. Expertise is typically associated with high levels of experience in a domain. However, high levels of experience do not necessarily mean that operators are capable of performing at the level of expertise. Based on evidence that pattern-recognition is the foundation of expert diagnostic performance, two studies investigated the utility of distinguishing competent from expert practitioners using measures of the component tasks of pattern-recognition. In two dissimilar domains, performance across the tasks clustered into two levels, reflecting competence and expertise. Performance on the tasks was only weakly correlated with years of experience in the domain. The significance of these results is discussed in relation to assessment and training evaluation.

Keywords: Cues, Expertise, Pattern-recognition.

1 Background

The expertise of diagnosticians has typically been associated with cumulative experience in the domain. However, it is apparent that some experienced and qualified practitioners may never achieve genuine expertise, and instead, will only achieve a level of performance that could be described as competent [1, 2].

To explain this observation, Gray [3] proposed that amongst highly experienced individuals, there are actually two levels of performance. The levels were presumed to reflect ‘competent non-experts’, who rely on prior cases and heuristics [4], and ‘genuine experts’, who utilize reliable and efficient cognitive shortcuts [5]. Specifically, it is evident that experts, who have been identified on the basis of their diagnostic performance, are more likely to using pattern recognition in comparison to their non-experts peers [6-8].

Pattern recognition is defined as the non-conscious recognition of problem-states based on patterns of features that prime appropriate scripts in memory [9, 10]. The efficiency of expert pattern-recognition appears to be based on highly nuanced and automated feature-outcome associations in memory [6, 11]. These ‘cue’ associations represent an association in memory between the features of the environment and a subsequent outcome or problem [12]. Cue-based pattern recognition reduces cognitive load during information acquisition without sacrificing depth of processing [13],

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 1–11.
DOI: 10.1007/978-3-642-36530-0_1

thereby allowing experts to generate rapid and appropriate responses to environmental stimuli [14]. For example, in a ‘think-aloud’ study of gastroenterologists, it was observed that pattern-recognition during diagnosis produced accurate, and seemingly automatic, treatment responses [6].

Since expert diagnostic performance invokes pattern-recognition, several researchers have sought to predict diagnostic performance using simple tasks that assess the ability of operators to recognize the relationships between features in the environment and subsequent events. For example, Morrison, Wiggins, Bond and Tyler [15] developed a paired association task to assess diagnostic expertise using feature and event pairs, and found that expert offender profilers’ recorded faster recognition of associated features and events in comparison to their novice counterparts. However, that study was limited insofar as it only measured one component skill of cue-based pattern recognition.

The present researchers proposed a more comprehensive assessment process, whereby experts are distinguished from non-experts using normative measures of four distinct component skills of cue-based pattern recognition. Specifically, a series of studies attempted to distinguish experts from non-experts within an experienced population based on their performance during diagnostic tasks in which the selection and extraction of appropriate cues was advantageous.

Two batteries of cue-based tasks were developed within the software package, EXPERTise, one of which was designed for pediatric intensive care and the other for electricity power-control. EXPERTise was specifically designed to identify expert practitioners in those domains by combining four diagnostic tasks in which performance is reliant on cue utilization:

- *Feature Identification*, which is a measure of the ability to extract diagnostic cues from the operational environment [16];
- *Paired Association*, which assesses the capacity to discern strong feature-event cues from weak feature event cues in the environment [15];
- *Feature discrimination*, which is a measure of the ability to discriminate diagnostic from irrelevant cues in the environment [17]; and the
- *Transition Task*, which assesses the capacity to acquire diagnostic cues from the environment in a strategic, non-linear pattern [18].

This chapter describes the two studies in detail. Study One describes the application of EXPERTise in pediatric intensive care and demonstrates its ability to distinguish expert from non-expert personnel. Study Two extends Study One by applying the tool to power-control, demonstrating the generalizability of cue-based measures. It also involved an assessment of the reliability of classifications of expertise based on cue utilization.

2 Study One

2.1 Aims and Hypothesis

Study One was designed to determine the utility of EXPERTise in distinguishing competent non-experts from genuine experts within an experienced sample of medical practitioners.

Since each of the four tasks used in the present study was selected to assess independent facets of the broader construct of cue-based pattern-recognition during diagnosis, it was hypothesized that performance amongst experienced practitioners would cluster into two levels across the tasks. This prediction was consistent with evidence that experienced practitioners can be distinguished as competent non-experts or experts, depending upon their performance. Since years of experience in the domain is only weakly associated with expert skill acquisition, performance during the tasks was not expected to correlate significantly with years of domain experience.

2.2 Method

Participants. A sample of fifty pediatric intensive care unit staff was recruited for the study. Twenty three were male and twenty seven were female. They ranged in age from 30 to 63 years with a mean of 42.3 years (SD = 8.3). The participants had accumulated between three and 26 years of experience within pediatric critical care, with a mean of 9.8 years (SD = 6.9).

Measures. The present study used two measures, a demographics survey and EXPERTise.

Demographic Survey. In addition to basic demographics, years of experience in the domain was recorded.

Expertise. EXPERTise is a 'shell' software package designed to record performance across four cue-based expert diagnostic reasoning tasks. EXPERTise was specifically designed so that these tasks could be customized to match stimuli used in the domain.

Stimuli. Cognitive interviews were conducted with two pediatric intensive care practitioners to develop the stimuli used in the present study. These practitioners were selected on the basis of peer recommendation. The information derived from the subject-matter experts was restructured into several scenarios that identified feature and outcome pairs that were available for patient diagnosis. These pairs and scenarios were validated in an untimed pilot test. The scenarios formed the basis of the stimuli used within the EXPERTise tasks. See Figure 1 for an example of the stimuli.

Feature Identification Task (FID). The feature identification task had two forms. In the first form, the participants were presented with a patient bedside monitor displaying an abnormal parameter that indicated that the patient was in a critical condition. The participants were asked to 'click' on the abnormal parameter. In the second form, the bedside monitor was displayed for 1.5 seconds, and the participant was asked to identify the abnormal parameter from one of four options. For both forms, response times were recorded and were aggregated across items to yield a mean response time. Accuracy was also recorded and summed for a single accuracy score.

Paired Association Task (PAT). The paired association task also had two forms. In both, two domain-relevant phrases were displayed on-screen, either sequentially (Form 1) or simultaneously (Form 2) for 1.5 seconds. The participant was asked to rate the relatedness of the two phrases on a six-point scale. Response latencies were recorded and aggregated across items to yield a mean reaction time for each participant. The association ratings were also aggregated into a single ‘discrimination’ metric based on the mean variance of the participants’ responses.

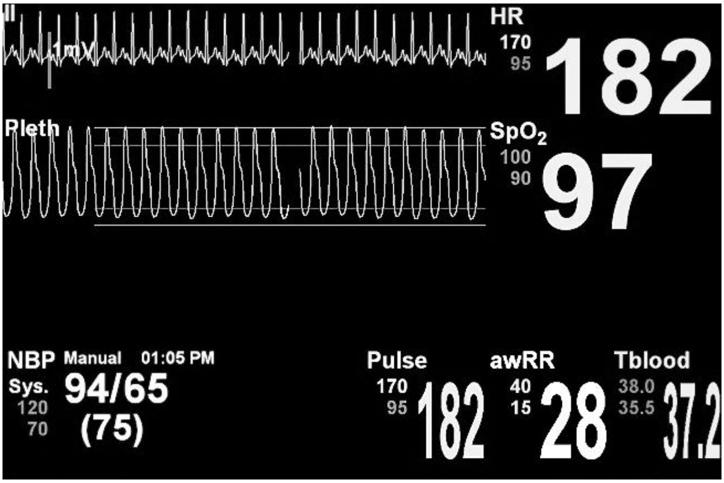


Fig. 1. Example patient bedside monitor output

Feature Discrimination Task (FDT). The feature discrimination task measured expert discrimination between sources of information during decision-making. The task presented the participant with a patient bedside monitor output and a short written scenario description. On a subsequent screen, the participants were asked to choose an appropriate response to the scenario from eight treatment options. The participants then rated, on a six-point scale, the utility of nine individual types of information in informing their decision. These ratings were aggregated into a single discrimination metric based on the variance of the participant’s ratings.

Transition Task (TT). The transition task consisted of a single scenario accompanied by a patient bedside monitor output. The scenario was intentionally vague and thus, forced participants to acquire additional information provided in a list of information screens. The participants then selected a diagnosis and response from four treatment options. The order in which the information screens were accessed was recorded. This was converted to a single metric based on the ratio of screens accessed in sequence to the total number of screens accessed.

Procedure. Participants were initially briefed on the purpose of the study and were then asked to sign a consent form if they wished to continue. They subsequently completed the demographics questionnaire and EXPERTise using a laptop computer.

2.3 Study 1 Results

Correlations with Experience. To investigate the relationship between years of experience in pediatrics and each task within EXPERTise, bivariate correlations were undertaken between years of experience in the domain and performance on the

EXPERTise tasks. Consistent with expectations, years of experience in the domain yielded only weak to moderate Pearson correlations with performance on the EXPERTise tasks, $0 \leq r \leq 0.33, p < 0.05$.

Cluster Models. The primary aim of the Study 1 was to determine the feasibility of identifying expert practitioners using tasks in which pattern recognition was advantageous. Because the sample comprised qualified individuals, it was expected that performance would cluster into two groups, reflecting competence and expertise.

Table 1 presents the results of a k-Means cluster analysis. As expected, two distinct groups formed based on performance across the EXPERTise tasks.

Cluster One (n = 24) comprised those individuals who, whilst qualified, demonstrated a lower level of performance across the EXPERTise tasks in comparison to the members of Cluster 2. Therefore, the participants in this cluster were described as 'competent non-experts'.

Cluster Two (n = 26) comprised those individuals who demonstrated superior performance across the EXPERTise tasks. Since the members of this cluster were generally faster, more accurate, more discriminating, and less sequential in their acquisition of information, they were described as 'experts.'

Table 1. Participant cluster means for Study 1

Measure	Non-Expert Mean (SD)	Expert Mean (SD)	Overall Mean (SD)
Feature Identification Reaction Time	11.1 (4.4)	7.7 (3.0)	9.2 (4.1)
Feature Identification Accuracy	5.3 (2.1)	6.7 (1.7)	6.0 (2.0)
Paired Association 1 Reaction Time	6.0 (2.2)	4.6 (1.3)	5.3 (1.9)
Paired Association 1 Variance	1.5 (0.6)	2.4 (0.8)	2.0 (0.8)
Paired Association 2 Reaction Time	4.3 (1.7)	3.6 (1.1)	3.9 (1.4)
Paired Association 2 Variance	1.2 (0.7)	1.8 (0.7)	1.5 (0.7)
Feature Discrimination Variance	2.72 (2.8)	4.5 (3.2)	3.7 (3.1)
Transition Ratio	0.91 (0.17)	0.63 (0.42)	0.8 (0.4)

2.4 Study 1 Discussion

The aim of Study One was to determine whether four measurements of pattern recognition could, when combined, distinguish competent from expert medical practitioners within an experienced sample. Since the judicious selection and extraction of cues was advantageous in each of the tasks, it was expected that pediatric experts would demonstrate consistently superior performance.

The results of Study One are consistent with expectations that the EXPERTise tasks could accurately distinguish competent from expert practitioners within an experienced sample of participants. Performance across the four assessment tasks clustered into two levels, with Cluster Two significantly outperforming Cluster One on all four tasks. This suggests that the Cluster One and Cluster Two represented, respectively, non-experts and experts within an experienced sample.

As expected, performance on the tasks was not strongly correlated with years of experience in the domain. This outcome is consistent with prior research [6-8], and thus, highlights the limitations of using years of experience as a means of identifying expert diagnosticians in pediatric healthcare. There is an increasingly strong case to be made that years of experience in the domain is only weakly associated with the progression to diagnostic expertise [3], suggesting that other indicators may be preferable.

3 Study Two

3.1 Aims and Hypothesis

Study One demonstrated that experts could be distinguished from non-experts within an experienced sample using measures of pattern-recognition. Study Two was designed to extend the outcomes of Study One by testing whether the effects observed are evident within other diagnostic roles. Power system control was selected as the domain of interest because this role involves forecasting network demands and diagnosing system events. Therefore, experienced operators would have had sufficient opportunity to acquire the feature and event associations in memory that facilitate pattern recognition and diagnostic expertise.

Consistent with the outcomes of Study One, it was hypothesized that performance amongst qualified power-controllers would cluster into two levels across the EXPERTise tasks. Performance during the tasks was not expected to correlate significantly with years of experience in the domain.

Study Two was also designed to enable an assessment of the reliability of pattern-recognition-based expert and non-expert classifications over time. Specifically, Study Two examined whether classifications of expertise, based on performance during the EXPERTise tasks, remained consistent over a six-month period.

Previous research suggests that the rate of skill acquisition differs, depending upon the level of experience within the domain. Specifically, the rate of skill acquisition in the progression from novice to competence tends to be much faster for a given period of time than the rate of skill acquisition in the progression from competence to expertise [19, 20]. Therefore, it was hypothesized that if EXPERTise produces a consistent and valid classification of expertise based on normative performance, then within an experienced sample a significant level of consistency in EXPERTise-based normative classifications should be observed between administrations of the battery at a six-month interval.

3.2 Method

Participants. Initially, twenty-one qualified power controllers, recruited from Transpower, New Zealand, elected to participate in the study. At retest, conducted six months later, five of the participants were not available due to work scheduling. A sixth participant was interrupted by an earthquake during the retest session and was not included in the final sample. In total, 15 participants completed both test and retest. The characteristics of the six participants who failed to complete the retest session were similar to the other participants in their years of experience, position, age, gender and performance.

Of the 15 participants who attended both test and retest, 13 were male and two were female. The participants ranged in age from 31 to 58 years with a mean age of 40.7 years (SD = 8.4). The participants had accumulated between two and 32 years of experience in the power transmission domain, with a mean of 11.2 years (SD = 9.9).

Measures. The present study used two measures, a demographics survey and EXPERTise.

Demographic Survey. The demographic survey was identical to that employed in Study One, including years of experience in the domain.

Expertise. The EXPERTise tasks were identical to those used in Study One, with the stimuli and scenarios adapted for power-control.

Stimuli. The stimuli employed in Study Two were developed using the same interview protocols described in Study One. Two subject matter experts from the power control domain were interviewed. Figure 2 provides an example of the stimuli used in Study Two.

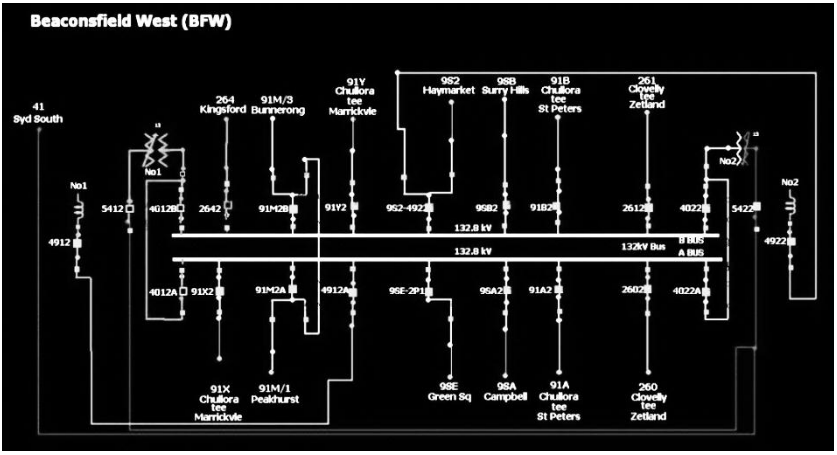


Fig. 2. Example network map and supervision control output

Procedure. The participants were asked to participate in the study during their scheduled work breaks. They were tested in single participant sessions and were briefed on the purpose of the study and asked to sign a consent form if they wished to continue. They completed the demographics questionnaire and then began the EXPERTise test battery online at their workstation. The participants were retested using the same procedure six-months later.

3.3 Results

Correlations with Experience. To investigate the relationship between years of experience in power control and each task within EXPERTise, bivariate correlations were undertaken between years of experience1 in the domain general and performance on the EXPERTise tasks. Consistent with expectations, years of experience in the domain yielded only weak to moderate Pearson correlations with performance on the EXPERTise tasks, $0 \leq r \leq 0.40, p < 0.05$.

Cluster Models. One of the aims of Study Two was to replicate the findings of Study One in a dissimilar domain, by identifying expert power controllers using tasks in which pattern recognition was advantageous. Consequently, following the initial test, participants were classified into two groups (non-expert and expert) using the k-means cluster procedure with *k = 2* groups.

Table 2. Participant cluster means for Study 2

Measure	Non-ExpertMean (SD)	ExpertMean (SD)	Overall Mean (SD)
Feature Identification Reaction Time	8.1 (6.1)	5.5 (1.9)	6.1 (3.4)
Feature Identification Accuracy	10.2 (1.8)	10.5 (2.3)	10.4 (2.1)
Paired Association 1 Reaction Time	10.5 (5.31)	6.4 (2.1)	7.4 (3.5)
Paired Association 1 Variance	3.2 (.60)	4.3 (.52)	4.0 (.7)
Paired Association 2 Reaction Time	5.2 (2.3)	4.2 (1.5)	4.4 (1.7)
Paired Association 2 Variance	3.5 (.8)	4.5 (.8)	4.3 (.9)
Feature Discrimination Variance	7.4 (3.3)	11.8 (3.4)	10.7 (3.8)
Transition Ratio	.76 (.1)	.68 (.23)	.70 (.21

Table 2 lists the results of the k-Means cluster analysis. As expected, two groups formed based on performance across the EXPERTise tasks.

Cluster One (*n = 5*) comprised those power controllers who demonstrated a lower level of performance across the EXPERTise tasks in comparison to the members of Cluster Two. Therefore, the participants in this cluster were described as 'non-experts'.

Cluster Two (*n = 16*) comprised those individuals who demonstrated superior performance across the EXPERTise tasks. Since the members of this cluster were generally faster, more accurate, more discriminating, and less sequential in their acquisition of information, they were described as 'experts.'

EXPERTise Classification Reliability. The participants were re-classified following retest six months later using the same cluster procedure. The results of the EXPERTise classifications at Test and Retest are summarized in Table 3.

An analysis of classification consistency was undertaken using the Kappa statistic. The consistency of classifications over the six-month interval was moderate, Kappa = 0.59, p < .05. In total, 80% of the participant's received the same classification at test and retest.

Table 3. Test by retest EXPERTise classifications

		Retest		
		Non-expert	**Expert**	**Total**
Test	**Non-expert**	4	0	4
	Expert	3	8	11
	Total	7	8	15

3.4 Study Two Discussion

Study Two was designed to replicate, in part, the results of Study One with diagnosticians drawn from a dissimilar domain. Specifically, it was designed to determine whether the EXPERTise measurements of pattern recognition could distinguish competent from expert power controllers within an experienced sample. It was hypothesized that expert power controllers would demonstrate consistently superior performance across the tasks.

The results of Study Two were consistent with the outcomes of Study One with performance across the four assessment tasks clustering into two levels. Moreover, Cluster Two demonstrated superior performance across all of the EXPERTise tasks, each of which was designed to measure a distinct component of expert pattern recognition. This suggests that Cluster 1 and Cluster 2 comprised, respectively, non-experts and experts within an experienced sample.

Study Two was also designed to determine whether normative classifications of expertise, based on performance during the EXPERTise tasks, were consistent over a six-month period. Since expertise typically requires more than ten years of dedicated practice [21], it was assumed that the interval between assessments was insufficient to produce genuine and significant changes in expert performance. Therefore, it was expected that non-expert and expert classifications, based on standardized norms, would be relatively consistent at test and retest.

Overall, the results supported the expectation that normative classifications would be consistent at test and retest. In fact, 80% of the participants received the same classification in both sessions. Therefore, EXPERTise appears to produce a consistent classification of domain expertise.

4 General Discussion and Conclusions

A number of previous studies of expert diagnosis have been based on assessments of expertise using years of experience within a domain [6]; [21-24]. Although these comparisons can be useful, they are based on a linear relationship between experience and diagnostic performance [1, 2]. However, in the present studies, performance on four expertise assessment tasks was only weakly associated with years of experience in the domain. Therefore, while years of experience may be a necessary precursor to expert diagnostic performance, it will not inherently confer expertise within the domain.

The results of Studies One and Two suggest that when investigating diagnostic performance, expertise should not be operationalized simply as years of experience in the domain or role. Rather, an alternative approach is to identify expertise using pattern recognition performance during domain-relevant tasks. In both studies, two distinct clusters emerged that appeared to represent two distinct levels of performance. These levels were consistent with the distinction made by Gray [3] between competent non-expert and expert practitioners. Moreover, these differences in performance were consistent across all four assessment tasks, each of which was designed to assess an independent dimension of expert pattern recognition [17]; [18]; [25-27].

The identification of diagnostic experts on the basis of their performance, rather than their years of experience in the domain, should assist with studies of feature extraction, pattern recognition and empirical comparisons between different levels of diagnostic performance. Further, the identification of genuine experts ought to improve the validity of research outcomes involving the observation of expert performance and, perhaps, provide the basis for an improved understanding of the process of cognitive skill acquisition.

At an applied level, the present results have important implications for evaluation and training. In particular, it is apparent that normative assessments of pattern-recognition during domain-relevant tasks are: (a) able to distinguish levels of expertise; and (b) are able to achieve a level of consistency necessary to track the acquisition of expertise over time. Consequently, the EXPERTise tool appears to provide a method of assessing the progression towards diagnostic expertise.

With the development of standardized norms, it should be possible to determine whether an individual learner is developing diagnostic skills consistent with expectations and/or whether a particular level of performance has been achieved following exposure to specialist training. By assessing the four components of expert pattern recognition, EXPERTise can also be used to identify the individual skills that an experienced competent practitioner may be struggling to acquire. This information can then guide remedial training efforts. Such cue-based approaches to training have already met with some success in other domains, including aviation [14] and mining [22].

Each of the EXPERTise assessment tasks were designed to assess a distinct component of expert pattern recognition and diagnosis. Therefore, if performance is relatively weaker on one or more of the tasks, it should be possible to identify the specific area of deficiency and thereby better target interventions. The application of this strategy can be used to improve the efficiency and the effectiveness of remedial diagnostic training and, as a consequence, minimize the costs associated with training interventions.

The present series of studies were designed to determine whether four independent assessments of expert pattern-recognition could, collectively, distinguish competent from expert practitioners within experienced samples of diagnosticians. In both power control and medicine, performance on all four assessment tasks successfully differentiated two groups, whereby experienced diagnosticians could be divided into competent and expert practitioners based on their capacity for pattern recognition or cue utilization.

The successful differentiation of non-experts and experts in dissimilar diagnostic domains demonstrates the utility of the EXPERTise tasks. It also highlights the importance of pattern-recognition in expert performance generally. In time, pattern-recognition based assessments, like EXPERTise, may be used to determine whether experienced practitioners are developing expertise at a rate that is consistent with their peers. Individuals' who perform at an unsatisfactory level may benefit from remedial training. It is expected that this combination of progressive assessment and remedial training may reduce the rate and severity of errors involving diagnosis.

References

1. Ericsson, K.A., Lehmann, A.C.: Expert and exceptional performance: Evidence of maximal adaption to task constraints. Annual Review of Psychology 47, 273–305 (1996)
2. Shanteau, J., Stewart, T.R.: Why study expert decision making? Some historical perspectives and comments. Organisational Behaviour and Human Decision Processes 53, 95–106 (1992)
3. Gray, R.: Attending to the execution of a complex sensorimotor skill: Expertise, differences, choking, and slumps. Journal of Experimental Psychology: Applied 10, 42–54 (2004)
4. Rasmussen, J.: Skills, rules, and knowledge: signals, signs, and symbols, and other distinctions in human performance models. IEEE Transactions on Systems, Man, and Cybernetics SMC-13, 257–266 (1983)

5. Wiggins, M.: Cue-based processing and human performance. In: Karwowski, W. (ed.) Encyclopaedia of Ergonomics and Human Factors, pp. 641–645. Taylor & Francis, London (2006)
6. Coderre, S., et al.: Diagnostic reasoning strategies and diagnostic success. Medical Education 37, 695–703 (2003)
7. Groves, M., O'Rourke, P., Alexander, H.: The clinical reasoning characteristics of diagnostic experts. Medical Teacher 25, 308–313 (2003)
8. Norman, G.R., Young, M., Brooks, L.: Non-analytical models of clinical reasoning: the role of experience. Medical Education 41(12), 1140–1145 (2007)
9. Croskerry, P.: A universal model of diagnostic reasoning. Academic Medicine 84(8), 1022–1028 (2009)
10. Klein, G.: Recognition-primed decisions (RPD). Advances in Man-Machine Systems 5, 47–92 (1989)
11. Jones, M.A.: Clinical reasoning in manual therapy. Physical Therapy 72, 875–884 (1992)
12. Schimdt, H.G., Boshuizen, H.P.A.: Acquiring expertise in medicine. Educational Psychology Review 3, 205–221 (1993)
13. Sweller, J.: Cognitive Load During Problem Solving: Effects on Learning. Cognitive Science 12, 257–285 (1988)
14. Wiggins, M., O'Hare, D.: Weatherwise: an evaluation of a cue-based training approach for the recognition of deteriorating weather conditions during flight. Human Factors 45, 337–345 (2003)
15. Morrison, B.W., et al.: Examining cue recognition across expertise using a computer-based task. In: NDM 2009, the 9th International Conference on Naturalistic Decision Making, London (2009)
16. Schriver, A.T., et al.: Expertise Differences in Attentional Strategies Related to Pilot Decision Making. Human Factors 50(6), 864–878 (2008)
17. Weiss, D.J., Shanteau, J.: Empirical Assessment of Expertise. Human Factors. The Journal of the Human Factors and Ergonomics Society 45, 104–114 (2003)
18. Wiggins, M., et al.: Expert, intermediate and novice performance during simulated preflight decision-making. Australian Journal of Psychology 54, 162–167 (2002)
19. Anderson, J.R.: A spreading activation theory of memory. Journal of Verbal Learning and Verbal Behavior 22, 261–295 (1983)
20. Fitts, P.M., Posner, M.I.: Human Performance. Brooks/Cole, Belmont (1967)
21. Simon, H.A., Chase, W.G.: Skill in chess. American Scientist 61, 394–403 (1973)
22. Blignaut, C.J.: The perception of hazard: I. Hazard analysis and the contribution of visual search to hazard perception. Ergonomics 22, 991–999 (1979)
23. O'Hare, D., et al.: Finding the Right Case: The Role of Predictive Features in Memory for Aviation Accidents. Applied Cognitive Psychology 22, 1163–1180 (2008)
24. Wallis, T.S.A., Horswill, M.S.: Using fuzzy signal detection theory to determine why experienced and trained drivers respond faster than novices in a hazard perception test. Accident Analysis and Prevention 39, 1177–1185 (2007)
25. Morrison, B.W., Wiggins, M., Porter, G.: User Preference for a Control-Based Reduced Processing Decision Support Interface. International Journal of Human-Computer Interaction 26(4), 297–316 (2010)
26. Ratcliff, R., McKoon, G.: Sequential effects in lexical decision: Tests of compound cue retrieval theory. Journal of Experimental Psychology: Learning, Memory, and Cognition 21, 1380–1388 (1995)
27. Wiggins, M., O'Hare, D.: Expertise in Aeronautical Weather-Related Decision Making: A Cross-Sectional Analysis of General Aviation Pilots. Journal of Experimental Psychology: Applied 1, 305–320 (1995)

Analysis of Factorial Designs with the Consideration of Interactions via the Stepwise Response Refinement Screener (SRRS)

Frederick Kin Hing Phoa

Institute of Statistical Science, Academia Sinica,
128 Academia Road Section 2, Nangang District, Taipei City 115, Taiwan R.O.C.
fredphoa@stat.sinica.edu.tw

Abstract. Factorial designs are widely used experimental plans for identifying important factors in screening studies where many factors are involved. In many practical situations where some interactions are significant, the design is supersaturated and the experimental analysis becomes infeasible due to the lack of degree of freedoms [9]. Recently, a new analysis procedure called the Stepwise Response Refinement Screener (SRRS) method is proposed to screen important effects for supersaturated designs [6]. This paper extends this method to the two-level factorial designs. The applications to several real-life examples suggest that the SRRS method is able to retrieve similar results as the existing methods do. Simulation studies show that compared to existing methods in the literature, the SRRS method performs well in terms of the true model identification rate and the average model size.

Keywords: Stepwise Response Refinement Screener (SRRS), Akaike Information Criterion (AIC), Screening experiment, factorial designs, Supersaturated designs.

1 Introduction

As science and technology have advanced to a higher level nowadays, investigators are becoming more interested in and capable of studying large-scale systems. To address these challenges of expensive experimental costs, research in experimental design has lately focused on the class of supersaturated designs (SSD) for their run-size economy and mathematically novelty. Under the condition of factor sparsity [2], these experiments aims at correctly identifying the subset of those active factors that have significant impact on the response, so that the whole investigation can be economically proceed via discarding inactive factors prior to the follow-up experiments.

Traditionally, SSDs are employed only for screening main effects, and interactions are discarded due to limited degree of freedom. More refined analysis methods were recently developed and Phoa, Pan and Xu (2009) [8] provides a comprehensive list of recent analysis methods found in the literature. Candes and Tao (2007) [3] proposed the Dantzig selector (DS) and showed that it has some remarkable properties under some conditions. Phoa, Pan and Xu (2009) [8] implemented the DS in practice, introducing a graphical procedure via a profile plot for analysis and an automatic variable selection

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 13–21.
DOI: 10.1007/978-3-642-36530-0_2

procedure via a modified Akaike information criterion (AIC). Traditionally, AIC is used for model selection. For linear models, it is defined as

$$AIC = n\log(RSS/n) + 2p \tag{1}$$

where $RSS = \sum_{i=1}^{n}(y_i - \hat{y_i})^2$ is the residual sum of squares and p is the number of parameters in the model. It is known that AIC tends to overfit the model when the sample size is small. Phoa, Pan and Xu (2009) [8] imposed a heavy penalty on the model complexity and proposed a new modified AIC for the DS method, which is defined as

$$mAIC = n\log(RSS/n) + 2p^2 \tag{2}$$

The $mAIC$ typically chooses a smaller model than AIC.

Recently, Phoa (2012) [6] introduce a new variable selection approach via the Stepwise Response Refinement Screener (SRRS). The SRRS chooses the best subset of variables or active factors by two procedures: Factor Screening and Model Searching. This method has shown its superior model selection ability via a comparison to five commonly used methods in the literature, namely SSVS [4], SSVS/IBF [1], SCAD [5], PLSVS [17] and the DS [8] method. Readers who are interested in the main idea of the SRRS method are referred to Phoa (2012) [6]. This paper aims at extending the SRRS method to the variants of supersaturated experiments. In section 2, we proposes the procedure of SRRS with Heredity Prinicple, which is modified from the original version introduced in Phoa (2012) [6]. Some notes about the procedure and modifications are disucssed briefly in this section. To demonstrate the value of the SRRS method, two real-life examples are demonstrated in section 3 and a simulation study is performed in section 4. The result shows that the SRRS method is powerful for analyzing not only SSDs but also its variant designs. The last section gives some concluding remarks.

2 Analysis of Fractional Factorial Designs via the SRRS Methods

Fractional factorial designs (FFDs) are classified into two broad types: Regular FFDs and Nonregular FFDs. Regular FFDs are constructed through defining relations among factors and are described in many textbooks [14]. These designs have been widely used in scientific researches and industrial processes because they are simple to construct and to analyze. On the other hands, nonregular FFDs such as Plackett and Burman (1946) [12] designs, Quaternary-code designs (Phoa and Xu 2009 [10], Zhang et. al. 2011 [16]) and other orthogonal arrays are often used in various screening experiments for their run size economy and flexibility [14]. Phoa, Xu and Wong (2009) [11]demonstrated the advantages of using nonregular FFDs using two real-life toxicological experiments. Phoa, Wong and Xu (2009) [9] used three real-life chemometrics examples to show the analysis pitfalls when the interactions are assumed to be insignificant without verifications.

In this section, we extend the use of the SRRS method to the analysis of two-level nonregular fractional factorial designs (FFDs).

2.1 Modification of the SRRS Method Accompanied for the Analysis of Nonregular Designs

Consider a nonregular FFDs with k_1 main effects and n runs, where $n < m$. There are $k_2 = k_1(k_1 + 1)/2$ interactions between two different main effects. If all two-factor interactions are considered together with all main effects, it is possible that $k_2 > m$, then the design matrix is supersaturated. We express the relationship via a linear regression model $y = X\beta + \epsilon$ where y is an $n \times 1$ vector of observations, X is an $n \times k$ model matrix for $k = k_1 + k_2$, β is a $k \times 1$ vector of unknown parameters, and ϵ is an $n \times 1$ vector of random errors. Assume that $\epsilon \sim N(\mathbf{0}, \sigma^2 I_n)$ is a vector of independent normal random variables. In addition, X is assumed to be supersaturated, i.e. $n < k$. We denote m to be the number of potentially important effects (PIEs) and S_{inf} to be the influential set of PIEs found in the process.

Traditionally, the analysis of nonregular FFDs is based on two assumptions: the factor sparsity principle and the effect heredity prinicple. The first assumption has been embedded in the SRRS method, but the second assumption does not. In order to implement the heredity principle into the SRRS method, the two procedures of the SRRS method are slightly modified and presented in the following steps:

I. SRRS (with Heredity Prinicple)–Factor Screening:

Step 1. Standardize data so that y_0 has mean 0 and columns of X have equal lengths.

Step 2. Compute the marginal correlations $\rho(X_i, y_0)$ for all main effects X_i, $i = 1, \ldots, k$. $(*)$

Step 3. Choose E_0 such that $|\rho(E_0, y_0)| = \max_{X_i} |\rho(X_i, y_0)|$. Identify E_0 as the first PIE and include E_0 in S_{Inf}.

Step 4. Obtain the estimate β_{E_0} by regressing y_0 on E_0.

Step 5. For the next m PIEs E_j where $j = 1, \ldots, m$, $m < n - 2$,

(a) Compute the refined response $y_j = y_{j-1} - E_{j-1}\beta_{E_{j-1}}$.

(b) Compute the marginal correlations $\rho(\{X_i, X_{ij}\}, y_j)$ for all main effects X_i, $i = 1, \ldots, k$ and all two-factor interactions X_{ij}, $X_j \in S_{Inf}$. $(*)$

(c) Choose T_j such that $|\rho(T_j, y_j)| = \max_{\{X_i, X_{ij}\}} |\rho(\{X_i, X_{ij}\}, y_j)|$. $(*)$

(d) Obtain the estimate β_{T_j} by regressing y_j on $E_0, \ldots, E_{j-1}, T_j$.

(e) If $|\beta_{T_j}| \geq \gamma$ and has not been included in S_{inf}, identify T_j as a PIE (i.e. $E_j = T_j$) and include E_j in S_{inf}.

(f) Repeat (a) to (e) up to m^{th} step, where $E_j = E_m$ is not included in S_{inf}. m is determined by either $m < n - 2$ or the threshold condition $|\beta_{T_j}| \geq \gamma$ or both.

II. SRRS (with Heredity Principle)–Model Searching:

Step 6. Perform all-subset search, with the consideration of the heredity principle, for all E_j, from models with one factor to models with m factors, where m is minimum between the ceiling of $n/3$ or the number of E_j in S_{inf}. $(*)$

Step 7. Compute $mAIC$ for each model and choose the final model with the smallest $mAIC$ among all models, and all E_j included in the final model are considered to be significant to the response y_0.

2.2 A Brief Discussion on the Main Idea of the SRRS Method

There are several notes that needs further discussions in the procedure of the SRRS method, including the threshold $gamma$, the refined response, the stopping criteria and the modifications from the original SRRS method used in the analysis of supersaturated designs.

The first note is about $gamma$. It is a threshold between signal and noise and a relatively small γ should be chosen. One can choose γ according to the information on the magnitude of effects or noise. For example, Phoa, Pan and Xu (2009) [8] suggested to choose γ to be approximately 10% of the maxmimum absolute estimates in their simulation study. It is recommended that the procedure should be repeated with a few choices of γ. When the signal and noise ratio is large, the choice of γ is not crucial. However, if the result is sensitive to the choice of γ, one should be cautious about the procedure and the result. Generally speaking, we choose γ to be approximately $5\% - 10\%$ of $|\beta_{E_0}|$ in the examples and simulation studies of this paper. Although $|\beta_{E_0}|$ may not be the maximum absolute slope estimate in some cases, it is conservative to set a slightly smaller γ, so that one or two more factors are considered as PIEs.

The second note is about the refined response. y_j is obtained by reducing a portion of magnitude that only corresponds to T_{j-1} from y_{j-1}. The portions of magnitude that corresponds to all other PIEs are preserved in y_j. Therefore, only the marginal correlation between y_j and T_{j-1} will be zero, and the marginal correlations between y_j and all other factors, including those that have been included in S_{inf}, are compared in step 5(b). The magnitudes of these marginal correlations consist of: (i) some middle to high values, which indicate that these factors still have possibilities to be PIEs after j refinements, and (ii) some close-to-zero values, which indicates that these factors do not have impact on the response anymore. Thus, the selection in Step 5(c) can be interpreted as the selection of the PIEs that has the highest marginal correlation to the refined response.

The third note is about the stopping criteria. There are two criteria that can stop the search. The first criterion is the number of PIEs in S_{inf}. The Model Searching procedure of the SRRS requires to build the regression models between the PIEs in S_{inf} and the original response y_0. This means the number of PIEs has to be at most the number of runs minus two, so that there are enough degrees of freedom to estimate all PIEs, the intercept and the residual of the model, or otherwise the design is supersaturated again. It leads to the first criterion: $m < n - 2$. The second criterion is related to the magnitude of the slope estimate. Any magnitudes that are lower than γ are considered as noise. If E_j is chosen in Step 5(c) but $|\beta_{E_j}|$ is found to be smaller than γ, Step 5(e) suggests that E_j is not a PIE. Then the search stops because even if $|\beta_{E_j}| < \gamma$, all other factors with smaller absolute marginal correlations to y_j have smaller slope estimates than $|\beta_{E_j}|$, and so all these slope estimates, whose absolutes are smaller than γ, will be considered as noise.

The last note is about the modification. There are several modifications in the current version specified for embedding the Heredity Principle. The first modification is in Step 2. Due to the heredity principle, two-factor interactions can never be selected as the first PIE, so only the marginal correlations of all main effects are compared for selecting the first PIE. The second and third modifications are in Step 5. During the search of the

j^{th} PIE, not all two-factor interactions are considered in the comparison of marginal correlation. Heredity principle suggests that a two-factor interaction X_{ij} is considered in Step 5(b) if and only if either X_i or X_j or both parents main effects have been included in S_{Inf} in the previous searches. Therefore, the modifications in Step 5 take away a subset of two-factor interactions that none of their corresponding parent main effects have been PIEs. The last modification is in Step 6. The reduced models built in this step must follow the heredity principle in order to avoid the situation that some significant two-factor interactions are included in the reduced model but none of their parent main effects have been included.

2.3 Two Illustrating Examples

We illustrate the analysis of nonregular FFDs via the SRRS method step by step using the following two examples. The Factor Screening procedures is terminated via the noise threshold in the first example and via the maximum number of PIEs in the second example.

Example 1. Consider the cast fatigue experiment (Wu and Hamada 2000 [14], section 7.1), a real data set consisting of seven two-level factors. The design matrix and the response are found in Wu and Hamada (2000) [14]. When all two-factor interactions are considered to be as important as the main effects, the design matrix consists of 21 additional interactions and is supersaturated.

In the Factor Screening procedure, the first PIE being identified is F and its absolute marginal correlation to y_0 is the highest among all main effects (0.6672). A regression model between y_0 and F is built and the magnitude of the slope estimate $|\beta_F| = 0.4576$. Then we set the threshold $\gamma = 0.04$, about 10% of β_F.

To search for the second PIE, the new response y_1 is refined by subtracting $F\beta_F$ from y_0. Then among all main effects and all the two-factor interactions that consist of F, FG (the interaction between main effects F and G) has the highest absolute marginal correlation (0.8980) to y_1 and so it is identified as the second PIE. A regression model between y_1 and FG, F is built and the magnitude of the slope estimate $|\beta_{FG}| = 0.4588 > \gamma$. This means FG is important enough to be included in the influential set S_{Inf} together with F.

The procedure continues to search for the next five PIEs. Table 1 shows every step of the process of Factor Screening. Note that in the last step, the absolute magnitude of the slope estimate of AE is close to 0, so the search stops and seven PIEs are identified in the Factor Screening procedure.

Since there are 12 observations in the data, the maximum number of active factors is suggested to be 4. There are totally 98 reduced models up to four-factors models that are constructed from seven PIEs, but only 49 of them fulfill the heredity principle. A comparison of the $mAIC$s of these 49 reduced models shows that the two-effects model with F and FG has the lowest $mAIC = -27.82$. Thus the SRRS method suggests that F and FG have significant impacts to the response y_0. This result is also recommended by Wu and Hamada (2000, Section 8.4) [14] and the Dantzig selector (DS) method in Phoa, Pan and Xu (2009) [8].

Table 1. Factor Screening of Cast Fatigue Experiment Data

m	PIE	Marginal Correlation	$\lvert\beta\rvert$	Continue or Stop
0	F	0.6672	0.4576	Continue
1	FG	-0.8980	0.4588	Continue
2	D	-0.4677	0.1183	Continue
3	EF	-0.6336	0.1442	Continue
4	C	0.5032	0.0758	Continue
5	E	-0.5817	0.0785	Continue
6	AE	-0.7667	0.1482	Continue
	AE	-0.6835	0	Stop

PIEs in S_{Inf} after Factor Screening:
C, D, E, F, AE, EF, FG

Example 2. Consider the high-performance liquid chromatography (HPLC) experiment [13], a real data set consisting of eight two-level factors. The design matrix and the response are found in Phoa, Wong and Xu (2009) [9]. When all two-factor interactions are considered to be as important as the main effects, the design matrix consists of 28 additional interactions and is supersaturated.

In the Factor Screening procedure, the first PIE being identified is E and its absolute marginal correlation to y_0 is the highest among all main effects (0.5019). A regression model between y_0 and E is built and the magnitude of the slope estimate $|\beta_F| = 0.5583$. Then we set the threshold $\gamma = 0.05$, about 10% of β_E.

To search for the second PIE, the new response y_1 is refined by subtracting $E\beta_E$ from y_0. Then among all main effects and all the two-factor interactions that consist of E, EF (the interaction between main effects E and F) has the highest absolute marginal correlation (0.8055) to y_1 and so it is identified as the second PIE. A regression model between y_1 and EF, E is built and the magnitude of the slope estimate $|\beta_{EF}| = 0.7750 > \gamma$. This means EF is important enough to be included in the influential set S_{Inf} together with E.

The procedure continues to search for the next eight PIEs. Table 2 shows every step of the process of Factor Screening. Note that in the last step, although the absolute magnitude of the slope estimate of EF is $0.0667 > \gamma$, the $m < n - 2$ criterion stops the search and nine PIEs are identified in the Factor Screening procedure.

Since there are 12 observations in the data, the maximum number of active factors is suggested to be 4. With nine PIEs found in the previous step, there are totally 255 reduced models up to four-factors models, but only 102 of them fulfill the heredity principle. A comparison of the $mAIC$s of these 102 reduced models shows that the three-effects model with E, F and EF has the lowest $mAIC = -6.48$. Thus the SRRS method suggests that E, F and EF have significant impacts to the response y_0.

Phoa, Wong and Xu (2009) [9] previously analyzed the same data and concluded that an additional effect H was also significant to the response. The $mAIC$ of the model consisting of E, F, H and EF is -3.95, which is slightly higher than our suggested model. The increase of $mAIC$ when H is added comes from the heavy penalty to the number of factors in the model. If other penalty terms are used, results may be different. For example, the original AIC favors the addition of H. Therefore, H may be barely

Table 2. Factor Screening of HPLC Experiment Data

m	PIE	Marginal Correlation	$\lvert\beta\rvert$	Continue or Stop
0	E	-0.5019	0.5583	Continue
1	EF	0.8055	0.7750	Continue
2	F	0.7747	0.4417	Continue
3	H	-0.7396	0.3000	Continue
4	FH	0.5897	0.1625	Continue
5	A	0.6922	0.1389	Continue
6	FI	-0.5295	0.0893	Continue
7	EI	0.5713	0.0836	Continue
8	AF	-0.6587	0.0792	Continue
	EF	0.6951	0.0667	Continue
PIEs in S_{Inf} after Factor Screening: $A, E, F, H, AF, EF, EI, FH, FI$				

significant and some follow-up experiments are suggested to investigate the significance of H to the response.

3 Simulation Studies

In order to judge the value of the SRRS method, we randomly generate some models and evaluate the performance of the SRRS method.

Example 3. In this example, we generate data from the same linear model as in Example 1. Since there are only 12 observations in the data, the maximum possible number of active factors is 4. Therefore, we consider four cases for $beta$. There are i active factors for case i, $1 \leq i \leq 4$. For each case, we generate 500 models where the selection of active factors is random without replacement, the signs of the active factors are randomly selected from either positive or negative, and the magnitudes are randomly

Table 3. Summary of Simulation Results in Example 3

Case		I	II	III	IV
Min	TMIR	94%	47%	5%	0%
	Size	1.00	1.85	2.05	1.06
1st Q.	TMIR	97%	97%	44%	15%
	Size	1.01	2.01	3.00	2.42
Median	TMIR	98%	97%	96%	53%
	Size	1.02	2.02	3.00	3.30
3rd Q.	TMIR	99%	99%	99%	88%
	Size	1.03	2.03	3.01	3.76
Max	TMIR	100%	100%	100%	99%
	Size	1.06	2.05	3.04	3.98

selected from 2 to 10 with replacement. For each model, we generate data 100 times and obtain the True Model Identified Rate (TMIR) and the average model size. In the simulations we fix $\gamma = 1$, which is approximately equal to 10% of $\max|\beta_i|$. Table 3 gives the summary statistics of these two quantities among 500 models.

The SRRS method is very effective in identifying $1, 2$ and 3 active factors; the TMIR in these cases are at least 96% in average true model identified rate and only a few cases that have average model sizes slightly higher than the true numbers of active factors. The performance of the method decreases in identifying 4 active factors. It is mainly because of the limit posted on the allowed number of active factors, which leads to a slightly underfitting situation.

4 Concluding Remarks

The Stepwise Response Refinement Screener (SRRS) method has shown its satisfactory performance on screening the supersaturated designs in Phoa (2011) [6]. In this paper, we modify the SRRS method in order to adapt for analyzing the nonregular FFDs with the consideration of interactions. Under the validity of the factor sparsity and effect heredity assumptions, the calculations needed to carry out the analysis are simple and easily performed with little computation time. Simulation suggests that the SRRS method performs well in most of the cases, except when it is on the line of maximum number of allowed active factors. However, we cannot ensure that this method works well in every case, and sometimes it may still possible to reach misleading conclusion. Although some theoretical works are still under investigation, the results of the SRRS are shown to be interesting from a partitioner point of view. The R function of the SRRS is available by email request from the author, and the standalone program for the SRRS will be available soon.

It is highly recommended that once the suggested set of significant factors is found, a follow-up experiment is needed for validating the results. It is more economical and efficient to use nonregular fractional factorial designs than full factorial designs in the validation process. A detailed review on nonregular fractional factorial designs is referred to Xu, Phoa and Wong (2009) [15] and a systematic construction method for nonregular fractional factorial designs of the required size is referred to Phoa and Xu (2009) [10] and Phoa, Mukerjee and Xu (2012) [7].

The procedure of the SRRS suggested in this paper can be easily modified and extended to the analysis when higher order interactions are found to be significant. For example, if it is necessary for considering the significance of three-factor interactions to the impact of the response, the procedure can be slightly modified to accomodate the inclusion of three-factor interactions under the rule of Heredity Prinicple. The procedure can also be extended to multi-level factorial designs with certain transformations.

Acknowledgements. This work was supported by National Science Council of Taiwan ROC grant number 100-2118-M-001-002-MY2.

References

1. Beattie, S.D., Fong, D.K.H., Lin, D.K.J.: A two-stage Bayesian model selection strategy for supersaturated designs. Technometrics 44, 55–63 (2002)
2. Box, G.E.P., Meyer, R.D.: An analysis for unreplicated fractional factorials. Technometrics 28, 11–18 (1986)
3. Candes, E.J., Tao, T.: The Dantzig selector: statistical estimation when p is much larger than n. Annals of Statistics 35, 2313–2351 (2007)
4. Chipman, H., Hamada, H., Wu, C.F.J.: A Bayesian variable-selection approach for analyzing designed experiments with complex aliasing. Technometrics 39, 372–381 (1997)
5. Li, R., Lin, D.K.J.: Analysis methods for supersaturated design: some comparisons. Journal of Data Science 1, 249–260 (2003)
6. Phoa, F.K.H.: The Stepwise Response Refinement Screener (SRRS) (in reivew)
7. Phoa, F.K.H., Mukerjee, R., Xu, H.: One-Eighth- and One-Sixteenth-Fraction Quaternary Code Designs with High Resolution. Journal of Statistical Planning and Inference 142, 1073–1080 (2012)
8. Phoa, F.K.H., Pan, Y.H., Xu, H.: Analysis of supersaturated designs via the Dantzig selector. Journal of Statistical Planning and Inference 139, 2362–2372 (2009)
9. Phoa, F.K.H., Wong, W.K., Xu, H.: The need of considering the interactions in the analysis of screening designs. Journal of Chemometrics 23, 545–553 (2009)
10. Phoa, F.K.H., Xu, H.: Quater-fraction factorial design constructed via quaternary codes. Annals of Statistics 37, 2561–2581 (2009)
11. Phoa, F.K.H., Xu, H., Wong, W.K.: The use of nonregular fractional factorial designs in combination toxicity studies. Food and Chemical Toxicology 47, 2183–2188 (2009)
12. Plackett, R.L., Burman, J.P.: The design of optimum multifactorial experiments. Biometrika 33, 305–325 (1946)
13. Vander-Heyden, Y., Jimidar, M., Hund, E., Niemeijer, N., Peeters, R., Smeyers-Verbeke, J., Massart, D.L., Hoogmartens, J.: Determination of system suitability limits with a robustness test. Journal of Chromatography A 845, 145–154 (1999)
14. Wu, C.F.J., Hamada, M.: Experiments: Planning, Analysis, and Parameter Design Optimization. Wiley, New York (2000)
15. Xu, H., Phoa, F.K.H., Wong, W.K.: Recent Developments in Nonregular Fractional Factorial Designs. Statistics Surveys 3, 18–46 (2009)
16. Zhang, R., Phoa, F.K.H., Mukerjee, R., Xu, H.: A Trigonometric Approach to Quaternary Code Designs with Application to One-Eighth and One- Sixteenth Fractions. Annals of Statistics 39, 931–955 (2011)
17. Zhang, Q.Z., Zhang, R.C., Liu, M.Q.: A method for screening active effects in supersaturated designs. Journal of Statistical Planning and Inference 137, 235–248 (2007)

Object Detection with Semi-local Features

Robert Sorschag

Vienna University of Technology, Institute of Software Technology and Interactive Systems,
Favoritenstrasse 9-11, A-1040 Vienna, Austria
sorschag@ims.tuwien.ac.at

Abstract. Class-level object detection is a fundamental task in computer vision and it is usually tackled with global or local image features. In contrast to these approaches, we propose semi-local features that exploit object segmentation as a pre-processing step for detection. The term semi-local features depicts that the proposed features are locally extracted from the image but globally extracted from the object. In particular, we investigate the impact of features generation approaches from differently transformed object regions. These transformations are, on the one hand, done with several object-background modifications and bounding-boxes. On the other hand, state-of-the-art texture and color features as well as different dissimilarity measures are compared against each other. We use the Pascal VOC 2010 dataset for evaluation with perfect and inaccurate object segments and to perform a case study with an automatic segmentation approach. The results indicate the high potential of semi-local features to assist object detection systems and show that a significant difference exists between different feature extraction methods.

Keywords: Object detection, Visual features, Segmentation.

1 Introduction

Recently, a set of object detection approaches have been proposed where segmentation is used as a pre-processing step [1-4]. They outperform sliding window approaches although almost the same features and classification techniques are used. We believe that customized features that are less distracted by the object's background can further improve these results. Thus, the main research question of this work is: How to extract state-of-the-art texture and color features best from segmented objects to improve detection systems? The proposed semi-local features exploit different region modifications to set the focus on specific object properties. Furthermore, these features are simple and fast to compute which makes them suitable to assist segmentation-based object detection systems.

Generally, the detection of class-level objects in real-world images is a challenging task for automated systems that is far from solved. Objects can be situated everywhere and at every size in an image. They can be occluded and shown under all kinds of perspective distortions or under different lighting conditions. Moreover, intra class differences and inter class similarities can complicate this task. Even humans sometimes fail to distinguish between closely related classes like bicycles and motorbikes,

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 23–35.
DOI: 10.1007/978-3-642-36530-0_3

when only a single image with difficult examples is shown. However, the complexity of object detection can be reduced when a set of segmented object hypotheses are given in the first place [2] as it is accurately known where to search for an object.

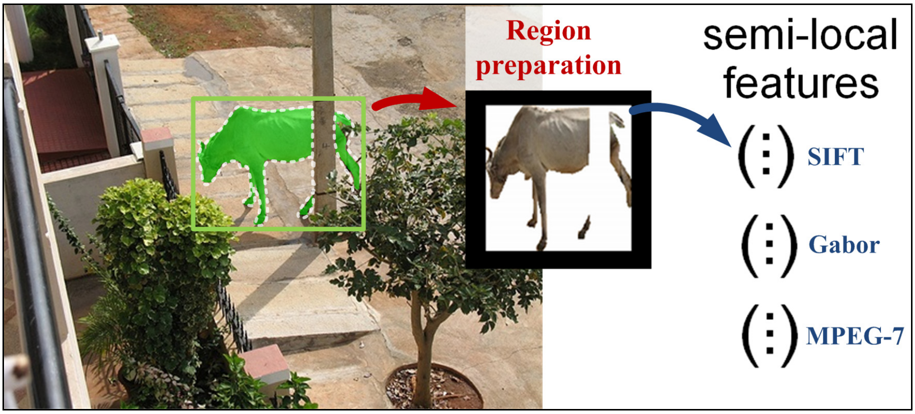


Fig. 1. Semi-local features: A region that covers the entire segmented cow is prepared (here: background replaced by white pixels) to extract color and texture features from it

In this work, we extract well-established image features semi-locally from segmented objects. Thereby, color and texture features are generated from image regions that contain the entire object. We use the term semi-local features because these features are locally extracted from the image but globally extracted from the object. Furthermore, we show that the use of differently prepared image regions facilitates the power of these features. For instance, the object background is excluded and replaced by white pixels in Fig. 1.

This work contributes to object detection research with an extensive study on the suitability of semi-local features for the classification of segmented objects and the influence of different region preparation techniques. The used set of image features and dissimilarity measures should ensure that the evaluation results are as universally valid as possible. On the one hand, we work on interactively generated segmentations that are provided by the Pascal VOC challenge [5] and use a simple nearest neighbor classification. In addition to this perfect segmentation, we simulate inaccurate segmentations for comparison. On the other hand, we perform a case study on automatically segmented regions to gain further insights into the semi-local feature approach.

The remainder of the paper is organized as follows. Section 2 describes related work in the field of object detection and segmentation. Section 3 presents semi-local features. Section 4 explains the experiments and Section 5 draws conclusions and future research directions.

2 Related Work

Local features [6] are a part of the best practice for object detection systems. First, these features are regularly sampled or extracted around interest-regions [7] before they are generalized to one or more bag-of-features (BoF) per image [8-9]. This BoF approach produces fixed-length vectors for classification. In order to locate objects within an image, many sub-regions are then investigated with a sliding window [10]. In addition to BoFs, global and semi-local features have been successfully used for related tasks, such as scene classification [11], geometric context retrieval [12], and human body detection [13].

2.1 Segmentation-Based Detection

Object detection approaches that operate on segmented objects [1-4] work similar to sliding window approaches but with a heavily reduced search-space. Thus, more powerful (and computationally more expensive) recognition approaches can be applied. However, this benefit is not extensively exploited so far: In [1] color histograms and RCF (regionSIFT) descriptors are extracted from the segmented objects. [2], [3], and [14] generate BoFs from SIFT [15], colorSIFT [8], local shape context [14], and gray-value patches. In [2] independent BoFs are extracted from the segmented object and its background within a bounding box as well as semi-local HoG features [13]. [3] sets all background pixels to black and extracts local features from interest-regions that overlap with the segmented object. We use a similar zero-masking step to generate features with a higher weighting of the object shape.

Only [16] propose segmentation specific features, called boundary object shape, where the geometric relations of object boundary edges are measured. We further explore this idea and propose customized features for the classification of segmented objects. To the knowledge of the authors, no work has been proposed so far that investigates such semi-local features for object detection.

2.2 Segmentation Approaches

Different object segmentation approaches including Normalized Cuts [17], MinCuts [18], and Mean-Shift [19] have been used for the object detection systems described above. A good overview of segmentation approaches can be found in [20]. In contrast to semantic segmentation [21], these approaches work without knowledge about the segmented objects and they are used to generate a ‘soup’ of many overlapping segmentations. Such multi-segmentation approaches can achieve higher object detection rates when overlapping segments are individually classified and combined afterwards [2]. All of the described object detection systems work with unsupervised segmentation. However, it can be useful to test single stages of such detection systems on interactively generated object segments that are almost perfect [1]. We use this strategy to compare different semi-local features that are extracted from perfectly and inaccurately segmented objects.

3 Semi-local Features

We extract and classify semi-local features from segmented objects in following steps. First, a set of transformed image regions are prepared from every segmented object. Next, different color and texture features are extracted from these regions and stored in a database. The features of each object are then matched against the features of all other objects using a nearest neighbor strategy with several dissimilarity measures. At last, we evaluate the percentage of correctly matched features for each object class.

3.1 Region Preparation

In the region preparation step, we use different object-background modifications, segmentation accuracies and bounding boxes to transform object segments into regions for semi-local feature extraction. In the following, these region preparation methods are explained and their effects on the resulting feature properties are discussed.

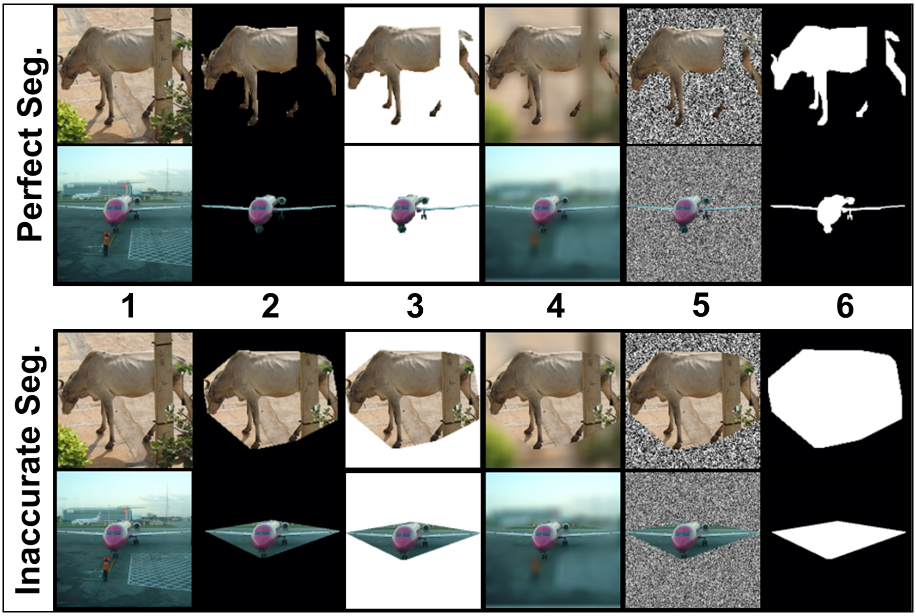


Fig. 2. Region preparation techniques from perfect (top row) and inaccurate segments (bottom row), shown for the cow of Fig. 1 and the plane of Fig. 3. The columns correspond to the regions explained in Table 1.

Object-Background Modifications. We use six different modification techniques, shown in the columns of Fig. 2 and in the rows of Table 1. Region 1 is equivalent to bounding boxes without segmentation. No focus is set to specific properties of the object in these regions. In the opposite, shape is the only attribute left to describe in Region 6. In Region 2 and Region 3 black and white backgrounds are used. These regions set the focus to the object shape and its content (texture and color). Region 4 keeps the characteristics of the original background although the object is focused and the object boundaries are sharpened. We use Gaussian smoothing to blur the background of these regions heavily. The Gaussian noise of Region 5 also sets focus to the object but with fewer weighting of the object shape. In preliminary experiments, we have tested further object-background modifications (e.g. object boundary expansion) but the six selected ones performed best.

Table 1. Object-background modifications. The focus of each region and the properties of the resulting semi-local features reflect these modifications.

Region	*Object*	*Background*	*Focus*
Region 1	original	original	none
Region 2	original	black	shape & object
Region 3	original	white	shape & object
Region 4	original	blurred	object & background
Region 5	original	Gaussian noise	object
Region 6	white	black	shape

Bounding Boxes. Most image features are extracted from square image regions. However, segmented objects are given as arbitrarily shaped polygons or image masks, and thus we operate on bounding boxes around such object segments. As shown in Fig. 3, we select two different bounding boxes for each object. First, we use tight, rectangular bounding boxes that touch the segment bounds on all four sides. These regions are resized to squares in a pre-processing step. Secondly, we use square bounding boxes that touch the object bounds only in the larger dimension. These regions contain larger parts of the object's background but no additional resize step changes the aspect ratio of these regions.

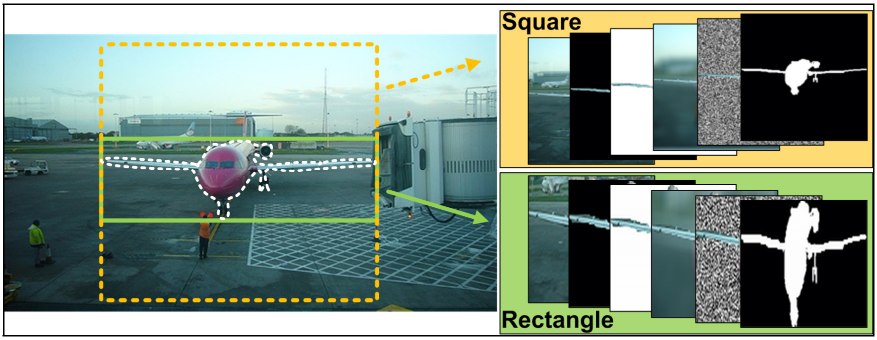


Fig. 3. Bounding boxes. Two different bounding boxes are used for region preparation from each segmented object (left). The square bounding box includes more background but does not change the aspect ratio of the resulting regions (right).

Segmentation Accuracy. As shown in Fig. 2, we use two different segmentation accuracies. On the one hand, perfect segmentations are given from the Pascal VOC dataset [5]. The object pixels are thereby used as foreground and all others are used as background. On the other hand, we simulate an inaccurate segmentation using the convex hull of all pixels that belong to a perfectly segmented object. No holes are retained in this approach but the actual object shape is heavily changed. In the case study of this work, we further use an automatic object segmentation approach. For these experiments no information about the segmentation accuracy is given.

3.2 Image Features

Four popular texture and color features are used in the experiments: SIFT, Gabor wavelets, MPEG-7 ColorLayout and ScalableColor. We omit to add specific shape features because the used texture features extracted from Region 6 (white object on black background) already present effective shape features.

Texture Features. *SIFT* features [15] consist of 8-dimensional orientation histograms that are computed from the image gradients in 16 slightly overlapping sub-regions on a 4x4 grid. They are normalized to increase the robustness against color and illumination changes. In the proposed semi-local feature approach, we extract only one SIFT feature from the entire object region without interest point detection. *Gabor wavelets* [23] are computed with a bank of orientation and scale sensitive Gabor filters. We use the mean and standard deviation of each filter output as final feature values.

MPEG-7 Color Features. *ColorLayout* [22] presents the spatial distribution of colors in a very compact form. They cluster an image or an image region into sub-regions of 8x8 pixels and compute the average pixel value for each of them. Finally, the first low frequency coefficients of a discrete cosine transform are selected. *ScalableColor* features [22] use a quantized HSV color histogram to build a scalable binary tree from their indexed probability values before a discrete Haar transformation is applied. The resulting features are scale invariant.

3.3 Object Detection

We compute the nearest neighbor for the segmented objects using all described region preparation techniques and feature types independently. Thereby, each segmented query object is matched against all other segmented objects in the dataset. The object

Table 2. Dissimilarity measures used to classify semi-local features: *n* specifies the dimension of feature *a* and *b*

Measure	*Formula*
Minkowski Family Distances	$\left(\sum_{i=1}^{n} \lvert a_i - b_i \rvert^p\right)^{\frac{1}{p}}, \; p = \left\{\frac{1}{2}, 1, 2\right\}$
Cosine-Based Dissimilarity	$1 - \frac{\sum_{i=1}^{n}(a_i * b_i)}{\sqrt{\sum_{i=1}^{n} a_i^2} * \sqrt{\sum_{i=1}^{n} b_i^2}}$
Canberra Metric	$\sum_{i=1}^{n} \frac{\lvert a_i - b_i \rvert}{\lvert a_i \rvert + \lvert b_i \rvert}$
Chi Square Statistics	$\sum_{i=1}^{n} \frac{(a_i - m_i)^2}{m_i}, \; m_i = \frac{a_i + b_i}{2}$
Jeffrey Divergence	$\sum_{i=1}^{n}\left(a_i \log \frac{a_i}{m_i} + b_i \log \frac{b_i}{m_i}\right)$

class of the nearest neighbor is then used to determine the class of a query object. We perform this nearest neighbor classification with following dissimilarity measures to get as general findings as possible.

The dissimilarity measures of Table 2 have been chosen according to their high performance for image retrieval with global features in [24]. We believe that more sophisticated classification approaches can be used to achieve better detection results, but it is out of the scope of this work to identify the best classification strategies. Instead, we try to perform a fair comparison between the proposed feature extraction techniques and want to show how these features can be used to improve existing detection systems.

3.4 Implementation

Most object detection systems consist of heterogeneous components and they are deeply integrated into their application workflow. This makes it difficult to alter specific components of these systems if changes are required for some reason. In contrast to this practice, we use the configurable object recognition infrastructure CORI [25] to enable the interchangeability of different segmentation approaches, region preparation methods, visual features, and dissimilarity measures. On the one hand, CORI facilitates the development of these components in a reusable way, independent from specific tasks. On the other hand, new processing chains can be arranged with simple configurations by the selection of desired components and their parameters.

In this work, we focused on the development of two novel CORI components: a segmentation wrapper and a region preparer. The segmentation wrapper operates on the output images of typical segmentation approaches instead of supporting only one single approach. As shown in Fig. 4, these output images contain each segmented region in a different color for both, image and object segmentation. The only difference is that object segmentation approaches (right image) only segment those regions that probably belong to an object while non-object pixels are black and object boundary pixels are shown in white. For the experiments of this work, perfect object segmentation images are used and inaccurate segmentations are simulated in a further preprocessing step. However, we also wanted to support a fully automated object detection workflow. Thus, we implemented the segmentation wrapper in a way that it is able to execute various segmentation approaches as an external process. Currently, this works for every segmentation approach that is executable from the command line with the arguments *input image directory* and *output directory*. Eventually, the segmentation wrapper returns the bounding box (square or rectangle) and the pixel mask of each segment.

The region preparer uses the original image and the segmentation wrapper results as input in order to generate the proposed image regions for semi-local features generation. In this implementation, we first generate a new image with the size of the bounding box of this segment and fill it with the background of the current region, see Table 1. The smoothing of Region 4 is thereby applied by convolution with a Gaussian mask using the Intel Performance Primitives. After this step, each pixel of the new image that is given in the region mask is replaced with the original pixel from the input image or with a white pixel for Region 6, respectively. Finally, we resize the new image to a fixed size of 64x64 pixels for the computation of color and texture features.

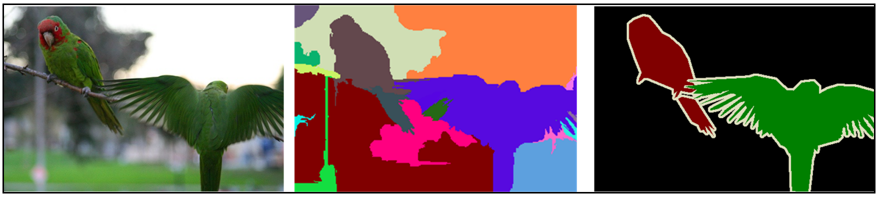


Fig. 4. Segmentation output: The input image (left) is transformed into a set of image regions (middle) or to segmented objects with black background and white boarder pixels (right)

4 Evaluation

In the experiments, two different evaluation strategies are used. On the one hand, we computed the recall of correctly classified objects for each object class and for all classes combined. On the other hand, we did a precision-at-k evaluation to count the number of query objects with at least one correct match in the top k entries (k = 1-10). Afterwards, we performed an initial case study to investigate semi-local features in combination with automatic image segmentation approaches.

4.1 Dataset

We used the open Pascal VOC 2010 segmentation dataset [5] for all experiments. In this dataset, 20 different object classes (see x-axis of Fig. 5) are perfectly segmented in 1928 Flickr images. The ground-truth contains a total number of 4203 objects whereby several object classes occur more often than other ones. For instance, 928 persons and 108 dining tables are given. All images are provided with JPEG encoding and a longer dimension side of 500 pixels.

Table 3. Overall recall (in %) for perfect and inaccurate segmentation

		R.1	***R.2***	***R.3***	***R.4***	***R.5***	***R.6***
Perfect Seg	**SIFT**	25.0	38.3	40.4	32.0	29.8	**46.5**
	GW	20.5	37.2	39.9	21.0	31.5	**45.0**
	CL	15.4	22.4	23.6	19.6	15.0	**28.7**
	SC	16.4	**21.8**	21.4	21.6	16.5	-
Inacc. Seg.	**SIFT**	25.0	27.2	**27.5**	22.5	27.2	12.1
	GW	20.5	**25.3**	24.8	19.1	25.1	10.8
	CL	15.4	16.8	**18.6**	17.9	15.2	15.1
	SC	16.4	16.5	**16.8**	16.5	15.8	-

4.2 Results

The results are organized according to following aspects: the suitability of semi-local features for object detection; the role of region preparation, segmentation accuracy, used image feature types, and dissimilarity measures. Fig. 5 and Table 3 are used to illuminate these points. Both show the achieved recall of a nearest neighbor classification with Jeffrey divergence on squared bounding boxes.

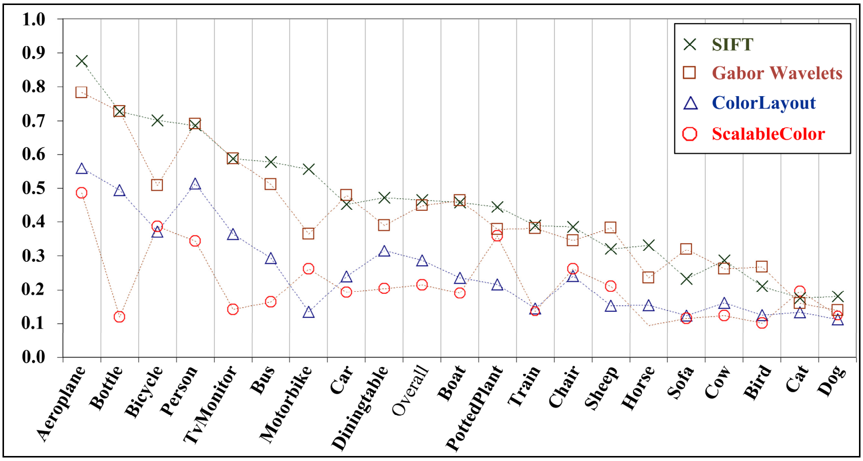


Fig. 5. Recall per object class from perfectly segmented objects. For each feature type the results of the best region are shown. The object classes are sorted according to their highest result from left to right.

Semi-local Features. Fig. 5 shows that the recall rates of the best matching object classes are significantly above 50% for texture features. Furthermore, the results of all objects are clearly above random classification (5%) independent of the used feature type. The fact that all 4-legged animals (sheep, horse, cow, cat, dog) are below the average, indicates that inter-class similarities decrease their classification. As shown in Table 3, the highest overall recall of 46.5% was achieved with SIFT features from perfectly segmented Region 6. Moreover, 80% of all objects have at least one correct match within the first 10 retrieved objects for the same configuration. These results clearly indicate that semi-local features are able to facilitate the detection of accurately segmented objects.

Region Preparation. Table 3 shows that texture features achieved the best results on Region 6 (white foreground on black background) where only shape information is given. This is also true for most object classes. MPEG-7 color descriptors generally perform best with original objects on black and white background (Regions 2 and 3). These regions are also the best choice for texture features when no accurate segmentation is given. At the first glance, white background outperforms black background on the given dataset but the results of the precision-at-k did not verify this assumption. Moreover, square bounding boxes always achieved better results than rectangle bounding boxes for SIFT and MPEG-7 features by an average increase of 2%. This indicates that the effect of changing the object's aspect ratio is worse than using a larger amount of background. However, for Gabor wavelets no significant changes have been measured between square bounding boxes and rectangle ones.

Segmentation Accuracy. In order to simulate inaccurate segmentations from the given test set, we used the convex hull around perfectly segmented objects. Table 3 shows the classification results of perfectly and inaccurately segmented objects. These results indicate that accurate segmentation can improve the classification significantly (up to +24.5%) when the region is prepared appropriately. In contrast, only smaller improvements of about 2% are achieved between unmodified regions (Region 1) and modified ones for inaccurate segmentation. Only the results of Gabor wavelets improve from 20.5% to 25.3% and 24.8% for black and white backgrounds. Region 6 performs worse than all other regions for inaccurate segmentation because these regions only contain very rough object contours, as shown in Fig. 2.

Feature Types. The performance of SIFT and Gabor wavelets is similar for both segmentation accuracies and all regions except Region 1 and Region 4 where the background is left unmodified and blurred, respectively. Gabor wavelets perform slightly better on rectangular bounding boxes while SIFT achieves better results on square regions. MPEG-7 ColorLayout and ScalableColor features perform worse than texture features for the given task. Although Fig. 5 indicates that ColorLayout outperforms ScalableColor this is only true because the best performing region preparation approach (Region 6) is not applicable for pure color features (ScalableColor) where no spatial information is used.

Dissimilarity Measures. The difference between the best and the worst dissimilarity measure for all features is about 3-5%. For instance, the results of SIFT features for Region 6 on perfect segmentations lie between 46.5% for the best (Jeffrey divergence) and 42.4% for the worst measure (Canberra metric). The highest variations are caused by MPEG-7 ScalableColor features. It seems that the ranking of dissimilarity measures does not depend on the used region preparation technique because the results of all measures are similarly ordered for all techniques. The best dissimilarity measure for all features was Jeffrey divergence followed by Chi-Squared statistics. The worst measure was Fractional distance for all features followed by Canberra metric for texture features. L1 metric performed best of the Minkowski family measures, especially for texture features where the difference to L2 distance was above 2.5%.

4.3 Case Study

In addition to the evaluation based on perfect object segmentations, we performed a case study with the automatic image segmentation approach of [26]. This case study does not aim at the execution of an entire object detection workflow but it tries to discover potential challenges in the combination of non-perfect segmentation approaches with semi-local features. We intentionally selected a segmentation approach that is not amongst the top teams of the Pascal VOC segmentation challenge (compare [5] and [18]) because it adds every image pixel to an image region without region classification. As a consequence, we get three different segment types to challenge the proposed semi-local feature approach. The first segment type only captures parts of an object or the entire object. The second type captures only non-object parts while both are captured by the third type, object parts and image parts that do not belong to this object. The third image of Fig. 6 shows these segment types in gray, red, and yellow.

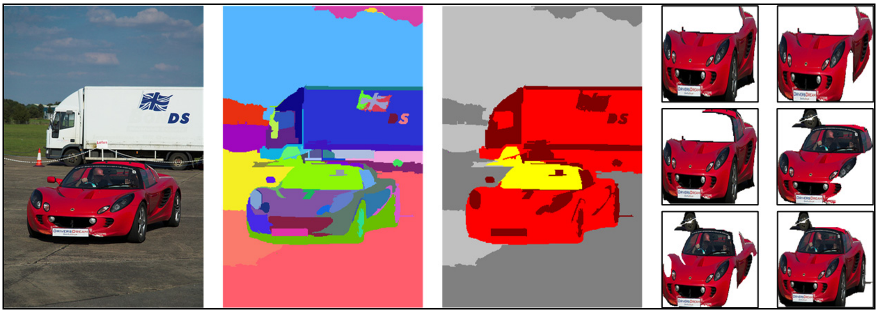


Fig. 6. Semi-local features from image segments of [26]. (a) original image, (b) image segments, (c) segment types: gray contains only background, red contains only object parts, yellow contains both, (d) Region 3 examples from combined segments.

Experimental Setup. The case study is done with images of the Pascal VOC test set including several instances of all 20 objects. In a first experiment, we used the default parameters of the automatic segmentation approach and extracted semi-local features for each segment. In a second experiment, we executed the segmentation several times for each image with different parameters. In this process, we got a couple of overlapping segments for each object similar to the multi-segmentation approaches that are explained in the related work section. First, we performed a manual inspection of the resulting semi-local feature regions, compare Fig. 2. Then, we extracted the semi-local features of each image region for all proposed region preparation techniques and performed a nearest neighbour search against the same features of all residual regions in the test set. In this process, we did another manual inspection of the matching regions. Note that no precision or recall values are given for this case study due to the small dataset size.

Observations. The image segments often capture the object boundaries accurately on some sites but seldom on all sites at the same time. Semi-local features that are extracted from these partially accurate regions do not often match with features of the same object. It only works if the rough object size and aspect ratio is preserved by the object segment whereby the results of Region 4 are the best ones. Sophisticated matching or region preparation techniques are required to improve this performance. Thus, we experimented with the combination of neighbouring segments to one semi-local feature, as shown on the right side of Fig 6. Improved performance pays the price of increased complexity when more segments are combined to one semi-local feature and we counted up to a few hundred segment combinations per image. Furthermore, we observed that the segmentation robustness of specific object parts is good. For instance, the wheels of cars and busses were regularly segmented as individual regions. Semi-local texture features of Region 6 seems to be good candidates for object detection with these object parts. The last observations consider the missing orientation invariance of the proposed semi-local features. If a large test set is given and the most common perspectives of each object are learned, orientation invariance is not important. Otherwise it is reasonable to rotate the image regions to their dominant orientation before feature extraction to gain rotation invariance similar to [15].

Discussion. From the object detection view, two different challenges arise from these observations. The first challenge is it to figure out which segment (or combination of segments) captures a trained object and which ones only capture background. The second challenge is which segment captures an object best if many overlapping hypotheses are given. Obviously, this second challenge mainly arises if we roughly know where in an image we should search for the object. This kind of information might stem from other object detection cues, such as a sliding window BoF approach. In order to tackle these challenges, we can either use perfectly segmented objects or automatically segmented objects to train the object detection system. In the first case, only accurately segmented objects would result in correct matches. However, perfectly segmented training examples seem to be the appropriate choice to identify the best segmentation of overlapping hypotheses, like the ones in the right of Fig. 6. In the second case, even partially accurate segmentations can lead to correct object detection if they are segmented similarly in training and test images.

5 Conclusions and Future Work

In this work, we have proposed semi-local features for object detection using segmentation as a pre-processing step. In this approach, state-of-the-art texture and color features are extracted from regions that cover the entire object with and without background modifications. Results of an extensive evaluation indicate that semi-local features are good candidates to improve object detection systems. The experiments investigated perfect segmentations and inaccurate ones, on the one hand, and automatically segmented image regions, on the other hand. The classification was done with a nearest neighbor matching strategy and different dissimilarity measures to keep the evaluation as simple and universally valid as possible.

In the evaluation, we have first shown that it does matter how the regions of segmented objects are prepared for semi-local feature extraction. Regions with modified objects and backgrounds can improve the overall classification rate significantly compared to unmodified regions, especially for accurately segmented objects. Secondly, square bounding boxes achieves better results than tight, rectangular bounding boxes. Thirdly, texture features perform better than color features and improvements of a few percent can be achieved when the right dissimilarity measures are chosen. The Jeffrey divergence and Chi-Square correlation performed best for all feature types and region preparation techniques.

For future work, we plan to investigate semi-local features in an integrated object detection system and on larger test sets of real-world images and videos. In this process, we want to evaluate the impact of different segmentation approaches and the compatibility of semi-local features with best practice object detection approaches. Furthermore, it might be interesting to apply semi-local features also for other computer vision tasks, for instance, in the area of robot navigation.

Acknowledgements. The author would like to thank Horst Eidenberger for his feedback and support. The research leading to this publication has received funding from the Austrian FIT-IT project ‘IV-ART – Intelligent Video Annotation and Retrieval Techniques’.

References

1. Pantofaru, C., Schmid, C., Hebert, M.: Object Recognition by Integrating Multiple Image Segmentations. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 481–494. Springer, Heidelberg (2008)
2. Li, F., Carreira, J., Sminchisescu, C.: Object recognition as ranking holistic Figure-ground hypotheses. In: CVPR (2007)
3. Rabinovich, A., Vedaldi, A., Belongie, S.: Does image segmentation improve object categorization? Tech. Rep. CS2007-090 (2007)
4. Russell, B., Freeman, W., Efros, A., Sivic, J., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
5. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes (VOC) challenge. IJCV (2010)
6. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. PAMI (2005)
7. Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., Van Gool, L.: A comparison of affine region detectors. IJCV (2005)
8. Van de Sande, K., Gevers, T., Snoek, C.: Evaluating color descriptors for object and scene recognition. PAMI (2010)
9. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
10. Lampert, C., Blaschko, M., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: CVPR (2008)
11. Oliva, A., Torralba, A.: Building the GIST of a Scene: The Role of Global Image Features in Recognition. Visual Perception, Progress in Brain Research (2006)
12. Hoiem, D., Efros, A., Hebert, M.: Geometric context from a single image. In: ICCV (2005)
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
14. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV (2008)
15. Lowe, D.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
16. Toshev, A., Taskar, B., Daniilidis, K.: Object detection via boundary structure segmentation. In: CVPR (2010)
17. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: CVPR (1997)
18. Carreira, J., Sminchisescu, C.: Constrained parametric min cuts for automatic object segmentation. In: CVPR (2010)
19. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI (2002)
20. Hoiem, D., Stein, A., Efros, A., Hebert, M.: Recovering occlusion boundaries. IJCV (2011)
21. Csurka, G., Perronnin, F.: An efficient approach to semantic segmentation. IJCV (2010)
22. Frigo, M., Johnson, S.: The design and implementation of FFTW3. In: Proc. Program Generation, Optimization, and Platform Adaptation (2005)
23. Manjunath, B., Ohm, J.-R., Vasudevan, V., Yamada, A.: Color and texture descriptors. Trans. on Circuits and Systems for Video Technology (2001)
24. Liu, H., Song, D., Rüger, S.M., Hu, R., Uren, V.S.: Comparing Dissimilarity Measures for Content-Based Image Retrieval. In: Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., Zhou, G. (eds.) AIRS 2008. LNCS, vol. 4993, pp. 44–50. Springer, Heidelberg (2008)
25. Sorschag, R.: CORI: A configurable object recognition infrastructure. In: International Conference on Signal and Image Processing Applications (2011)
26. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)

Inferring Time Varying Dynamic Linear Model Parameters from a Mixture Model

Kevin R. Keane and Jason J. Corso

Department of Computer Science and Engineering,
University at Buffalo, The State University of New York, Buffalo, NY, U.S.A.
{krkeane,jcorso}@buffalo.edu
http://www.cse.buffalo.edu/

Abstract. Non-stationary time series are extremely challenging to model. We propose a Bayesian mixture model framework for obtaining time varying parameters for a dynamic linear model. We discuss on-line estimation of time varying DLM parameters by means of a dynamic mixture model composed of constant parameter DLMs. For time series with low signal-to-noise ratios, we propose a novel method of constructing model priors. We calculate model likelihoods by comparing forecast distributions with observed values. We utilize computationally efficient moment matching Gaussians to approximate exact mixtures of path dependent posterior densities. The effectiveness of our approach is illustrated by extracting insightful time varying parameters for an ETF returns model in a period spanning the 2008 financial crisis; and, by demonstrating the superior performance in a statistical arbitrage application.

Keywords: Bayesian inference, Dynamic linear models, Multi-process models, Statistical arbitrage.

1 Background

1.1 Linear Models

Linear models are utilitarian work horses in many domains of application. A model's linear relationship between a *regression vector* F_t and an *observed response* Y_t is expressed through coefficients of a *regression parameter vector* θ. Allowing an *error of fit* term ϵ_t, a linear regression model takes the form:

$$Y = F^{\mathsf{T}}\theta + \epsilon \quad , \tag{1}$$

where Y is a column vector of individual observations Y_t, F is a matrix with column vectors F_t corresponding to individual regression vectors, and ϵ a column vector of individual errors ϵ_t.

The vector Y and the matrix F are observed. The *ordinary least squares* ("OLS") estimate $\hat{\theta}$ of the regression parameter vector θ is [1]:

$$\hat{\theta} = \left(FF^{\mathsf{T}}\right)^{-1} FY \quad . \tag{2}$$

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 37–50.
DOI: 10.1007/978-3-642-36530-0_4

1.2 Stock Returns Example

In modeling the returns of an individual stock, we might believe that a stock's return is roughly a linear function of market return, industry return, and stock specific return. This could be expressed as a linear model in the form of (1) as follows:

$$r = F^{\mathsf{T}}\theta + \epsilon, \quad F = \begin{bmatrix} 1 \\ r_M \\ r_I \end{bmatrix}, \quad \theta = \begin{bmatrix} \alpha \\ \beta_M \\ \beta_I \end{bmatrix}, \tag{3}$$

where r represents the stock's return, r_M is the market return, r_I is the industry return, α is a stock specific return component, β_M is the sensitivity of the stock to market return, and β_I is the sensitivity of the stock to it's industry return.

1.3 Dynamic Linear Models

Ordinary least squares, as defined in (2), yields a single estimate $\hat{\theta}$ of the regression parameter vector θ for the entire data set. Problems arise with this framework if we don't have a *finite* data set, but rather an *infinite* data stream. We might expect θ, the coefficients of a linear relationship, to vary slightly over time $\theta_t \approx \theta_{t+1}$. This motivates the introduction of *dynamic linear models* [2]. DLMs are a generalized form, subsuming Kalman filters [3], flexible least squares [4], linear dynamical systems [5,6], and several time series methods — Holt's point predictor, exponentially weighted moving averages, Brown's exponentially weighted regression, and Box-Jenkins autoregressive integrated moving average models [2]. The regime switching model in [7] may be expressed as a DLM, specifying an autoregressive model where evolution variance is zero except at times of regime change.

1.4 Contributions and Paper Structure

The remainder of the paper is organized as follows. In section §2, we introduce DLMs in further detail; discuss updating estimated model parameter distributions upon arrival of incremental data; show how forecast distributions and forecast errors may be used to evaluate candidate models; the generation of data given a DLM specification; inference as to which model was the likely generator of the observed data; and, a simple example of model inference using synthetic data with known parameters. Building upon this base, in section §3 multi-process mixture models are introduced. We report design challenges we tackled in implementing a mixture model for financial time series. In section §4, we introduce an alternative set of widely available financial time series permitting easier replication of the work in [8]; and we provide an example of applying a mixture model to real world financial data, extracting insightful time varying estimates of variance in an ETF returns model during the recent financial crisis. In section §5, we augment the statistical arbitrage strategy proposed in [8] by incorporating a hedge that significantly improves strategy performance. We demonstrate that an on-line dynamic mixture model outperforms all statically parameterized DLMs. Further, we draw attention to the fact that the period of unusually large mispricing identified by our mixture model coincides with unusually high profitability for the statistical arbitrage strategy. In §6, we conclude.

Algorithm 1. Updating a DLM given G, V, W.

```
Initialize $t = 0$
$\{$Initial information $p(\theta_0|D_0) \sim$ N$[m_0, C_0]\}$
Input: $m_0, C_0, G, V, W$
loop
  $t = t + 1$
  $\{$Compute prior at $t$: $p(\theta_t|D_{t-1}) \sim$ N$[a_t, R_t]\}$
    $a_t = Gm_{t-1}$
    $R_t = GC_{t-1}G^\mathsf{T} + W$
  Input: $F_t$
  $\{$Compute forecast at $t$: $p(Y_t|D_{t-1}) \sim$ N$[f_t, Q_t]\}$
    $f_t = F_t^\mathsf{T} a_t$
    $Q_t = F_t^\mathsf{T} R_t F_t + V$
  Input: $Y_t$
  $\{$Compute forecast error $e_t\}$
    $e_t = Y_t - f_t$
  $\{$Compute adaptive vector $A_t\}$
    $A_t = R_t F_t Q_t^{-1}$
  $\{$Compute posterior at $t$: $p(\theta_t|D_t) \sim$ N$[m_t, C_t]\}$
    $m_t = a_t + A_t e_t$
    $C_t = R_t - A_t Q_t A_t^\mathsf{T}$
end loop
```

2 Dynamic Linear Models

2.1 Specifying a DLM

In the framework of [2], a dynamic linear model is specified by its parameter quadruple $\{F_t, G, V, W\}$. DLMs are controlled by two key equations. One is the *observation equation*:

$$Y_t = F_t^\mathsf{T} \theta_t + \nu_t, \quad \nu_t \sim N(0, V) \quad , \tag{4}$$

the other is the *evolution equation*:

$$\theta_t = G\theta_{t-1} + \omega_t, \quad \omega_t \sim N(0, W) \quad . \tag{5}$$

F_t^T is a row in the *design matrix* representing independent variables effecting Y_t. G is the *evolution matrix*, capturing deterministic changes to θ, where $\theta_t \approx G\theta_{t-1}$. V is the *observational variance*, $\mathrm{Var}(\epsilon)$ in ordinary least squares. W is the *evolution variance matrix*, capturing random changes to θ, where $\theta_t = G\theta_{t-1} + w_t, \quad w_t \sim N(0, W)$. The two parameters G and W make a linear model *dynamic*.

2.2 Updating a DLM

The Bayesian nature of a DLM is evident in the careful accounting of sources of variation that generally increase system uncertainty; and, information in the form of incremental observations that generally decrease system uncertainty. A DLM starts with initial information, summarized by the parameters of a (frequently multivariate) normal distribution:

$$p\left(\theta_0 | D_0\right) \sim N\left(m_0, C_0\right) \quad . \tag{6}$$

At each time step, the information is augmented as follows:

$$D_t = \{Y_t, D_{t-1}\} \quad . \tag{7}$$

Algorithm 1 details the relatively simple steps of updating a DLM as additional regression vectors F_t and observations Y_t become available. Note that upon arrival of the current regression vector F_t, a one-step forecast distribution $p(Y_t|D_{t-1})$ is computed using the prior distribution $p(\theta_t|D_{t-1})$, the regression vector F_t, and the observation noise V.

2.3 Model Likelihood

The one-step forecast distribution facilitates computation of *model likelihood* by evaluation of the density of the one-step forecast distribution $p(Y_t|D_{t-1})$ for observation Y_t. The distribution $p(Y_t|D_{t-1})$ is explicitly a function of the previous periods information D_{t-1}; and, implicitly a function of static model parameters $\{G, V, W\}$ and model state determined by a series of updates resulting from the history D_{t-1}. Defining a model at time t as $M_t = \{G, V, W, D_{t-1}\}$, and explicitly displaying the M_t dependency in the one-step forecast distribution, we see that the one-step forecast distribution is equivalent to model likelihood[1]:

$$p\left(Y_t | D_{t-1}\right) = p\left(Y_t, D_{t-1} | D_{t-1}, M_t\right) = p\left(D_t | M_t\right) \tag{8}$$

Model likelihood, $p(D_t|M_t)$, will be an important input to our mixture model discussed below.

2.4 Generating Observations

Before delving into mixtures of DLMs, we illustrate the effect of varying the evolution variance W on the state variable θ in a very simple DLM. In Figure 1 we define three very simple DLMs, $\{1, 1, 1, W_i\}, W_i \in \{.0005, .05, 5\}$. The observations are from simple random walks, where the level of the series θ_t varies according to an evolution equation $\theta_t = \theta_{t-1} + \omega_t$, and the observation equation is $Y_t = \theta_t + \nu_t$. Compare the relative stability in the level of observations generated by the three models. Dramatic and interesting behavior materializes as W increases.

2.5 Model Inference

Figure 1 illustrated the difference in appearance of observations Y_t generated with different DLM parameters. In Figure 2, note that models with smaller evolution variance

[1] $D_t = \{Y_t, D_{t-1}\}$ by definition; M_t contains D_{t-1} by definition; and, $p(Y_t, D_{t-1}|D_{t-1}) = p(Y_t|D_{t-1})p(D_{t-1}|D_{t-1}) = p(Y_t|D_{t-1})$.

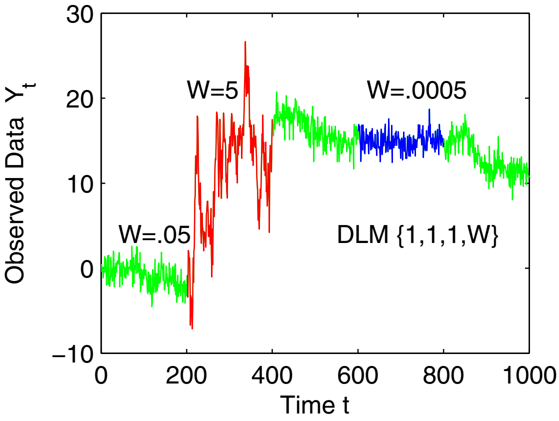


Fig. 1. Observations Y_t generated from a mixture of three DLMs. Discussion appears in §2.4

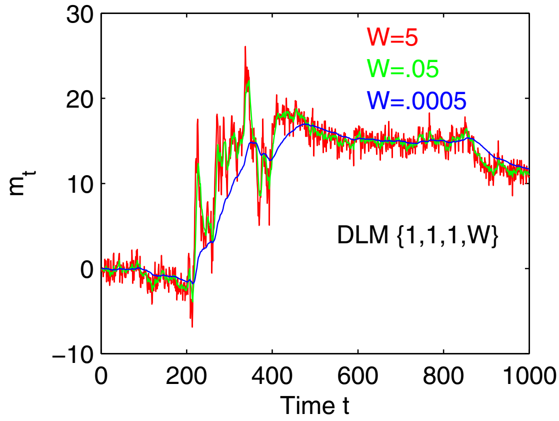


Fig. 2. Estimates of the mean of the state variable θ_t for three DLMs when processing generated data of Figure 1

W result in smoother estimates — at the expense of a delay in responding to changes in level. At the other end of the spectrum, large W permits rapid changes in estimates of θ — at the expense of smoothness. In terms of the model likelihood $p(D_t|M_t)$, if W is too small, the standardized forecast errors $e_t/\sqrt{Q_t}$ will be large in magnitude, and therefore model likelihood will be low. At the other extreme, if W is too large, the standardized forecast errors will appear small, but the model likelihood will be low now due to the diffuse forecast distribution.

In Figure 3, we graph the trailing interval log likelihoods for each of the three DLMs. We define trailing interval (k-period) likelihood as:

$$\begin{aligned} L_t(k) &= p(Y_t, Y_{t-1}, \ldots, Y_{t-k+1}|D_{t-k}) \\ &= p(Y_t|D_{t-1})p(Y_{t-1}|D_{t-2})\ldots \\ &\quad\; p(Y_{t-k+1}|D_{t-k}) \quad . \end{aligned} \tag{9}$$

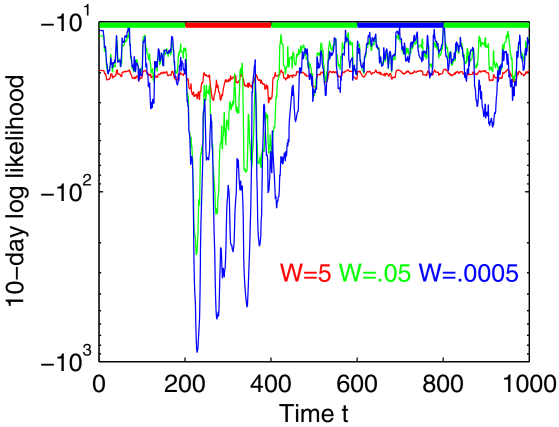


Fig. 3. Log likelihood of observed data during most recent 10 days given the parameters of three DLMs when processing generated data of Figure 1. Bold band at top of figure indicates the true generating DLM.

This concept is very similar to Bayes' factors discussed in [2], although we do not divide by the likelihood of an alternative model. Our trailing interval likelihood is also similar to the likelihood function discussed in [9]; but, we assume the errors e_t are not autocorrelated.

Across the top of Figure 3 appears a color code indicating the true model prevailing at time t. It is interesting to note when the likelihood of a model exceeds that of the true model. For instance, around the $t = 375$ mark, the model with the smallest evolution variance appears most likely. Reviewing Figure 2, the state estimates of DLM $\{1, 1, 1, W = .0005\}$ just happened to be in the right place at the right time. Due to the more concentrated forecast distributions $p(Y_t|D_{t-1})$ of this model, it briefly attains the highest trailing 10-period log likelihood. A similar occurrence can be seen for the DLM $\{1, 1, 1, W = .05\}$ around $t = 325$.

While the series on Figure 3 appear visually close at times, note the log scale. After converting back to normalized model probabilities, the favored model at a particular instance is more apparent as illustrated in Figure 4. In §5, we will perform model inference on the return series of exchange traded funds (ETFs).

3 Parameter Estimation

In §2, we casually discussed DLMs varying in parameterization. Generating observations from a specified DLM or combination of DLMs, as in §2.4, is trivial. The inverse problem, determining model parameters from observations is significantly more challenging. There are two distinct versions of this task based upon area of application. In the simpler case, the parameters are unknown but assumed constant. A number of methods are available for model identification in this case, both off-line and on-line. For example, [10] use E-M off-line, and [9] use the likelihood of a fixed-length trailing window of prediction errors on-line. Time varying parameters are significantly more

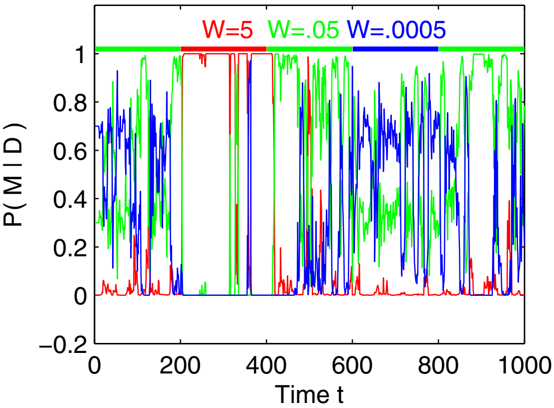


Fig. 4. Model probabilities from normalized likelihoods of observed data during most recent 10 periods. Bold band at top of figure indicates the true generating DLM.

challenging. The posterior distributions are path dependent and the number of paths is exponential in the length of the time series. Various approaches are invoked to obtain approximate solutions with reasonable computational effort. [2] approximate the posterior with a single Gaussian that matches the moments of the exact distribution. [11,12] propose variational Bayesian approximation. [13] discusses Gaussian-sum and assumed-density filters.

3.1 Multi-process Mixture Models

[2] define sets of DLMs, where the defining parameters $M_t = \{F, G, V, W\}_t$ are indexed by λ[2], so that $M_t = M(\lambda_t)$. The set of DLMs at time t is $\{M(\lambda_t) : \lambda_t \in \Lambda\}$. Two types of multi-process models are defined. A *class I multi-process model*, where for some unknown $\lambda_0 \in \Lambda, M(\lambda_0)$ holds for all t; and, a *class II multi-process model* for some unknown sequence $\lambda_t \in \Lambda, (t = 1, 2, \ldots), M(\lambda_t)$ holds at time t. We build our model in §4 in the framework of a class II mixture model. We do not expect to be able to specify parameters exactly or finitely. Instead, we specify a set of models that quantize a range of values. In the terminology of [12], we will create a *grid approximation* to the evolution and observation variance distributions.

Class II mixture models permit the specification of a model per time period, leading to a number of potential model sequences exponential in the steps, $|\Lambda|^T$. However, in the spirit of the localized nature of dynamic models and practicality, [2] exploit the fact that the value of information decreases quickly with time, and propose collapsing the paths and approximating common posterior distributions. In the filtering literature, this technique is referred to as the *interacting multiple model (IMM) estimator* [15, Ch. 11.6.6]. In our application, in §5, we limit our sequences to two steps, and

[2] [2] index the set of component models $\alpha \in \mathcal{A}$; however, by convention in finance, α refers to stock specific return, consistent with §1.2. To avoid confusion, we index the set of component models $\lambda \in \Lambda$, consistent with the notation of [14].

approximate common posterior distributions by collapsing individual paths based on the most recent two component models. To restate this briefly, we model two step sequences — the component model M_{t-1} just exited, and the component model M_t now occupied. Thus, we consider $|\Lambda|^2$ sequences. Reviewing Algorithm 1, the only information required from $t-1$ is captured in the collapsed approximate posterior distribution $p\left(\theta_{t-1}|D_{t-1}\right) \sim N\left(m_{t-1}, C_{t-1}\right)$ for each component model $\lambda_{t-1} \in \Lambda$ considered.

3.2 Specifying Model Priors

One key input to mixture models are the model priors. We have tried several approaches to this task before finding a method suitable for our statistical arbitrage modeling task in §5. The goal of our entire modeling process is to design a set of model priors $p(M(\lambda_t))$ and model likelihoods $p(D|M(\lambda_t))$ that yield in combination insightful model posterior distributions $p(M(\lambda_t)|D)$, permitting the computation of quantities of interest by summing over the model space $\lambda_t \in \Lambda$ at time t:

$$p(X_t|D_t) \propto \sum_{\lambda_t \in \Lambda} p(X_t|M(\lambda_t))p(M(\lambda_t)|D_t) \tag{10}$$

In the context of modeling ETF returns discussed in §5, the vastly different scales for the contribution of W and V to Q left our model likelihoods unresponsive to values of W. This unresponsiveness was due to the fact that parameter values W and V are of similar scale; however, a typical $|F_t|$ for this model is approximately 0.01, and therefore the respective contributions to the forecast variance $Q = F^{\mathsf{T}}RF + V = F^{\mathsf{T}}(GCG^{\mathsf{T}} + \mathbf{W})F + \mathbf{V}$ are of vastly different scales, 1 : 10,000. Specifically, density of the likelihood $p(Y_t|D_{t-1}) \sim N(f_t, Q_t)$ is practically constant for varying W after the scaling by 0.01^2. The only knob left for us to twist is that of the model priors.

DLMs with static parameters embed evidence of recent model relevance in their one-step forecast distributions. In contrast, mixture model component DLMs move forward in time from posterior distributions that mask model performance. The situation is similar to the game *best ball* in golf. After each player hits the ball, all players' balls are moved to a best position as a group. Analogously, when collapsing posterior distributions, sequences originating from different paths are approximated with a common posterior based upon end-point model. While some of us may appreciate obfuscation of our golf skills, the obfuscation of model performance is problematic. Due to the variance scaling issues of our application, the path collapsing, common posterior density approximating technique destroys the accumulation of evidence in one-step forecast distributions for specific DLM parameterizations $\lambda \in \Lambda$. In our current implementation, we retain local evidence of model effectiveness by running a parallel set of standalone (not mixed) DLMs. Thus, the total number of models maintained is $|\Lambda|^2 + |\Lambda|$, and the computational complexity remains asymptotically constant. In our mixture model, we define model priors proportional to trailing interval likelihoods from the standalone DLMs. This methodology locally preserves evidence for individual models as shown in Figure 3 and Figure 4.

The posterior distributions $p(\theta_t|D_t)_{M(\lambda)}$ emitted by identically parameterized standalone and component DLMs differ in general. A standalone constant parameter DLM computes the prior $p(\theta_t|D_{t-1})_{M(\lambda_t)}$ as outlined in Algorithm 1 using its own posterior $p(\theta_{t-1}|D_{t-1})_{M(\lambda_t=\lambda_{t-1})}$. In contrast, component DLMs compute prior distributions using a weighted posterior:

$$p(\theta_{t-1}|D_{t-1})_{M(\lambda_t)} = \sum_{\lambda_{t-1}} p(M(\lambda_{t-1})|M(\lambda_t))p(\theta_{t-1}|D_{t-1})_{M(\lambda_{t-1})} \quad . \tag{11}$$

4 A Financial Example

[8] proposed a model for the returns of the S&P 500 Index based upon the largest principal component of the underlying stock returns. In the form $Y = F^{\mathsf{T}}\theta + \epsilon$ used throughout this paper,

$$Y = r_{\text{s\&p}}, \quad F = r_{\text{pc1}}, \text{ and } \quad \theta = \beta_{\text{pc1}}. \tag{12}$$

The target and explanatory data in [8] spanned January 1997 to October 2005. We propose the use of two alternative price series that are very similar in nature; but, publicly available, widely disseminated, and tradeable. The proposed alternative to the S&P Index is the *SPDR S&P 500 ETF* (trading symbol SPY). SPY is an ETF designed to mimic the performance of the S&P 500 Index[16]. The proposed alternative to the largest principal component series is the *Rydex S&P Equal Weight ETF* (trading symbol RSP). RSP is an ETF designed to mimic the performance of the S&P Equal Weight Index [17]. While perhaps not as obvious a pairing as S&P Index / SPY, a first principal component typically is the mean of the data — in our context, the mean is the equal weighted returns of the stocks underlying the S&P 500 Index. SPY began trading at the end of January 1993. RSP began trading at the end of April 2003. We use the daily closing prices P_t to compute daily log returns:

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right) \quad . \tag{13}$$

Our analysis is based on the months during which both ETFs traded, May 2003 to present (August 2011).

The price levels, scaled to 100 on April 30, 2003 are shown in Figure 5. Visually assessing the price series, it appears the two ETFs have common directions of movement, with RSP displaying somewhat greater range than SPY. Paralleling the work of [8], we will model the return of SPY as a linear function of RSP, $Y = F^{\mathsf{T}}\theta + \epsilon$:

$$Y = r_{\text{spy}}, \quad F = r_{\text{rsp}}, \text{ and } \quad \theta = \beta_{\text{rsp}}. \tag{14}$$

We estimate the time varying regression parameter θ_t using a class II mixture model composed of 50 candidate models with parameters $\{F_t, 1, V, W\}$. $F_t = r_{\text{rsp}}$, the return of RSP, is common to all models. The observation variances are the values $V \times 10^6 \in \{\, 1,\ 2.15,\ 4.64,\ 10,\ 21.5,\ 46.4,\ 100,\ 215,\ 464,\ 1{,}000 \,\}$. The evolution variances are the values $W \times 10^6 \in \{\, 10,\ 56,\ 320,\ 1{,}800,\ 10{,}000 \,\}$. Our on-line process computes

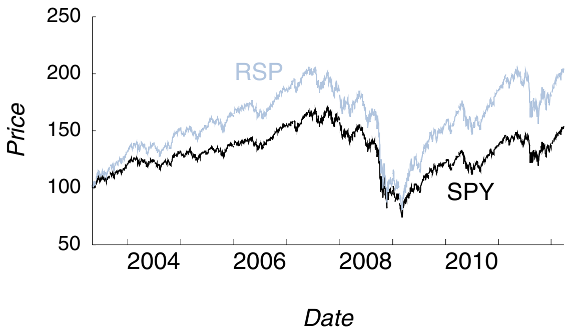


Fig. 5. SPDR S&P 500 (SPY) and Rydex S&P Equal Weight (RSP) ETF closing prices, scaled to April 30, 2003 = 100

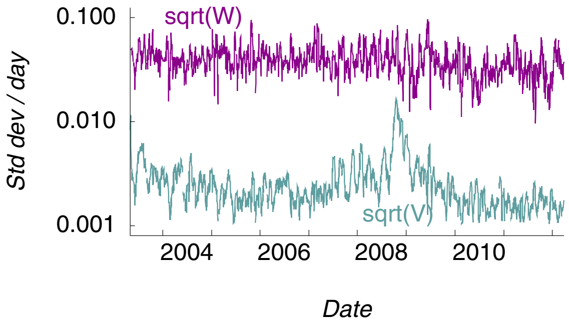


Fig. 6. The daily standard deviation of ν_t and ω_t as estimated by the mixture model. Observation noise $\nu_t \sim N(0, V)$; evolution noise $\omega_t \sim N(0, W)$.

$50^2 + 50 = 2550$ DLMs, 50^2 DLMs corresponding to the two-period model sequences, and 50 standalone DLMs required for trailing interval likelihoods. In the mixture model, the priors $p(M(\lambda_t))$ for component models $M(\lambda_t),\ \lambda_t \in \Lambda$, are proportional to trailing interval likelihoods (9) of corresponding identically parameterized standalone DLMs.

Subsequent to running the mixture model for the period May 2003 to present, we are able to review estimated time varying parameters V_t and W_t, as shown in Figure 6. This graph displays the standard deviation of observation and evolution noise, commonly referred to as volatility in the financial world. It is interesting to review the decomposition of this volatility. Whereas the relatively stationary series $\sqrt{W}$ in Figure 6 suggests the rate of evolution of θ_t is fairly constant across time; the observation variance V varies dramatically, rising noticeably during periods of financial stress in 2008 and 2009. The observation variance, or standard deviation as shown, may be interpreted as the end-of-day mispricing of SPY relative to RSP. In §5, we will demonstrate a trading strategy taking advantage of this mispricing. The increased observational variance at the end of 2008, visible in Figure 6 results in an increase in the rate of profitability of the statistical arbitrage application plainly visible in Figure 7. Conversely, the low observational variance beginning in 2010 to present (March 2012) in Figure 6 corresponds to a period of stagnation in the trading strategy in Figure 7.

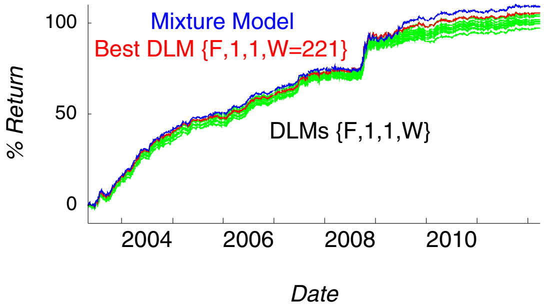


Fig. 7. Cumulative return of the various implementations of a statistical arbitrage strategy based upon a time varying mixture model and 10 constant parameter DLMs

5 Statistical Arbitrage

[8] describe an illustrative statistical arbitrage strategy. Their proposed strategy takes equal value trading positions opposite the sign of the most recently observed forecast error ϵ_{t-1}. In the terminology of this paper, they tested 11 constant parameter DLMs, with a parameterization variable δ equivalent to:

$$\delta = \frac{W}{W+V} \quad . \tag{15}$$

They note that this parameterization variable δ permits easy interpretation. With $\delta \approx 0$, results approach an ordinary least squares solution: $W = 0$ implies $\theta_t = \theta$. Alternatively, as δ moves from 0 towards 1, θ_t is increasingly permitted to vary.

Figure 6 challenges the concept that a constant specification of evolution and observation variance is appropriate for an ETF returns models. To explore the effectiveness of class II mixture models versus statically parameterized DLMs, we evaluated the performance of our mixture model against 10 constant parameter DLMs. We set $V = 1$ as did [8], and specified:

$$W \in \{29, 61, 86, 109, 139, 179, 221, 280, 412, 739\}\,.$$

These values correspond to the 5, 15, $\ldots 95\%$-tile values of W/V observed in our mixture model.

Figure 6 offers no justification of using $V = 1$. While the prior $p(\theta_t|D_{t-1})$, one-step $p(Y_t|D_{t-1})$ and posterior $p(\theta_t|D_t)$ “distributions” emitted by these DLMs will not be meaningful, the intent of such a formulation is to provide time varying *point estimates* of the state vector θ_t. The *distribution* of θ_t is not of interest to modelers applying this approach. In the context of the statistical arbitrage application considered here, the distribution is not required. The trading rule proposed is based on the sign of the forecast error; and, the forecast is a function of the prior mean a_t (a point estimate) for the state vector θ_t and observed values F_t and Y_t: $\quad \epsilon_t = Y_t - F_t^{\mathsf{T}} a_t$.

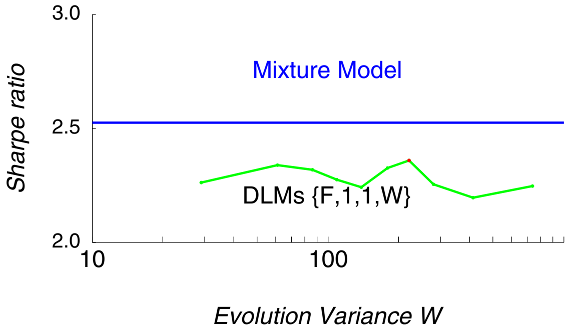


Fig. 8. Sharpe ratios realized by the time varying mixture model and 10 constant parameter DLMs

5.1 The Trading Strategy

Consistent with [8], we ignore trading and financing costs in this simplified experiment. Given the setup of constant absolute value SPY positions taken daily, we compute cumulative returns by summing the daily returns. The rule we implement is:

$$\texttt{portfolio}_t(\epsilon_{t-1}) = \begin{cases} +1 & \text{if } \epsilon_{t-1} \leq 0, \\ -1 & \text{if } \epsilon_{t-1} > 0. \end{cases} \tag{16}$$

where $\texttt{portfolio}_t = +1$ denotes a long SPY and short RSP position; $\texttt{portfolio}_t = -1$ denotes a short SPY and long RSP position. The SPY leg of the trade is of constant magnitude. The RSP leg is $-a_t \times$ SPY-value, where a_t is the mean of the prior distribution of θ_t, $p(\theta_t|D_{t-1}) \sim N(a_t, R_t)$; and, recall from (14) the interpretation of θ_t is the sensitivity of the returns of SPY Y_t to the returns of RSP F_t. Note that this strategy is a modification to [8] in that we hedge the S&P exposure with the equal weighted ETF, attempting to capture mispricings while eliminating market exposure. The realized Sharpe ratios appear dramatically higher in all cases than in [8], primarily attributable to the hedging of market exposure in our variant of a simplified arbitrage example. Montana et al. report Sharpe ratios in the 0.4 - 0.8 range; in this paper, after inclusion of the hedging technique, Sharpe ratios are in the 2.3 - 2.6 range.

5.2 Analysis of Results

We reiterate that we did not include transaction costs in this simple example. Had we done so, the results would be significantly diminished. With that said, we will review the relative performance of the models for the trading application.

In Figure 7, it is striking that all models do fairly well. The strategy holds positions based upon a comparison of the returns of two ETFs, one scaled by an estimate of $\beta_{\texttt{rsp},t}$. Apparently small variation in the estimates of the regression parameter are not of large consequence. Given the trading rule is based on the *sign* of the error ϵ_t, it appears that on many days, slight variation in the estimate of θ_t across DLMs does not result in a change to $\texttt{sign}(\epsilon_t)$. Figure 8 shows that over the interval studied, the mixture model provided a higher return per unit of risk, if only to a modest extent. What is worth mentioning is that the comparison we make is the *on-line* mixture model against the *ex post* best performance of all constant parameter models. Acknowledging this distinction, the mixture model's performance is more impressive.

6 Conclusions

Mixtures of dynamic linear models are a useful technology for modeling time series data. We show the ability of DLMs parameterized with time varying values to generate observations for complex dynamic processes. Using a mixture of DLMs, we extract time varying parameter estimates that offered insight to the returns process of the S&P 500 ETF during the financial crisis of 2008. Our *on-line* mixture model demonstrated superior performance compared to the *ex post* optimal component DLM in a statistical arbitrage application.

The contributions of this paper include the proposal of a method, trailing interval likelihood, for constructing component model prior probabilities. This technique facilitated successful modeling of time varying observational and evolution variance parameters, and captured model evidence not adequately conveyed in the one-step forecast distribution due to scaling issues. We proposed the use of two widely available time-series to facilitate easier replication and extension of the statistical arbitrage application proposed by [8]. Our addition of a hedge to the statistical arbitrage application from [8] resulted in dramatically improved Sharpe ratios.

We have only scratched the surface of the modeling possibilities with DLMs. The mixture model technique eliminates the burden of *a priori* specification of process parameters.

References

1. Johnson, R., Wichern, D.: Applied Multivariate Statistical Analysis. Prentice Hall, Upper Saddle River (2002)
2. West, M., Harrison, J.: Bayesian Forecasting and Dynamic Models. Springer-Verlag New York, Inc., New York (1997)
3. Kalman, R., et al.: A new approach to linear filtering and prediction problems. Journal of Basic Engineering 82, 35–45 (1960)
4. Kalaba, R., Tesfatsion, L.: A multicriteria approach to model specification and estimation. Computational Statistics & Data Analysis 21, 193–214 (1996)
5. Minka, T.: From hidden Markov models to linear dynamical systems. Technical Report 531, Vision and Modeling Group of Media Lab, MIT (1999)
6. Bishop, C.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer Science+Business Media, LLC, New York (2006)
7. Hamilton, J.: Time series analysis. Princeton University Press, Princeton (1994)
8. Montana, G., Triantafyllopoulos, K., Tsagaris, T.: Flexible least squares for temporal data mining and statistical arbitrage. Expert Systems with Applications 36, 2819–2830 (2009)
9. Crassidis, J., Cheng, Y.: Generalized Multiple-Model Adaptive Estimation Using an Autocorrelation Approach. In: 2006 9th International Conference on Information Fusion, pp. 1–8. IEEE (2007)

10. Ghahramani, Z., Hinton, G.: Parameter estimation for linear dynamical systems. Technical Report CRG-TR-96-2, University of Toronto (1996)
11. Valpola, H., Harva, M., Karhunen, J.: Hierarchical models of variance sources. Signal Processing 84, 267–282 (2004)
12. Sarkka, S., Nummenmaa, A.: Recursive noise adaptive Kalman filtering by variational Bayesian approximations. IEEE Transactions on Automatic Control 54, 596–600 (2009)
13. Minka, T.P.: Bayesian inference in dynamic models: an overview (2007), http://research.microsoft.com
14. Chen, R., Liu, J.: Mixture Kalman filters. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 62, 493–508 (2000)
15. Bar-Shalom, Y., Li, X., Kirubarajan, T., Wiley, J.: Estimation with applications to tracking and navigation. John Wiley & Sons, Inc. (2001)
16. PDR Services LLC: Prospectus. SPDR S&P 500 ETF (2010), https://www.spdrs.com
17. Rydex Distributors, LLC: Prospectus. Rydex S&P Equal Weight ETF (2010), http://www.rydex-sgi.com/

A Performance Evaluation of Mutual Information Estimators for Multivariate Feature Selection

Gauthier Doquire* and Michel Verleysen

Université catholique de Louvain, Machine Learning Group-ICTEAM
Place du Levant 3, 1348 Louvain-la-Neuve
{gauthier.doquire,michel.verleysen}@uclouvain.be
http://www.ucl.ac.be/mlg/

Abstract. Mutual information is one of the most popular criteria used in feature selection, for which many estimation techniques have been proposed. The large majority of them are based on probability density estimation and perform badly when faced to high-dimensional data, because of the *curse of dimensionality*. However, being able to evaluate robustly the mutual information between a subset of features and an output vector can be of great interest in feature selection. This is particularly the case when some features are only jointly redundant or relevant. In this paper, different mutual information estimators are compared according to important criteria for feature selection; the interest of a nearest neighbors-based estimator is shown.

Keywords: Mutual information estimation, Feature selection, Density estimation, Nearest neighbors.

1 Introduction

Nowadays, machine learning practitioners often have to deal with databases of very large dimension (containing data described by a lot of features). When considering a prediction task, all the features are not equally relevant to predict the desired output while some can be redundant; irrelevant or redundant features can increase the variance of the prediction models without reducing their bias while most of distance-based methods are quite sensitive to useless features. More generally, learning with high-dimensional data is a hard task due to the problems related to the *curse of dimensionality* [1].

Two main approaches exist to reduce the dimensionality of a data set. One solution is to project the data on a space of smaller dimension. Projections can be very effective but do not preserve the original features; this is a major drawback in many industrial or medical applications where interpretability is primordial. On the contrary, feature selection, by trying to find a subset of features with the largest prediction power, does allow such an interpretability.

Even if many ways of selecting features can be thought of, this paper focuses on filters. Filters are independent from the model used for prediction and thus do not require building any prediction model (including time-consuming learning and potential meta-parameters to tune by resampling methods). They are faster than wrappers which try

* Gauthier Doquire is funded by a grant from the Belgian F.R.I.A.

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 51–63.
DOI: 10.1007/978-3-642-36530-0_5

to find the best subset of features for a specific model through extensive simulations. Filters are often based on an information-theoretic criterion measuring the quality of a feature subset and a search procedure to find the subset of features maximising this criterion; the mutual information (MI) criterion [2] has proven to be very efficient for feature selection and has been used successfully for this task since many years (see e.g. [3,4]).

As it is not possible in practice to evaluate the MI between all the 2^{f-1} (f being the initial number of features) feature subsets and the output vector when f grows, incremental greedy procedures are frequently used, whose most popular ones are forward, backward or forward/backward. Such procedures are said to be *multivariate*, since they require the evaluation of the MI (or of another chosen criterion) directly between a set of features and the output vector. These methods have the advantage over bivariate ones such as ranking that they are able to detect subsets of features which are jointly relevant or redundant. Consider the XOR problem as a simple example; it consists in two features and an output scalar which is zero if both features have the same value and one otherwise. Individually each feature does not carry any information about the outptut; univariate procedures will never be able to detect them as relevant. However, when combined, the two features completely determine the output; when one is selected, a multivariate procedure will select the other one as relevant. A detailed introduction to the feature selection problem can be found in [5].

As will be seen, the MI generally cannot be computed analytically but has to be estimated from the data set. Even if this task has been widely studied, it remains very challenging for high-dimensional vectors. In this paper, it is shown how a MI estimator based on the principle of nearest neighbors (NN) outperforms traditional MI estimators with respect to three feature selection related criteria. This study is, to the best of our knowledge, the first one to compare MI estimators in such a context.

The rest of the paper is organized as follows. Section 2 briefly introduces the MI criterion and describes five of the most popular MI estimators. Section 3 presents the experiments carried out to compare these estimators and shows the results obtained on artificial and real-world data sets. Discussions and conclusions are given in Section 4.

2 Mutual Information

This section introduces the MI and presents the estimators used for comparison.

2.1 Definitions

Mutual information [2] is a symmetric measure of the dependence between two (groups of) random variables X and Y, assumed to be continuous in this paper. Its interest for feature selection comes mainly from the fact that MI is able to detect non-linear relationships between variables, whereas, as an example, it is not the case for the popular correlation coefficient which is limited to linear dependencies. Moreover, the MI can be naturally defined for groups of variables and is thus well-suited for multivariate search procedures. MI is defined as

$$I(X;Y) = H(X) + H(Y) - H(X,Y) \qquad (1)$$

where $H(X)$ is the entropy of X, defined for a continuous random variable as:

$$H(X) = -\int f_X(\zeta_X) \log f_X(\zeta_X)\, d\zeta_X. \tag{2}$$

In this last equation, f_X is the probability density function (pdf) of X. The MI can then be rewritten as

$$I(X;Y) = \int\int f_{X,Y}(\zeta_X, \zeta_Y) \log \frac{f_{X,Y}(\zeta_X, \zeta_Y)}{f_X(\zeta_X) f_Y(\zeta_Y)}\, d\zeta_X\, d\zeta_Y. \tag{3}$$

In practice, neither f_X, f_Y nor $f_{X,Y}$ are known for real-world problems; the MI has thus to be estimated.

2.2 Estimation

Plenty of methods have been proposed to estimate the MI. The great majority of them starts by estimating the pdf before plugging these results into Equation (1) or an equivalent expression. However, the dimension of X increases at each step of a forward feature selection procedure (or is already high at the beginning of a backward procedure) and most of these methods suffer dramatically from the curse of dimensionality [1]; they require an exponentially growing number of samples as the dimension of X grows while the number of available samples is in practice often limited. Such MI estimations do not thus seem well suited for feature selection. A NN-based MI estimator [6] avoiding the pdf estimation step has been used successfully in a feature selection context [7,3]. In the rest of this section, this estimator and four popular other ones are introduced.

The Basic Histogram. The histogram is one of the oldest and simplest ways to estimate a pdf. The basic idea is to divide the observation, prediction and joint spaces into non overlapping bins of fixed size and then to count the number of points falling in each of the bins. The entropy of X, Y and (X, Y) can be estimated using the discretized version of (2) and the estimation of the MI then naturally follows from (1). If histograms with bins of the same fixed size are considered, as it is the case in this paper, the size of the bins needs to be determined. Here, the approach by Sturges [8] will be followed: the number k of bins will be $\lceil 1 + \log_2(N) \rceil$, where N is the number of samples in the data set; other approaches could also be thought of [9].

The Kernel Estimator. The basic histogram suffers from many drawbacks. Among others, it is sensitive to the choice of the origin and to the size of the bins. In order to avoid sharp steps between the bins (and hence discontinuities), one can use the kernel density estimator (KDE) given by:

$$\hat{f}_X(x) = \frac{1}{Nh} \sum_{i=1}^{N} K(\frac{x - x_i}{h}), \tag{4}$$

where N is the number of observations in X, h is the window width and K is the kernel function required to integrate to one, leading $\hat{f}$ to be a probability density [10];

x_i denotes the i^{th} observation of the data set X. One possible choice for K is the Gaussian kernel, leading to the following density estimator:

$$\hat{f}(x) = \frac{1}{Nh\sqrt{2\pi}} \sum_{i=1}^{N} \exp(\frac{-(x-x_i)^2}{2h^2}). \tag{5}$$

In practice, the choice of the bandwidth h is fundamental. In this paper, the approach by Silverman [11] using a *rule of thumb* will be followed. It is often used as a good trade-off between performance and computational burden. The idea is to choose the width minimizing the asymptotic mean integrated square error (AMISE) between the estimation and the true density, assuming the underlying distribution is Gaussian. The resulting width is:

$$\hat{h}_{rot} \approx \sigma(\frac{4}{f+2})^{1/(f+4)}\ N^{-1/(f+4)} \tag{6}$$

where f is again the dimensionality of X. A large overview of different ways to select the kernel bandwidth is given in [12].

The B-splines Estimator. Another generalisation of the simple binning approach is given by the use of B-splines functions [13]. The idea is again to first discretize the X, Y and (X,Y) spaces. However, in this approach, the data points are allowed to be assigned to more than one bin a_i simultaneously in order to prevent the positions of the borders of the bins from affecting too much the estimation. The weights with which each point belongs to a bin are given by the B-spline functions $B_{i,k}$ (k being the spline order). Without getting too much into details, B-splines are recursively defined as:

$$\begin{aligned} B_{i,1}(x) &:= \begin{cases} 1 & \text{if } t_i \le x \le t_{i+1} \\ 0 & \text{otherwise} \end{cases} \\ B_{i,k}(x) &:= B_{i,k-1}(x)\frac{x-t_i}{t_{i+k-1}-t_i} + B_{i+1,k-1}(x)\frac{t_{i+k}-x}{t_{i+k}-t_{i+1}} \end{aligned} \tag{7}$$

where t is a knot vector defined for a number of bins M and a spline order $k = 1...M-1$ as:

$$t_i := \begin{cases} 0 & \text{if } i < k \\ i-k+1 & \text{if } k \le i \le M-1 \\ M-1-k+2 & \text{if } i > M-1 \end{cases} \tag{8}$$

To estimate the density $\hat{f}_x$, M_X weights $B_{i,k}(x_u)$ are determined for each datapoint x_u (where M_X is the number of bins in the X space). As the sum of the weights corresponding to each data point is 1, the sum of the mean values of each bin is also 1. The weights can thus be seen as the probability of each bin ($p(a_i) = \frac{1}{N}\sum_{u=1}^{N} B_{i,k}(x_u)$) and the entropy of the distribution can be estimated. The process is repeated for the Y space and for the joint (X,Y) space. The notion of B-splines can be extended to the multivariate case from univariate splines by the tensor produt construct. As an example, in two dimensions, the probability of a bin $a_{i,j}$ is given by $p(a_{i,j}) = \frac{1}{N}\sum_{u=1}^{N} B_{i,k}(x_u) \times B_{j,k}(y_u)$ where x denotes the first variable and y the second one.

The Adaptive Partition of the Observation Space. Darbellay and Vajda proved [14] that the MI can be approximated arbitrarily closely in probability by calculating relative frequencies on appropriate partitions. More precisely, they use an adaptive partitioning of the observation scheme, different from the traditional product partitions, to take into account the fact that with such basic partitions, much of the bins are not used to estimate the MI and can be replaced by fewer bins; they proved the weak consistency of the proposed method. Mathematical details can be found in [14]. In the rest of this paper, this methodology will be denoted *adaptive histogram*.

The Nearest Neighbors-Based or Kraskov Estimator. Since the hardest part when estimating the MI is the estimation of the underlying probability densities, another alternative is simply not to estimate densities and therefore directly estimating the MI by using NN statistics. The intuitive idea behind Kraskov's estimator [6] is that if the neighbors of a specific observation in the X space correspond to the same neighbors in the Y space, there must be a strong relationship between X and Y. More formally, the estimator is based on the Kozachenko-Leonenko estimator of entropy defined as:

$$\hat{H}(X) = -\psi(K) + \psi(n) + \log(c_d) + \frac{d}{N}\sum_{n=1}^{N} \log(\epsilon_X(N, K)) \tag{9}$$

where ψ is the digamma function, N the number of samples in X, d the dimensionality of these samples, c_d the volume of a d-dimensional unitary ball and $\epsilon_X(n, K)$, twice the distance (usually chosen as the Euclidean distance) from the n^{th} observation in X to its K^{th} NN. Two slightly different estimators are then derived whose most popular one is:

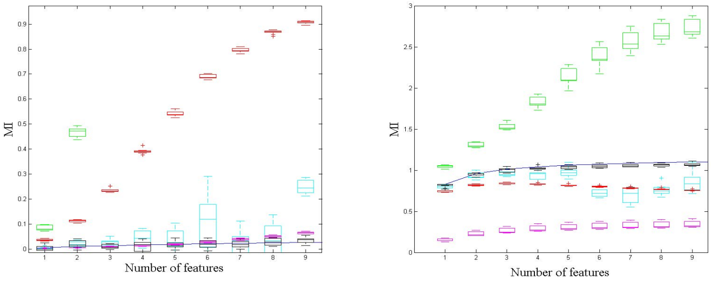
$$\hat{I}(X;Y) = \psi(N) + \psi(K) - \frac{1}{K} - \frac{1}{N}\sum_{i=1}^{N}(\psi(\tau_{x_i}) + \psi(\tau_{y_i})) \tag{10}$$

where τ_{x_i} is the number of points whose distance from x_i is not greater than $0.5 \times \epsilon(n, K) = 0.5 \times \max(\epsilon_X(n, K), \epsilon_Y(n, K))$. Avoiding the evaluation of pdf, the hope is to reach better results than with traditional estimators.

It is also important to note that other NN based density estimators have been proposed in the litterature; a recent example is [15]. As they are less popular than [6] for feature selection, they are not used in the present comparison.

3 Experiments

Three sets of experiments are carried out in this section. The objective is to assess the interest of the different estimators for *incremental feature selection algorithms*. The criteria of comparison and the experimental setup are thus very different from the ones used in previous papers only focused on MI estimation (see e.g. [16]). First, a suitable estimator should be accurate, i.e. it should reflect the true dependency between groups of features and increases (resp. decreases) when the dependance between groups of features increases (resp. decreases). Then it should also be able to detect uninformative features and return a value close to zero when two independent groups of features are



r=0.1

r=0.9

Fig. 1. Boxplots of the approximation of the MI for correlated Gaussian vectors by several estimators: the basic histogram (green), a KDE (red), an adaptive histogram (cyan), a NN-based estimator (black) and a B-splines estimator (magenta). The solid line represents the true MI.

given. Eventually, a good estimator should be quite independent from the value of its parameters or some fast heuristics to fix them should be available.

The implementation by A. Ihler has been used for KDE[1]. For the NN-based estimator, the parameter K is set to 6 unless stated otherwise. For the B-splines estimator, the degree of the splines is set to 3 and the number of bins to 3. These values correspond to those advised in the respective original papers [6,13].

3.1 Accuracy of the Estimators

The first set of experiments consists in comparing the precision of the MI estimators as the dimension of the data set increases. To this end, they will be used to estimate the MI between n correlated Gaussians $X_1 \ldots X_n$ with zero mean and unit variance. This way, the experimental results can be compared with exact analytical expressions as the MI for n such Gaussians is given by [14]:

$$I(X_1 \ldots X_n) = -0.5 \times \log[det(\sigma)] \tag{11}$$

where σ is the covariance matrix.

All the correlation coefficients are set to the same value r, chosen to be 0.1 and 0.9. The estimation is repeated 100 times on randomly generated datasets of 1000 instances; the results are shown for $n = 1 \ldots 9$. Even if this can be seen as a small number of dimensions, there are practical limitations when using splines and histogram-based estimators in higher dimensions. Indeed the generalization of the B-splines-based estimator to handle vectors of dimension d involves the tensor product of d univariate B-splines, a vector of size M^d, where M is the number of bins. Histogram-based methods are also limited in the same way since they require the storage of the value of k^d bins, where k is the number of bins per dimension. Nearest neighbors-based methods are not affected by this kind of problems and have only a less restrictive limitation regarding the number n of data points since they require the calculation of $O(n^2)$ pairwise distances. As will be

[1] http://www.ics.uci.edu/~ihler/code/

seen, the small number of dimensions used in the experiments is sufficient to underline the drawbacks and advantages of the compared estimators.

Figure 1 shows that, as far as the precision is concerned, Kraskov *et al.*'s estimator largely outperforms its competitors for the two values of r ($r = 0.1$ and $r = 0.9$). The estimated values are always very close to the true ones and show small variations along the 100 repetitions. The adaptive histogram provides on average accurate estimations up to dimension 8 and 6 for $r = 0.1$ and $r = 0.9$ respectively, with however very strong fluctuations observed accross the experiments. The B-spline estimator is also extremely accurate for the five first dimensions and $r = 0.1$. For $r = 0.9$ (and thus for higher values of MI), it severely underestimates the true values while the aspect of the true MI curve is preserved. This cannot be considered as a major drawback in a feature selection context where we are interested by the *comparison* of MI between groups of features. The results achieved by the kernel density estimator are very poor as soon as n exceeds 1, largely overestimating the true values for $r = 0.1$ while immediately decreasing for $r = 0.9$. Finally, as one could expect, the basic histogram produces the worst results; the estimated values are too high to be reported on Figure 1 for $r = 0.1$ when the dimension of the the data exceeds two.

3.2 Mutual Information between Independent Variables

In feature selection, a suitable estimator should assign a value close to zero to the MI between independent (groups of) variables. More precisely, one has to make sure that a greatly above zero value of MI is not the result of a weakness or a bias of the estimator but does correspond to a dependence between the variables. Moreover, as the MI is not bounded to a known interval, the relevance of each feature subset cannot be directly assessed based only on the value of the MI. A solution is to establish the relevance of a feature subset by looking if the MI between this subset and the outptut vector is significantly larger than the MI between this subset and a randomly permuted version of the output. It is thus important in practice to study how the MI between the actual data points and a randomly permuted objective vector is estimated. In theory, the MI estimated in this way should be 0 as no more relationship exists between the observations and the permuted output.

Experiments have been carried out on one artificial and two real-world data sets. The artificial problem is derived from [17] and is often used as a benchmark for feature selection algorithms. It has 10 input variables X_i uniformly distributed over $[0, 1]$ and an output variable Y given by $Y = 10\sin(X_1X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \epsilon$ where ϵ is a Gaussian noise with zero mean and unit variance. The sample size is 1000 and 100 data sets are randomly generated. As can be deducted easily, only the five first features are relevant to predict Y.

The first real data set is the well known Delve census data set, available from the University of Toronto[2] for which the 2048 first entries of the training set are kept. The dimension of the data set is 104. The second real data set is the Nitrogen data set[3], containing only 141 spectra discretized at 1050 different wavelengths. The goal is to

[2] http://www.idrc-chambersburg.org/index.html

[3] http://kerouac.pharm.uky.edu/asrg/cnirs/

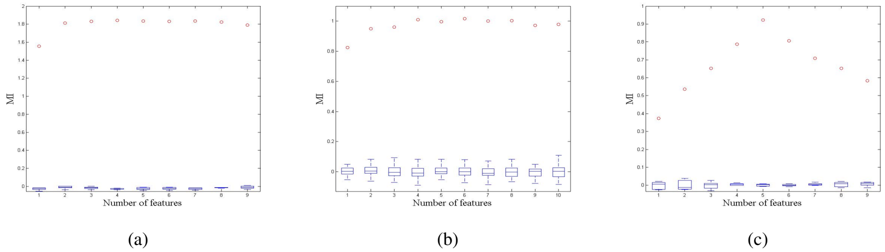


Fig. 2. Estimated MI between a group of features and the output (circles) and boxplots of the estimated MI between the same features and a permuted output for the NN-based estimator: (a) Delve dataset, (b) Nitrogen dataset, (c) Artificial dataset

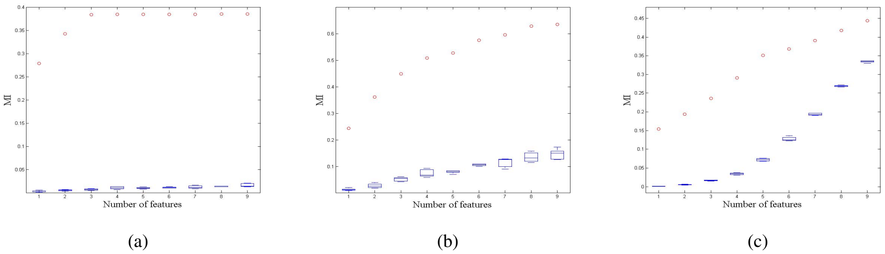


Fig. 3. Estimated MI between a group of features and the output (circles) and boxplots of the estimated MI between the same features and a permuted output for the B-splines density estimator: (a) Delve dataset, (b) Nitrogen dataset, (c) Artificial dataset

predict the Nitrogen content of a grass sample. As pre-processing, each spectrum is represented using its coordinates in a B-splines basis, in order to reduce the amount of features to a reasonable number of 105 ([18]). For each data set, a forward feature selection procedure using the NN-based estimator is conducted (since it performed the best in the previous section) and is halted when nine features have been selected. The MI is then estimated as well as the MI with the permuted output for 100 random permutations of the output and for each of the nine subsets of features of increasing dimension. The performance of the estimators is thus compared on the same sets of relevant features. In Figure 2, it can be seen that for the three problems, the NN-based estimator used with permuted output produces values very close to 0, even when working with few samples as for the Nitrogen data set (the variance is however larger in this case). This satisfactory observation is in good agreement with previous results found in [6] where the authors conjectured the fact that equation (10) is exact for independent variables, without proof of this result. Let us also notice two undesirable facts about the estimator. First it sometimes produces slightly negative values. Even if this has no theoretical justification [19], this can easily be dealt with in practice, by setting negative values to 0. Secondly, it can be seen that the MI decreases after the addition of some variables. Once again, this phenomenon is not theoretically founded [19] even if it has often been used as a stopping criterion in greedy feature selection algorithms.

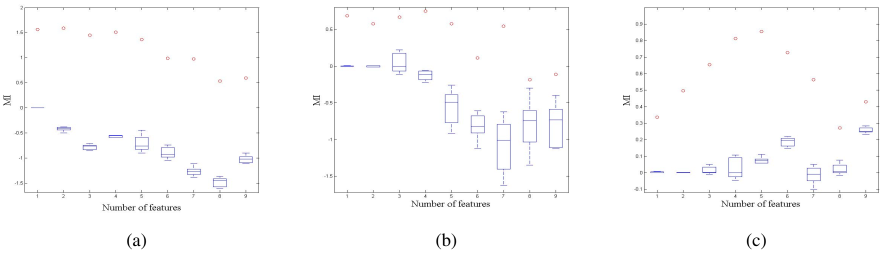
The B-splines estimator (Figure 3) also performs well on the Delve data set, while the results on the two other datasets contrast with this behaviour; regarding the artificial data set, the eight and nine first features have a higher MI with the permuted output than the first three with the actual output. This can be understood as the eight and nine first permuted features having a higher MI with the output than the three first original features have. This is a very undesirable fact in the context of feature selection, as it is obvious that permuted features do not carry any information about Y while the first three original ones actually do. The adaptative histogram (Figure 4) produces highly negative values for the Delve and the Nitrogen data sets. Even if the same *trick* as the one used for the Kraskov estimator could also be applied here (setting the negative values to 0), things are different. First, the absolute values of the negative results are very large, traducing instabilities of the algorithm as the dimension grows. Next, for the Nitrogen data set, the first, third and fourth features have a higher MI with the permuted output than the first eight and nine have with the actual output. For the artificial data set, the first nine features have a higher MI with the permuted output than the first six have with the true output.

The KDE (Figure 5) also returns values highly above 0 with the permuted output; on the artificial data set, the MI between the features and the actual or the permuted output becomes equal as the dimension increases. However, no confusion is possible for the two real-world data sets. Eventually, the histogram (Figure 6) shows dramatically incorrect results, with almost equal values for the MI between any subset of features and the permuted or the actual output; things are better for the Delve Census dataset.

3.3 Choice of the Parameters

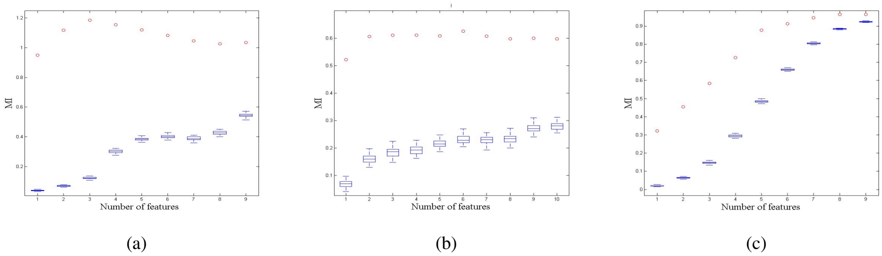
The last experiment is about the choice of the parameters in the estimators. As already mentioned, the basic histogram, the KDE, the B-splines approach and the NN-based estimator all have at least one parameter to fix, which can be fundamental for the quality of the estimations. Since the performances of the basic histogram in high-dimensional spaces are obviously dramatic, this estimator is not studied in more details. To compare the different estimators, the same data sets are used as in the previous section and the MI estimations are shown for dimension 2 to 5. Once again this limitation is due to the time and space-consuming generalization of the B-splines approach in high-dimensional spaces. Moreover, the choice of the parameter is less related to feature selection.

The Kernel Density Estimator. For the KDE, the parameter to be fixed is the kernel width. As an alternative to the *rule of thumb* used so far (see Equation (6)), two other methods are considered. The first one is the popular Least Squares Cross-Validation (LSCV) introduced by Rudemo and Bowman [20] [21] whose goal is to estimate the minimizer of the Integrated Square Error. The second one is the Plug-In method proposed by Hall, Sheater Jones and Marron [22]. Figure 7 shows the extreme sensitivity of the KDE to the width of the kernel since the results obtained with both bandwidth determination strategies are totally different for the three data sets. Moreover, as illustrated in Figure 8 showing the estimation of the MI for correlated Gaussians and $r = 0.9$, none of the methods used to set the kernel width clearly the other ones.



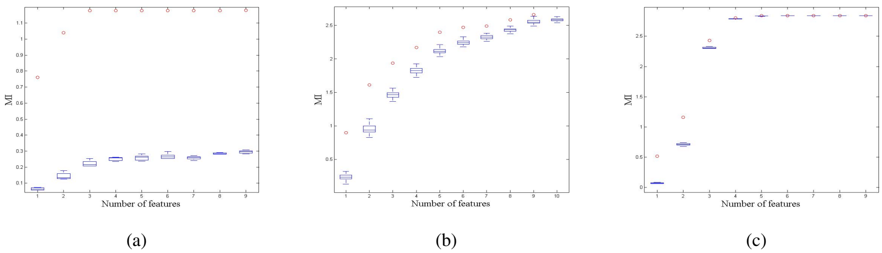
(a) (b) (c)

Fig. 4. Estimated MI between a group of features and the output (circles) and boxplots of the estimated MI between the same features and a permuted output for the adaptive histogram estimator: (a) Delve dataset, (b) Nitrogen dataset, (c) Artificial dataset



(a) (b) (c)

Fig. 5. Estimated MI between a group of features and the output (circles) and boxplots of the estimated MI between the same features and a permuted output for the kernel density estimator: (a) Delve dataset, (b) Nitrogen dataset, (c) Artificial dataset



(a) (b) (c)

Fig. 6. Estimated MI between a group of features and the output (circles) and boxplots of the estimated MI between the same features and a permuted output for the histogram based estimator: (a) Delve dataset, (b) Nitrogen dataset, (c) Artificial dataset

The B-splines Estimator. Two parameters have to be determined in this approach: the degree of the splines and the number of bins. We fix the degree of the splines to three (as suggested in the original paper) and only focus on the number of bins per dimension as this parameter has been shown to influence much more the output [13]; it will be taken between 2 and 5. Even if these values can seem surprisingly small, only three bins are used in [13]. The results presented in Figure 9 show that the estimated MI increases with the number of bins. These conclusions are consistent with those found in [13] for the one-dimensional case. However, even if the estimator is extremely sensitive to the number of bins, the relative values of the MI between the output and different groups

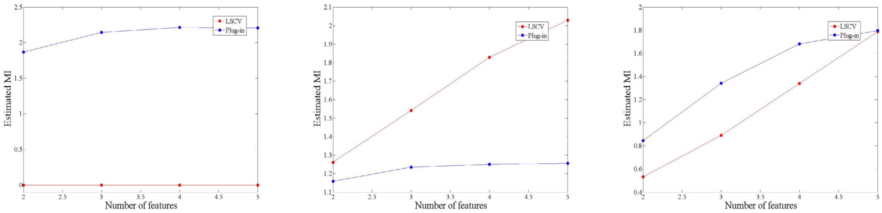


Fig. 7. Estimated MI with the kernel density estimator for different values of the kernel width: (a) Delve dataset, (b) Nitrogen dataset, (c) Artificial dataset

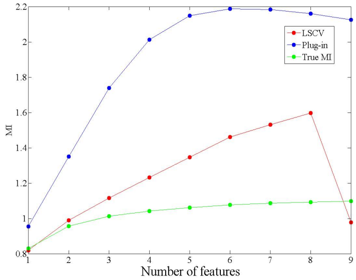


Fig. 8. Estimated MI for correlated Gaussians with a kernel density estimator whose kernel's width has been determined by three different procedures

of features is preserved, and so is the relative significance of the feature subsets. The sensitivity of the estimator is thus not a drawback for feature selection.

The Nearest Neighbors-Based Algorithm. The only parameter to fix in the NN-based estimator is the number of neighbors K. Kraskov *et al.* suggest a value of 6, arguing it leads to a good trade-off between the variance and the bias of the estimator [6]. Here, K is considered between 4 and 8.

Figure 10 shows very little sensitivity of the estimator in terms of absolute differences between estimations and thus a small sensitivity of the estimator to the number of neighbors used. However the results on the Delve data set indicate that even a small variation in the values of the estimated MI can lead to a different ranking of the features subsets in terms of relevance. As an example, in this data set, when using $K = 4$ or $K = 5$ neighbors, the subset of the five first features is less informative for the output than the subset of the four first features, while the opposite conclusion (which is in theory true) can be drawn when using 6, 7 or 8 neighbors. This is something that must be taken care of when performing feature selection because it could lead to the selection of irrelevant (or less relevant than other) features. One idea to overcome this issue is to average the estimations obtained within a reasonable range of values of K. In [23], this principle is applied to feature selection using a version of the Kraskov estimator adapted for classification problems. Another idea is to choose the value of K using the permutation test and resampling techniques [7].

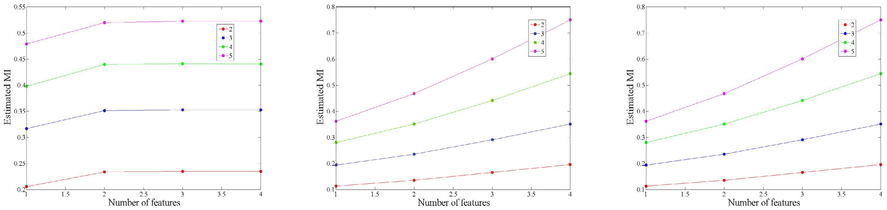


Fig. 9. Estimated MI with the B-splines estimator for different values of bins per dimension: (a) Delve dataset, (b) Nitrogen dataset, (c) Artificial dataset

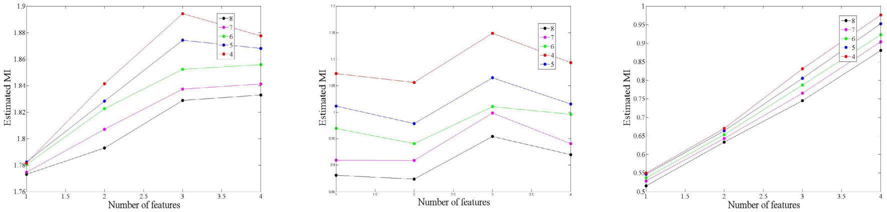


Fig. 10. Estimated MI with the NN-based estimator for different values of the parameter k: (a) Delve dataset, (b) Nitrogen dataset, (c) Artificial dataset

4 Conclusions and Discussions

In this paper, several approaches to the estimation of multi-dimensional MI are compared through three important criteria for feature selection: the accuracy, the consistency with an independence hypothesis and the sensitivity to the values of the parameter(s). The conclusion is the superiority of the NN-based algorithm which is by far the most accurate and the most consistent with an independent hypothesis (i.e. it returns values very close to 0 when estimating the MI between independent variables) on the three datasets used for comparison. The B-splines estimator presents interesting properties as well but can hardly be used when dimension becomes higher than 9 or 10, because of the exponential number of values to compute; the NN-based estimator is not affected by this major drawback, since it only requires the computation of the distances between each pair of points of the dataset. By avoiding the hazardous evaluation of high-dimensional pdf, it is able to produce very robust results as the dimension of the data increases. It is also the less sensitive to the value of its single parameter, the number of neighbors K. However, the choice of this parameter is important since slight variations in the estimation of the MI can lead to a different ranking of the feature subsets relevance. Being aware of all these facts, it thus appears to be a good choice to use the Kraskov estimator, or its counterpart for classification, to achieve MI-based multivariate filter feature selection.

References

1. Bellman, R.E.: Adaptive control processes - A guided tour. Princeton University Press (1961)
2. Shannon, C.E.: A mathematical Theory of Communication. Bell Syst. Tech. J. 27, 379–423, 623–656 (1948)
3. Rossi, F., Lendasse, A., François, D., Wertz, V., Verleysen, M.: Mutual Information for the Selection of Relevant Variables in Spectrometric Nonlinear Modelling. Chemometr. Intell. Lab. 80, 215–226 (2006)
4. Peng, H., Fuhui, L., Chris, D.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE T. Pattern Anal. 27, 1226–1238 (2005)
5. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. J. Mach. Lear. Res. 3, 1157–1182 (2003)
6. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating Mutual Information. Phys. Rev. E 69, 066138 (2004)
7. François, D., Rossi, F., Wertz, V., Verleysen, M.: Resampling Methods for Parameter-free and Robust Feature Selection with Mutual Information. Neurocomputing 70, 1276–1288 (2007)
8. Sturges, H.A.: The Choice of a Class Interval. J. Am. Stat. Assoc. 21, 65–66 (1926)
9. Scott, D.W.: On optimal and data-based histograms. Biometrika 66, 605–610 (1979)
10. Parzen, E.: On Estimation of a Probability Density Function and Mode. Ann. Math. Statist. 33, 1065–1076 (1962)
11. Silverman, B.W.: Density estimation for statistics and data analysis. Chapman and Hall, London (1986)
12. Turlach, B.A.: Bandwidth Selection in Kernel Density Estimation: A Review. CORE and Institut de Statistique, 23–493 (1993)
13. Daub, C., Steuer, R., Selbig, J., Kloska, S.: Estimating mutual information using B-spline functions - an improved similarity measure for analysing gene expression data. BMC Bioinformatics 5 (2004)
14. Darbellay, G.A., Vajda, I.: Estimation of the information by an adaptive partitioning of the observation space. IEEE T. Inform. Theory 45(4), 1315–1321 (1999)
15. Li, S., Mnatsakanov, R.M., Andrew, M.E.: k-Nearest Neighbor Based Consistent Entropy Estimation for Hyperspherical Distributions. Entropy 13, 650–667 (2011)
16. Walters-Williams, J., Li, Y.: Estimation of Mutual Information: A Survey. In: Wen, P., Li, Y., Polkowski, L., Yao, Y., Tsumoto, S., Wang, G. (eds.) RSKT 2009. LNCS, vol. 5589, pp. 389–396. Springer, Heidelberg (2009)
17. Friedman, J.H.: Multivariate Adaptive Regression Splines. Ann. Stat. 19, 1–67 (1991)
18. Rossi, F., Delannay, N., Conan-Guez, B., Verleysen, M.: Representation of functional data in neural networks. Neurocomputing 64, 183–210 (2005)
19. Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience (1981)
20. Bowman, A.W.: An alternative method of cross-validation for the smoothing of density estimates. Biometrika 71, 353–360 (1984)
21. Rudemo, M.: Empirical Choice of Histograms and Kernel Density Estimators. Scand. J. Stat. 9 (1982)
22. Hall, P., Sheater, S.J., Jones, M.C., Marron, J.S.: On optimal data-based bandwidth selection in kernel density estimation. Biometrika 78, 263–269 (1991)
23. Gomez-Verdejo, V., Verleysen, M., Fleury, J.: Information-Theoretic Feature Selection for Functional Data Classification. Neurocomputing 72, 3580–3589 (2009)

Modelling and Explaining Online News Preferences

Elena Hensinger, Ilias Flaounas, and Nello Cristianini

Intelligent Systems Laboratory, University of Bristol,
Merchant Venturers Building, Bristol, UK
{elena.hensinger,ilias.flaounas,nello.cristianini}@bristol.ac.uk

Abstract. We use Machine Learning techniques to model the reading preferences of audiences of 14 online news outlets. The models, describing the appeal of a given article to each audience, are formed by linear functions of word frequencies, and are obtained by comparing articles that became “Most Popular” on a given day in a given outlet with articles that did not. We make use of 2,432,148 such article pairs, collected over a period of over 1.5 years. Those models are shown to be predictive of user choices, and they are then used to compare both the audiences and the contents of various news outlets. In the first case, we find that there is a significant correlation between demographic profiles of audiences and their preferences. In the second case we find that content appeal is related both to writing style – with more sentimentally charged language being preferred, and to content with “Public Affairs” topics, such as “Finance” and “Politics”, being less preferred.

Keywords: Pattern Analysis, Ranking SVM, News Appeal, Text Analysis, User Preference Modelling, Prediction of user choices.

1 Introduction

Understanding the appeal of a given article to its potential readers is a vital question for journalists and editors, who need to select which articles to publish, particularly in a situation of intense competition among news media. But it is also useful to media analysts, who are interested in explaining the choices of editors: are they driven by pleasing their readers, or by other motives as well?

We use Machine Learning techniques to model the reading preferences of audiences of 14 online news outlets. The models describe the appeal of a given article to each audience, and they are formed by linear functions of word frequencies. Models are obtained by comparing articles that became “Most Popular” on a given day in a given outlet with articles that did not. We make use of 2,432,148 such article pairs, collected over a period of over 1.5 years, using our News Outlets Analysis & Monitoring (NOAM) system [9]. These models are shown to be predictive of reader choices, and they are used in various different ways to compare news outlets, as well as readers’ preferences. News outlets are compared both from the point of view of their readers, and from the point of view of their contents.

The comparison of readers of different outlets is performed by comparing the different user models generated, based on datasets from each particular outlet, and is found to be significantly correlated with demographic profiles of the readers of those outlets.

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 65–77.
DOI: 10.1007/978-3-642-36530-0_6

This investigation is only possible for the 11 outlets for which we had access to "Most Popular" articles as well as to demographic data.

The comparison of the contents of different outlets is done by assessing them through a combined model of reader preferences, and ranking them according to how appealing their contents are to an average reader. This comparison of contents can be extended to a larger number of outlets, since it only requires that we have a sample of the contents of those outlets. We find that there is a significant correlation between usage of sentimentally charged language and appeal to readers, as well as there being an anti-correlation between reader preferences and the coverage of "Public Affairs" stories, such as "Finance" or "Politics".

The success of this modelling approach is remarkable, if we consider that we ignore important clues to user choices, such as position in the web page, font size or accompanying images or media. We even ignore the full content of the articles, basing our models of user choices solely on the title and description of articles – the same limited information users will use to make their choices. Furthermore, we do not have access to the actual download figures corresponding to a given article. If we did, this would lead to a problem of linear regression. Instead, we frame the task as a problem of Learning to Rank, or preference learning, since we only have access to pairs of articles where one became popular and the other did not.

This work extends paper [17], where we built a set of six models and presented initial explorations of the models' applications. Here, we present models for 14 different audience groups, along with three approaches to explain why certain news articles are preferred to others, based on: demographics of readership, style of news articles, measured in sentiment-loaded words, and content of articles, in being about "Public" of "Non-Public" affairs.

Previous work in news analysis and readers' news preferences was mainly carried out by scholars of media studies and communication sciences: Research on identifying factors which influence choices of newspaper editors has been carried out since 1960s, ranging from what becomes news [13] to media bias, as in [12] in terms of liberal ideological views. More recently, online news moved also into focus of research, as for instance in [3]. They compared journalists choices of news to publish to audiences' choices of news to read, discovering discrepancies in these sets.

One main challenge in previous studies is the fact that data is collected, processed and analysed by hand by individual researchers. This poses limitations on the amount of data that can be assesseds and its interpretation. Automatic processing of news and readers' clicks has been realised in recent years, mostly aimed at resulting possibilities, for instance for news recommendations, as in [7].

In order to build user profiles, one has to acquire data about user preferences. Common approaches include to ask users about their preferences, or to collect click data. The first approach is more direct, but also more tedious and obtrusive for users. The second approach usually requires a log-in system to link user profiles and demographic information to user click choices, as in [21]. We explore a third approach which does not directly interfere with users, and which is based on click information, as published by the outlets themselves in news feeds of "Most Popular" stories. The drawback is that this information is not available for all outlets and there is not a fine-grained user segmentation.

In our previous work we explored such datasets with different techniques to model user preferences in terms of prediction performance and applications [15],[16],[17].

The paper is structured as follows: Sect. 2 focuses on creation of preference models: we present the theoretical framework to learn pairwise preference relations; the data used; and the resulting models' performances. Here, we also visualise models via word clouds. In Sect. 3, we introduce appeal computation for individual articles, and compare models based on the appeal they assign to a reference set of articles. This is compared to demographics of models. We work on explanations of appeal based on article style and content in Sect. 4, and we discuss the results of our work and conclude the paper in Sect. 5.

2 Modelling Reader Preferences

This section describes the theoretical framework of learning pairwise preference relations; the selection and preparation of the data we used in our experiments; and the resulting models, their prediction performance, and visualisation of their influential aspects. The key task is to model news preferences of different groups of audiences.

2.1 Ranking Pairs with Ranking SVM

Our modelling is based on two assumptions: a) news article preference is directly connected to the appeal of articles, and b) this appeal can be quantified via a linear utility function.

More formally, we say that an item x_i is preferred to x_j by notion $x_i \succ x_j$. The linear utility function $f : \mathbb{R}^n \to \mathbb{R}$ of the form $f(x) = \langle w, x \rangle + b$ captures this "better than" relationship via

$$x_i \succ x_j \iff f(x_i) > f(x_j) \iff \langle w, (x_i - x_j) \rangle > 0 \tag{1}$$

The last inequality is used to form constraints in the quadratic optimisation problem of Ranking SVM, which was introduced in [18] in the context of search engine queries. The approach builds upon the method for binary classification of SVM [4],[6].

In Ranking SVM, learning the relationship between two items x_i and x_j is expressed as a binary classification problem on the data item of their *difference* $x_{(i,j)} = x_i - x_j$. The class label y is determined via $\left\langle w, x_{(i,j)} \right\rangle$: if the value is greater or equal to 0, then $y_{(i,j)} = +1$, otherwise $y_{(i,j)} = -1$.

The optimisation problem for Ranking SVM for ℓ training data pairs of form $x_{(i,j)}$, with slack variables $\xi_{(i,j)}$ for non-linearly separable data is expressed, over all pairs $x_{(i,j)}$, as:

$$\underset{\xi, w}{\text{minimise}} \quad \langle w, w \rangle + C \sum_{x_{(i,j)}} \xi_{(i,j)} \tag{2}$$

$$\text{subject to} \qquad y_{(i,j)}(\langle w, x_{(i,j)} \rangle) \geq 1 - \xi_{(i,j)}, \tag{3}$$

$$\xi_{(i,j)} \geq 0 \; \forall \; x_{(i,j)} \tag{4}$$

The solution w is a parameters vector and can be used in two ways: to predict the preference relationship between two items x_i and x_j; and to compute the appeal score for an individual item x_i via $s(x_i) = \langle w, x_i \rangle$, where x_i is the vector space representation of the article.

We exploit both these properties: we learn preference models on pairwise data, and we quantify the appeal of individual items via their utility scores $s(x_i)$. For all our experiments, we used the implementation SVM^{rank} [19].

2.2 News Articles Dataset

The data used in our study comes from RSS (Really Simple Syndication) and Atom feeds, published by the different news outlets. A feed contains news articles in a structured format including a title, a short description and the publication date of each article.

Our approach to create preference data pairs, which are needed for the Ranking SVM technique, relies on two feeds:

(a) The "Top Stories" feed, carrying items published in the "Main Page" of an outlet.
(b) The "Most Popular" feed, which presents articles the readers found most interesting – by clicking on them in order to read them.

News Popularity. We define an item as *"Popular"* if it has been published in both the "Top Stories" and the "Most Popular" feeds, and as *"Non-Popular"* when it occurred in "Top Stories" but not in the "Most Popular" feeds. This captures the fact that both articles had the same starting conditions, but one of them was clicked on by the readers more often than the other. Such an approach allows now to pair up "Popular" with "Non-Popular" items, from same day and outlet, to serve as input to the Ranking SVM.

Data Sources. We use data from 14 English-writing news outlets to learn models, including online presences of newspapers, magazines and news wires, namely "BBC", "CBS", "CNN", "Forbes", "KSBW", "Los Angeles Times", "news.com.au", "New York Times", "NPR", "Reuters", "Seattle Times", "Time", "Wall Street Journal" and "Yahoo! News". All these outlets provide both the feeds "Top Stories" and "Most Popular" needed to create preference data pairs. We used 20 months of news articles between 1st December 2009 and 31st July 2011. Furthermore, we also used 579,805 articles published in the "Top Stories" feeds of 37 English-writing outlets between 1st June 2010 to 31st May 2011 as a reference set for appeal score computations.

Data Processing. News data was collected, pre-processed and managed via the News Outlets Analysis & Monitoring (NOAM) system [9]. For each article, we extracted its title and description, to imitate the snippet of text a user would see on a typical news outlet webpage. We applied standard text mining pre-processing techniques of stop word removal, stemming [24], and transfer into the bag-of-words (TF-IDF) space [20] – a standard representation in information retrieval and text categorisation [25]. The overall vocabulary was comprised from 179,238 words.

Train and Test Datasets. Data was divided into 18 months for training and 2 months for testing, in a 10-fold cross-validation setting. We excluded monthly datasets of small size, of which there were 3 for "BBC", "NPR" and "Wall Street Journal" and one for

“Forbes”. Average monthly pairs sizes are reported in Table 1. The overall number of articles pairs used was 2,432,148. The average number of non-zero word features per article’s title and description was 16.48.

Table 1. Average sizes of preference data pairs per month

Outlet	*Data pairs per month*
BBC	41,757
CBS	4,946
CNN	712
Forbes	5,577
KSBW	1,334
Los Angeles Times	1,941
New York Times	6,242
News.com.au	1,698
NPR	4,455
Reuters	3,675
Seattle Times	28,325
Time	3,004
Wall Street Journal	3,538
Yahoo! News	24,102

2.3 User Preference Models

Each of the 10 training sets per outlet led to one model, which were evaluated on the respective two months of testing data. Results for pairwise preference prediction accuracy, averaged over all datasets, are shown in Fig. 1, with error bars representing the standard error of the mean. Average performance over all models is 70.6%, and 77.2% for the five best performing models, showing that it is possible to model news preferences for the different audience groups with the proposed approach.

For all further work, we used one model per audience group: we computed cosine similarity between model vectors of one and the same outlet, along with the centroid of these models. The model most similar to the centroid was used as the representative.

Each model is a vector which assigns weights to terms that affect the appeal score of articles. We can visualise this information as a word cloud, focusing on the strongest 50 positively and negatively weighted terms, as shown in Fig. 2. The influential terms reflect the character of those two different outlets and the articles their audience prefer to read.

3 Comparing Models

This section shows how appeal scores can be used for comparison of models – and thus the audience group preferences the models embody. These results are then compared to demographics of audiences in an attempt to interpret our findings.

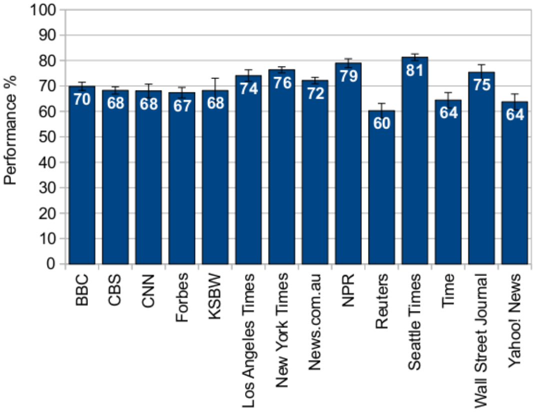
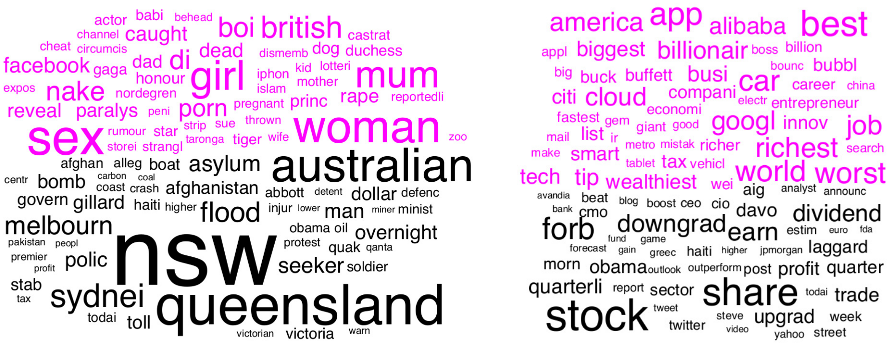


Fig. 1. Pairwise preference prediction accuracy for 14 models on 20 months of data. Error bars represent error of the mean. Average model performance is 70.6%.



(a) "News.com.au"

(b) "Forbes"

Fig. 2. The 50 strongest positive weighted terms (magenta) and 50 strongest negative weighted terms (black) in models for news preferences of audience of (a) "News.com.au" and (b) "Forbes"

A preference model w, applied to an input article x_i, produces an appeal score $s_w(x_i)$. We used a normalised version of the scoring function to exclude possible effects of article x_i's text length on the appeal score:

$$s_w(x_i) = \langle \frac{w}{||w||}, \frac{x_i}{||x_i||} \rangle \tag{5}$$

Appeal scores were computed for the reference set of 579,805 "Top Stories" articles from 37 outlets, with each of the 14 models. A distance matrix was created, based on the measure of $(1 - sample\ linear\ correlation)$, and multidimensional scaling was applied to enable a visualisation of the data in the first two resulting dimensions, as shown in Fig. 3.

This comparison shows similarities in what different audience groups find more or less appealing: with “Reuters” and “Wall Street Journal” readers on one side of the image; “Time” audience on top and “Yahoo! News” on the bottom, framing the space; and a cluster of five models showing similar appeal tendencies on the left-hand side. Other models are placed in-between.

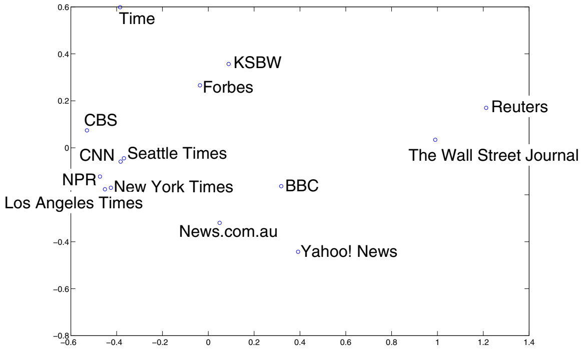


Fig. 3. Models’ distances, based on their appeal score assignments to 579,805 “Top Stories” articles in the reference set

Next, we aimed to explain these similarities by comparing them to other data about the audiences: their demographics. We acquired demographic information from `www.alexa.com` for 11 outlets, constructed of values for age groups, education categories, gender, parenthood, and ethnicity of the website’s audiences “relative to the general internet population”. Each audience was thus represented by a 25-dimensional demographics vector. As before, we computed correlation distances between audience data, with the resulting visualisation in Fig. 4.

In order to compare the two distance matrices, we used the Mantel test [22] implemented in the “Vegan” package of the statistical computation system “R” [23]. We use 10,000 permutations to produce the Mantel statistic r with Spearman’s rank correlation, which measures a monotonic relationship, finding r=0.451 with p-value=0.034.

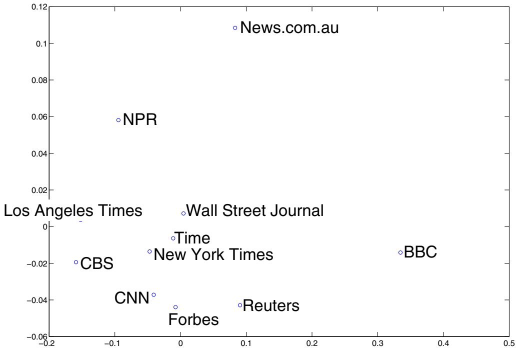


Fig. 4. Models’ distances, based on their online audiences’ demographics

4 Explaining News Appeal

Previous sections presented modelling of preferences, and a possible explanation based on audience demographics. This section presents exploration of appeal in relation to characteristics of the articles themselves, and their outlets. We introduce the notion of "Global appeal" and relate it to style of articles – via usage of sentiment-loaded words, and articles' contents – via their topics. In both cases, we find significant correlations, serving as explanations for articles being preferred by audiences.

In previous steps, we used the individual preference models to each compute an appeal score for every news item. In this section, we average the scores of all models per item to form a "global" appeal score, *i.e.* measuring how appealing an article is perceived by a general audience. Furthermore, we operate on the level of entire outlets which publish news by grouping articles by their publication source. Working on style and content of articles, we found title and description to be of insufficient data quantity, and thus used title, description and article content text for all work presented in this section.

4.1 Global Appeal and Linguistic Subjectivity of Outlets

Our first exploration focuses on the global appeal and its relation to the linguistic subjectivity of articles. This characteristic quantifies the usage of sentiment-loaded words. While in theory, a news article should be rather neutral in its selection of words and report only the facts, in reality outlets have the choice of wording news and grasping the attention of their readers by using either positively or negatively loaded words. Our measure of linguistic subjectivity focuses on adjectives as the strongest sentiment carriers [14], and it is defined as the ratio of adjectives with sentiment over the total number of adjectives in a text.

We computed linguistic subjectivity scores for articles in "Top Stories" feeds of a subset of 31 outlets in our reference set. Global appeal was computed with a subset of six models presented in [17]. We observe a strong and significant correlation (Pearson correlation coefficient = 0.6791, p-value <0.0001) between articles' global appeal scores and their linguistic subjectivity values, and we visualise all outlets in the two-dimensional space of appeal and linguistic subjectivity in Fig. 5. Both axis show a ranking of outlets by how appealing they are perceived (x-axis) and by their choice of language (y-axis). Both, global appeal and linguistic subjectivity are higher for UK tabloids and the online presence of the "People" magazine, which are positioned close to each other, and further apart from all other outlets. On the opposite directions, we can find the newswire "Reuters" and "BBC". Another observation is that "The Boston Globe", "The New York Times" and its international version "International Herald Tribune" – all assets of "The New York Times Company"[1] – have similar linguistic subjectivity and appeal.

[1] Source (Aug. 2011): http://www.nytco.com/company/index.html

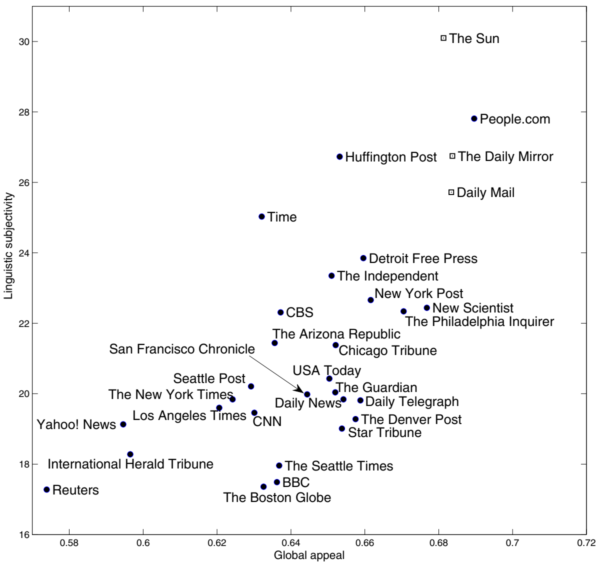


Fig. 5. Outlets in the space of global appeal and linguistic subjectivity of their "Top Stories" articles. UK tabloids, marked as rectangles, cluster together in both dimensions.

4.2 Appeal, "Public" and "Non-Public" Affairs

It is not only style of news that attracts reader's interest, but also what the news are about. In this part, we automatically detect topics of news articles and relate them to the articles' global appeal scores, showing that the broad themes of "Public Affairs" and "Non-Public Affairs" are related to article appeals.

For each article in our reference set of "Top Stories" from 37 outlets, we assign a topic score for a variety of topics based on SVM classifiers [10]. We then group the topics into two broad themes: "Public Affairs" and "Non-Public Affairs". Topics assigned to "Public Affairs" are *Elections, Inflation and Prices, Markets, Business, Politics* and *Petroleum*, topics of "Non-Public Affairs" are *Crime, Disasters, Fashion, Art, Environmental issues, Religion, Science, Sports, Travel* and *Weather*. The average of topic scores for the corresponding theme leads to two additional scores per each article.

In our first experiment, we look at the topics of the "Top Stories" news published by the different outlets. We use the topic classifiers to decide whether an article is of a certain topic or not. Averaging over topics for the two themes, we can assign 63% of articles over all outlets to belong to either "Public" and "Non-Public" affairs. We visualise in Fig. 6 the outlets in the space of their ratios of "Non-Public" to "Public" article themes, set against the global appeal of their articles.

As expected from previous results on global appeal, articles from the online presence of "People" score high in being most interesting for general audiences to read. They also show the highest ratio of topicalities (11.8), focusing strongly on "Non-Public Affairs" news. On the opposite along this axis, we find "The Wall Street Journal" with a ratio value of 0.12. Indeed, there are only two outlets, "Wall Street Journal" and "Reuters",

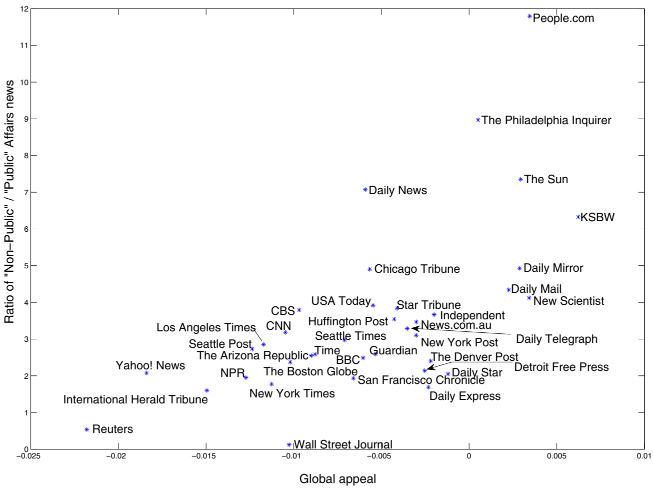


Fig. 6. Outlets in the space of global appeal and the ratios of "Non-Public" to "Public" themes of their published "Top Stories" articles

for which we can assign a majority of their articles to be of "Public" theme rather than of "Non-Public", resulting in their reported ratio being less than one.

If we do not threshold the output of the topics' SVM classifiers, we obtain real-valued topic scores. In our next experiment, we use the resulting "Public Affairs" and "Non-Public Affairs" values to compute correlation coefficients with the articles' global appeal scores. We look separately at articles from the different outlets in our reference set, leading to 37 sets of coefficients, as shown in Fig. 7. Over all outlets and for a general audience, we can observe a significant correlation of 0.28 between appeal of an article and its content being about "Non-Public Affairs", and a significant anti-correlation of -0.31 between appeal and "Public Affairs" themes (p-value < 0.001 in all cases).

The last experiment looks at the appeal scores assigned to articles by the 14 individual models, for all 579,805 articles in the reference set. We compute correlation coefficients between the vectors of articles' appeal scores, and their "Public" and "Non-Public" affairs scores – for each individual model. We find for all models a significant correlation (p-value < 0.001) between appeal scores and the news to be about "Non-Public Affairs", except for the "Reuters" model, which had p-value=0.2023. For 12 models, we find an anti-correlation between "Public Affairs" themes and appeal, and for the remaining two models "Wall Street Journal" and "Reuters", on the contrary, a correlation of 0.032 and 0.094, resp., with p-value < 0.001 in all cases. A visualisation of these results is given in Fig. 8.

The results of this Section show that there exists a connection between a news article's appeal and its style – as measured by linguistic subjectivity. In terms of content, by automatically assigning topic themes to news, we observe two general trends across news from all outlets: the higher the "Non-Public Affairs" score of an article, the more appealing it is perceived by a general audience (*i.e.* across all preference models), and the higher the "Non-Public Affairs" score, the less appealing. Broken up across the

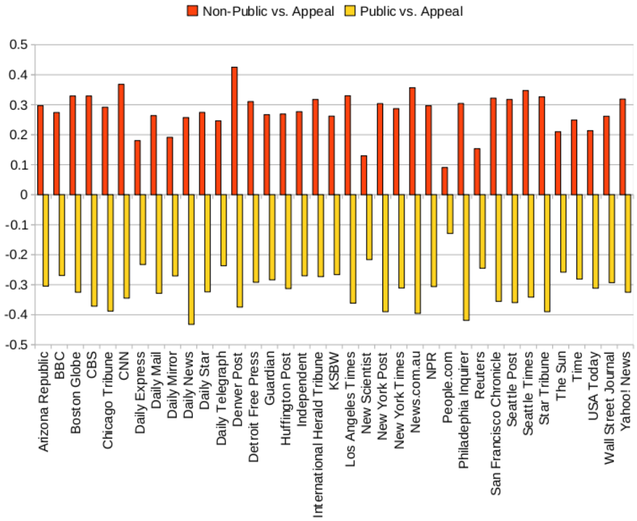


Fig. 7. Correlation coefficients between global appeal of "Top Stories" articles and their scores for "Public" and "Non-Public" Affairs, grouped by the 37 outlets

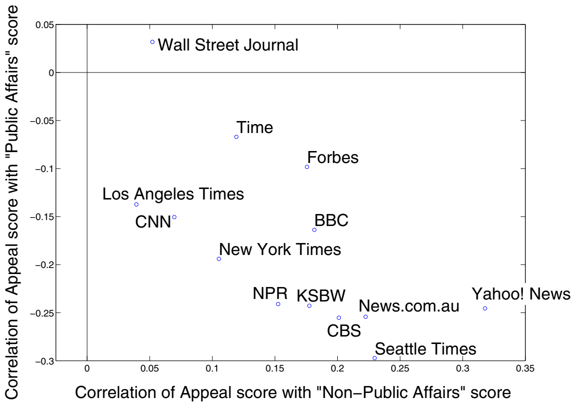


Fig. 8. Models in space of correlation coefficients of the appeal scores they assign to "Top Stories" articles from 37 outlets, and the articles' themes. For all modelled audiences, the "Non-Public Affairs" theme is correlated with appeal score.

individual models, we get a more detailed picture: all audiences perceive articles with higher "Non-Public Affairs" score as more appealing, and most of them, with the exception of two, perceive articles with higher "Public Affairs" score as less appealing.

5 Conclusions and Future Work

We show how Machine Learning approaches can be used for large-scale analysis of millions of news data items in order to model "What people prefer to read", and we present some explanations to the question of “What influences those preferences?”. Such analyses can be helpful for journalists and editors to understand what their readers prefer reading about and which words trigger the audience’s attention.

The data we used for modelling is limited in the following aspects: textual content, as published in news feeds, does not capture other factors that might influence a reader’s interest in an article, such as accompanying pictures or videos. However, we assume that text is the strongest information carrier, and thus the strongest factor for an article’s appeal. Furthermore, feed data contains no demographic information about the readers of specific articles, leading to a rather coarse-grained segmentation of users by their choice of outlet. Such an approach is known in marketing as “behavioural segmentation” [1]. Consequently, all users are treated as one homogeneous group with similar preferences. Our use of titles and descriptions of articles only – in order to mimic a realistic setting of browsing news websites – has an effect on the available amount of text, and thus non-zero features, per article. Given all these characteristics of our data, it is remarkable that it is still possible to reliably predict news preferences of audiences.

Avenues for future work include improvements of data used to understand influences on appeal, such as more extensive demographic information for all the outlets’ audiences. Other possible factors on appeal worth investigating, apart from linguistic subjectivity and topics, are geographic proximity of users to news, the presence of celebrities and other named entities in the text, or the reporting of scandals.

Acknowledgements. I. Flaounas and N. Cristianini are supported by the CompLACS project (European Community’s Seventh Framework Programme – grant agreement No. 231495); N. Cristianini was supported by a Royal Society Wolfson Research Merit Award; all authors are supported by Pascal2 Network of Excellence.

References

1. Assael, H., Roscoe Jr., A.M.: Approaches to Market Segmentation Analysis. The Journal of Marketing 40(4), 67–76 (1976)
2. Boczkowski, P.J., Mitchelstein, E.: Is There a Gap between the News Choices of Journalists and Consumers? A Relational and Dynamic Approach. The International Journal of Press/Politics 15(4), 420–440 (2010)
3. Boczkowski, P.J., Peer, L.: The Choice Gap: The Divergent Online News Preferences of Journalists and Consumers. Journal of Communication 61(5), 857–876 (2011)
4. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A Training algorithm for Optimal Margin Classifiers. In: Proceedings of the 5th Conference on Computational Learning Theory (COLT), pp. 144–152. ACM (1992)
5. Burgoon, J.K., Burgoon, M., Wilkinson, M.: Writing Style as a Predictor of Newspaper Readership, Satisfaction and Image. Journalism Quarterly 58, 225–231 (1981)
6. Cristianini, N., Shawe-Taylor, J.: An introduction to Support Vector Machines and other kernel-based learning methods. Cambridge University Press (2000)
7. Das, A.S., Datar, M., Garg, A., Rajaram, S.: Google news personalization: scalable online collaborative filtering. In: Proceedings of the 16th International Conference on World Wide Web (WWW), pp. 271–280. ACM (2007)

8. Flaounas, I.N., Turchi, M., Cristianini, N.: Detecting macro-patterns in the european mediasphere. In: Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and International Conference on Intelligent Agent Technology - Workshops, pp. 527–530. IEEE (2009)
9. Flaounas, I., Ali, O., Turchi, M., Snowsill, T., Nicart, F., De Bie, T., Cristianini, N.: NOAM: news outlets analysis and monitoring system. In: Proceedings of the 2011 International Conference on Management of Data (SIGMOD 2011), pp. 1275–1278. ACM (2011)
10. Flaounas, I.: Pattern Analysis of News Media Content. PhD thesis, University of Bristol (2011)
11. Flesch, R.: A New Readability Yardstick. Journal of Applied Psychology 32(3), 221–233 (1948)
12. Groseclose, T., Milyo, J.: A Measure of Media Bias. The Quarterly Journal of Economics 120(4), 1191–1237 (2005)
13. Harcup, T., O'Neill, D.: What is News? Galtung and Ruge revisited. Journalism Studies 2(2), 261–280 (2001)
14. Hatzivassiloglou, V., Wiebe, J.M.: Effects of adjective orientation and gradability on sentence subjectivity. In: Proceedings of the International Conference on Computational Linguistics (COLING), pp. 299–305. Morgan Kaufmann (2000)
15. Hensinger, E., Flaounas, I., Cristianini, N.: Learning the Preferences of News Readers with SVM and Lasso Ranking. In: Papadopoulos, H., Andreou, A.S., Bramer, M. (eds.) AIAI 2010. IFIP AICT, vol. 339, pp. 179–186. Springer, Heidelberg (2010)
16. Hensinger, E., Flaounas, I.N., Cristianini, N.: Learning Readers' News Preferences with Support Vector Machines. In: Dobnikar, A., Lotrič, U., Šter, B. (eds.) ICANNGA 2011, Part II. LNCS, vol. 6594, pp. 322–331. Springer, Heidelberg (2011)
17. Hensinger, E., Flaounas, I.N., Cristianini, N.: What Makes Us Click? - Modelling and Predicting the Appeal of News Articles. In: Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods (ICPRAM), pp. 41–50. SciTePress (2012)
18. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 133–142. ACM (2002)
19. Joachims, T.: Training linear SVMs in linear time. In: Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pp. 217–226. ACM (2006)
20. Liu, B.: Web Data Mining, Exploring Hyperlinks, Contents, and Usage Data. Springer (2007)
21. Liu, J., Dolan, P., Pedersen, E.R.: Personalized news recommendation based on click behavior. In: Proceedings of the 15th International Conference on Intelligent user Interfaces (IUI 2010), pp. 31–40. ACM (2010)
22. Mantel, N.: The Detection of Disease Clustering and a Generalized Regression Approach. Cancer Research 27, 209–220 (1967)
23. Oksanen, J., Blanchet, F.G., Kindt, R., Legendre, P., Minchin, P.R., O'Hara, R.B., Simpson, G.L., Solymos, P., Stevens, M.H.H., Wagner, H.: vegan: Community Ecology Package. R package version 2.0-3 (2012), http://CRAN.R-project.org/package=vegan
24. Porter, M.F.: An Algorithm for Suffix Stripping. Program 14, 130–137 (1980)
25. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM. 18,11, November, 613–620 (1975)
26. Sculley, D., Wachman, G.M.: Relaxed online SVMs for spam filtering. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 415–422. ACM (2007)

Combining Graph Seriation and Substructures Mining for Graph Recognition

Lorenzo Livi, Guido Del Vescovo, and Antonello Rizzi

SAPIENZA University of Rome,
Department of Information Engineering, Electronics and Telecommunications,
Via Eudossiana 18, 00184 Rome, Italy
{livi,delvescovo}@diet.uniroma1.it, antonello.rizzi@uniroma1.it

Abstract. Many interesting applications of Pattern Recognition techniques can take advantage in dealing with labeled graphs as input patterns. To this aim, the most important issue is the definition of a dissimilarity measure between graphs. In this paper, we outline an ensemble of methods for dealing with such data, focusing on two specific methods. The first one is simply based on a global alignment approach applied to seriated versions of the graphs. The second one is a two-stages method, which applies a recurrent substructures analysis to the seriated graphs, individuating a set of frequent subsequences, employed for embedding the graphs into a real valued feature vector space. Tests have been performed by synthetically generating a set of classification problem instances with increasing problem hardness, and with a shared benchmarking database of labeled graphs.

Keywords: Granular computing, Inexact graph matching, Sequence embedding, Graph embedding.

1 Introduction

Many recognition problems coming from interesting practical applications deal directly with *structured patterns*, such as images [7], audio/video signals [24], biochemical compounds [2] and metabolic networks [33], for instance. Usually, in order to take advantage of the existing data driven modeling systems [31], each pattern of a structured domain $\mathcal{X}$ is transformed to an $\mathbb{R}^m$ feature vector by adopting a suitable *explicit preprocessing* function $\phi : \mathcal{X} \rightarrow \mathbb{R}^m$. The design of these functions is a challenging problem, mainly due to the implicit *semantic and informative gap* between $\mathcal{X}$ and $\mathbb{R}^m$. A key element to design an automatic system dealing with these recognition problems is the *information granulation and compression* of the input set $\mathcal{X}$, that can be achieved through the definition of suited *information granules* [1]. Another approach is the one provided by kernels-based learning machines [30], where the representation of the input data in a high dimensional embedding space is performed *implicitly*, defining a suitable *valid* kernel function $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$.

Labeled graphs are general and flexible structures able to model both topological and semantic information in data. Consequently, the graph-based representation has been adopted extensively in different contexts. A labeled graph is a tuple $g = (\mathcal{V}, \mathcal{E}, \mu, \nu)$, where $\mathcal{V}$ is the (finite) set of vertices (also referred as nodes), $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 79–91.
DOI: 10.1007/978-3-642-36530-0_7

edges, $\mu : \mathcal{V} \rightarrow \mathcal{L}_{\mathcal{V}}$ is the vertex labeling (total) function with $\mathcal{L}_{\mathcal{V}}$ the vertex-labels set, and $\nu : \mathcal{E} \rightarrow \mathcal{L}_{\mathcal{E}}$ is the edge (total) labeling function with $\mathcal{L}_{\mathcal{E}}$ the edge-labels set. The generality of both $\mathcal{L}_{\mathcal{E}}$ and $\mathcal{L}_{\mathcal{V}}$ permits to represent a broad set of patterns. Consequently, each inductive modeling engine that has to deal with labeled graphs as input patterns, must be able to understand effectively, and efficiently, both structural and labels-related commonalities. For this purpose, a suited *graph matching* procedure [10,17] must be defined, able to act as the basic matching measure for any given pair of graphs of $\mathcal{G}$. Of great interest are Inexact Graph Matching (IGM) procedures, that can be defined, from a very high level of abstraction, as nonnegative functions of the form $f : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}^{+}$.

1.1 Contribution and Paper Organization

In this paper, we describe two constructive tools for dealing with inexact matching of labeled graphs. The first one is a seriation method able to transform a general graph into a sequence of vertices labels. The second one is an approach based on a Granular Computing paradigm [1], that performs a recurrent patterns search in the structured domain, resulting in an alphabet of significant *symbols*. These symbols can be usefully employed to embed each original structured pattern into a real valued feature vector, which can be fed into well established Pattern Recognition algorithms, such as Support Vector Machines, Neurofuzzy Networks, etc. [31].

These two constructive tools are then employed to build two types of IGM techniques [17,10]. The first one is based on the principle of applying Dynamic Programming based alignment algorithms, such as Levenshtein [14] and Dynamic Time Warping (DTW) [29] on the data domain of the seriated graphs. The second one is based on the application of the recurrent substructures mining on the seriated graphs, in order to build an embedding procedure able to map the input graphs into real valued feature vectors. It is worth to remark that the embedding based on recurrent substructures search could be also applied directly on the input graphs domain $\mathcal{G}$, without performing the seriation stage. Moreover, it could be used for data compression in order to prepare the graphs for a subsequent seriation stage. The latter two approaches, however, have not been investigated in the present paper, and they are subject for future works.

This paper is organized as follows. In Section 2 we outline the tools we employ for building IGM algorithms. More specifically, in Section 2.1 the graph seriation algorithm is outlined and in Section 2.2 the two-stages method is briefly introduced. In Section 3 we give a detailed description of the core element of the second stage of the two-stages algorithm, *i.e.* the proposed Granular Computing method for the recurrent substructures analysis in the special case of sequences, such as the ones yielded by the seriation procedure. In Section 4 we integrate these two algorithms, described in Section 2.1 and 3, respectively, in an optimized classification system. Experiments of the graph-based recognition system follow in Section 5. Finally, in Section 6 we draw our conclusions.

2 Inexact Graph Matching

Pattern Recognition and Soft Computing systems are founded on the capability of dealing with the notion of *dissimilarity* (or equivalently, the *similarity*) between the input patterns of a generalized input set $\mathcal{X}$. For example, the very simple k-NN classifier is totally based on a suited dissimilarity measure tailored to the specific input set $\mathcal{X}$. When dealing with labeled graphs, that is when $\mathcal{X} = \mathcal{G}$, the notion of graphs (dis)similarity is developed through the well known IGM problem. Basically, IGM algorithms are conceived to be able to understand, in a unified framework, both topological and semantic commonalities among graphs. As a consequence, an IGM algorithm can be though as a function $f : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}^+$, that assigns a (dis)similarity value to a given pair of labeled graphs.

It is possible to distinguish three mainstream approaches in the technical literature [17,10]:

1. **Graph Edit Distance** [4,8,19,9]: these methods match the graphs directly in their domain and, in general, are applicable to a wide class of graphs.
2. **Graph Kernels** [2,13,16,18]: they are based on the notion of similarity between two discrete objects, that is evaluated on an implicitly induced high dimensional *feature space*. Being able to define a kernel function for graphs permits to import the whole class of kernel machines on this domain.
3. **Graph Embedding** [23,11,27,6]: these methods are based on the embedding of the graph to obtain a general (and usually relative to the data) explicit vector representation. These methods can be seen as a generalization of the graph kernels approach.

Therefore, IGM algorithms must be thought as the key procedures of any, from very simple to highly complex, Graph-based Pattern Recognition and Soft Computing system.

2.1 Graph Seriation

Given a graph g, the aim of the seriation is to establish an order on the set of vertices $\mathcal{V}(g)$, with $|\mathcal{V}(g)| = n$, such that the derived sequence of vertices $s = (v_{i_1}, v_{i_2}, ..., v_{i_n})$ respects a given property of the graph. For example, an interesting approach is the one that analyzes the *spectrum* (*i.e.* the set of its eigenvalues/eigenvectors) of the matrix representation of the graph [28]. The leading eigenvector ψ of the adjacency matrix contains the information about the structural connectivity of each vertex of the graph. Similarly, analyzing the (symmetric) transition matrix $\mathbf{T}^{n \times n}$, it is possible to obtain information about the a priori probability of a given vertex in a random walk scenario. In general this approach can be used for unweighted and weighted graphs. However, we observe that it is possible to extend this method also to the graphs with real vectors as weights on the edges, taking into account the (Euclidean) norm $W_{ij} = \parallel \nu(e_{ij}) \parallel$ of the vector, assuming that the highest is the norm, the stronger is the relationship. In Algorithm 1 it is shown the procedure to obtain the sequence of vertices s using the leading eigenvector ψ of the matrix representation of the graph. Note that in Algorithm 1 the sequence s is assumed to behave as a list of vertices.

Once the seriated version of the graph is available, some well established tools can be adopted to build some useful dissimilarity measure between sequences.

Algorithm 1. Graph Seriation.

```
Input: The leading eigenvector $\psi$ of $g$
Output: A sequence of vertices $s = (v_{j_1}, ..., v_{j_n})$
$k = 1$
$j_1 = \arg\max_{j \in \mathcal{V}} \psi(j)$
$add(s_1, j_1)$
$k = 2$
repeat
    $j_k = \arg\max_{j \in \mathcal{N}_{j_{k-1}} \wedge j \notin L} \psi(j)$
    $add(s_k, j_k)$
until All vertices $\mathcal{V}(g)$ are in $s$
return $s$
```

2.2 Graph Embedding through Substructures Analysis

This powerful tool performs a transformation $f : \mathcal{G} \to \mathcal{E}$, mapping each graph $g \in \mathcal{G}$ to a numeric vector $\underline{\mathbf{h}} \in \mathcal{E}$, where usually $\mathcal{E} \subseteq \mathbb{R}^n$. This transformation is realized by first creating the alphabet $\mathcal{A}$, a set of substructures frequently recurring in the whole dataset. Once the alphabet is formed by a procedure of substructures mining, the graph to transform is tested for the presence of occurrences of each element of $\mathcal{A}$. The resulting values can be used to form a vector having dimension equal to the cardinality of $\mathcal{A}$. Once the representation in form of real valued vectors is available, well know tools such as *Minkowski* or *Mahalanobis* distances can be used to build the final IGM function, depending on the semantic of data.

It can be noted that this procedure can be applied directly in the graphs domain $\mathcal{G}$. However, the method mainly investigated in this paper consists in applying the embedding procedure on the simplified sequenced data obtained from the graphs with the seriation algorithm described in Section 2.1. In this case, the transformation function can be viewed as the composition of two functions, $f = f_2 \circ f_1$, where $f_1 : \mathcal{G} \to \mathcal{S}$ transforms a graph g into a sequence s, and $f_2 : \mathcal{S} \to \mathcal{E}$ transforms the sequence s into the final embedding vector $\underline{\mathbf{h}}$. The substructures mining procedure is performed in the $\mathcal{S}$ domain, and consequently each pattern of $\mathcal{A}$ is actually a subsequence. The second transformation function will be described in the next section.

3 A Granular Computing-Based Mining Procedure for Sequences Embedding

In this section, we describe two variants of a Granular Computing embedding procedure able to cope with a set of sequences $\mathcal{S}$. For the purpose of this paper, we will assume that a sequence is actually a string of characters belonging to a finite nominal set. Consequently, if not specified differently, the two terms must be considered equivalent. The first algorithm, called *GRADIS* [15], follows strictly a *clustering ensembles* based approach [32], while the second one is called *RL-GRADIS* [25], and adds a *reinforcement learning* approach [12] to the same clustering scheme. *GRADIS* individuates different clusters of similar patterns by making complete partitions of the input dataset using a standard clustering algorithm. To this aim, a pure clustering approach is taken into account, specifically the well known *BSAS* algorithm [31]. Each cluster is viewed as a granule of similar, and hence indistinguishable, occurrences of a frequent pattern. The candidates are evaluated by means of a cost function taking into account compactness and cardinality of the clusters/granule. If a candidate cluster contains an adequate number of patterns which are similar enough, the candidate individuates a frequent pattern, otherwise it is discarded. The main drawback of this approach, in the case of large datasets, is the high computational cost required by the clustering ensembles procedure, usually performed with settings which yield a high number of clusters. This is due to the fact that system parameters settings producing a low number of clusters are not suited to the search of frequent patterns. The approach followed by *RL-GRADIS* overcomes the limits of the pure clustering based approach, by introducing a dynamic behavior for discarding clusters which are not updated *frequently* during the presentation of patterns. In this way, the number of clusters can be automatically limited even with low settings of the clustering threshold, only yielding compact clusters. Moreover, only clusters containing frequent patterns live up to the end of the process, limiting also the general computational cost of the system. For this purpose, *RL-GRADIS* should be defined as an unsupervised learning scheme based on clustering the input set $\mathcal{S}$.

3.1 Frequent Substructures Identification

Given an input set of sequences $\mathcal{S}$, the set of *n*-grams $\mathcal{N}$ (subsequences of length *n*) is extracted performing an *n*-grams analysis [15]. The set $\mathcal{N}$ is built considering each variable length *n*-gram extracted from each input sequence of $\mathcal{S}$. The length of each *n*-gram vary between two user-defined parameters l and L. For very large datasets, it can be unfeasible to extract and retain all the *n*-grams. Consequently, by using a suited user-defined selection probability p, only a subset $\mathcal{N}^* \subseteq \mathcal{N}$ can be retained. Therefore, the following analysis is performed on the set $\mathcal{N}^*$. As an example, if an input sequence is $s = (A, B, C, D)$, with $l = 2, L = 3$, we obtain the set of its *n*-grams $\mathcal{N}_s = \{(A, B), (A, B, C), (B, C), (B, C, D), (C, D)\}$. The same expansion is repeated for each $s \in \mathcal{S}$, eventually yielding the set $\mathcal{N} = \bigcup_{s\in\mathcal{S}} \mathcal{N}_s$ (and $\mathcal{N}^*$ if $p < 1$).

Algorithm 2, taken from [25, Algorithm 1], shows the pseudo-code that describes the *n*-grams discovery strategy adopted by *RL-GRADIS*. This algorithm is based on a dynamic list of *receptors* $\mathcal{R}$, which play the same role of cluster representatives as described for *GRADIS*, therefore providing a model of the corresponding cluster. In the case of substrings, a *MinSOD* representative [5] employing the Levenshtein distance is used. However, we stress that different matching measures for sequences could be employed, such as for example the DTW, tailored for the specific nature of the sequence itself (*e.g.*, time series or complex composite types). Each receptor is assigned a strength parameter denoted with f. This value is dynamically updated to track receptors associated to highly frequent patterns. It is easy to recognize the basic sequential analysis performed by the *BSAS* clustering algorithm (from line 3 to 20). The additional procedure aimed at the removal of the not-frequently updated receptors (*i.e.*, clusters together with its elements) is outlined from line 21 to 26. Basically, it consists in a loop that checks the strength f of each identified receptor in $\mathcal{R}$. If the algorithm finds that a receptor is not adequately updated over time, it will be removed from the current set of receptors $\mathcal{R}$. This test is performed matching the strength of each receptor with a user-defined threshold ϵ. The α, β and σ parameters, each falling in the interval $[0,\ 1]$, control

Algorithm 2. Symbols Alphabet Computation using Reinforcement Learning Approach.

```
Input: The set of $n$-grams $\mathcal{N}^* = \{n_0, n_1, ..., n_{|\mathcal{N}^*|}\}$, the maximum number of allowed clusters $Q$, the clustering
    threshold $\Theta$, $\alpha$, $\beta$, $\sigma$ and $\epsilon$ parameters
Output: A set of receptors $\mathcal{R}$
$\mathcal{R} = \emptyset$
for all $n_i \in \mathcal{N}^*$ do
    if $\mathcal{R} = \emptyset$ then
        Create new receptor $\hat{r}$ such that $f(\hat{r}) = \sigma$
        $update(\hat{r}, n_i)$
        $push(\mathcal{R}, \hat{r})$
    else
        $d_{min} = \min_{r_j \in \mathcal{R}} diss(r_j, n_i)$
        if $d_{min} > \Theta$ AND $size(\mathcal{R}) < Q$ then
            Create new receptor $\hat{r}$ such that $f(\hat{r}) = \sigma$
            $update(\hat{r}, n_i)$
            $push(\mathcal{R}, \hat{r})$
        else
            $\bar{r} = \arg\min_{r_j \in \mathcal{R}} diss(r_j, n_i)$
            $update(\bar{r}, n_i)$
            $f(\bar{r}) = f(\bar{r}) + \alpha(1 - f(\bar{r}))$
        end if
    end if
    for all $r_j \in \mathcal{R}$ do
        $f(r_j) = (1 - \beta) \cdot f(r_j)$
        if $f(r_j) < \epsilon$ then
            $pop(\mathcal{R}, r_j)$
        end if
    end for
end for
return $\mathcal{R}$
```

the dynamic behavior of the strength value f over time. In particular, σ stands for the default strength value, α is used as a reinforcing factor when the cluster is updated and β is used to adjust the speed of forgetfulness of receptors.

It is worth to stress that, even if the symbols identification algorithm developed in *RL-GRADIS* results to be usually more faster and essential than the clustering ensembles strategy of *GRADIS* [25], it is based on a single clustering threshold parameter, Θ. Consequently, the definition of the *right* value of this parameter becomes an important objective of study.

At the end of the procedure, the set of receptors $\mathcal{R} = \{r_1, r_2, ..., r_{|\mathcal{R}|}\}$ is completely identified. From $\mathcal{R}$ we derive the symbols alphabet $\mathcal{A}$, using as a measure of cluster quality the strength of the respective receptor. Compactness and size costs are associated to each cluster as additional descriptive measures, and are defined as $K(\mathcal{C}_j) = \frac{1}{n-1}\sum_{i=1}^{n-1} d_{LEV}(n_i, r_j)$ and $S(\mathcal{C}_j) = 1 - \frac{|\mathcal{C}_j|}{|\mathcal{N}^*|}$, respectively. Each symbol a_j is thus defined as a triple $(r_j, K(\mathcal{C}_j)\cdot\delta, f(r_j))$, where $K(\mathcal{C}_j)\cdot\delta$ is a cluster-dependent thresholding factor used in the subsequent embedding phase, and $f(r_j)$ is the quality of the cluster $\mathcal{C}_j$ (and in turn the quality of the receptor r_j) in the alphabet $\mathcal{A}$. Therefore, each symbol is equipped with the domain-dependent semantic, that in our case is defined as a pair of metric and quality information concerning the cluster from which it has been derived. We stress that the size cost $S(\mathcal{C}_j)$ is taken into account indirectly into the force factor computation (*i.e.*, $f(r_j)$), since a heavily populated cluster is updated frequently.

Conversely, the *GRADIS* approach defines a threshold τ, used for symbols filtering. The cost of each candidate symbol, denoted with $\Gamma(\mathcal{C}_j)$, is defined as a convex linear linear combination of the compactness and size costs

$$\Gamma(\mathcal{C}_j) = (1 - \mu) \cdot K(\mathcal{C}_j) + \mu \cdot S(\mathcal{C}_j) \tag{1}$$

If this cost is lower than the threshold τ, the cluster is retained in the alphabet $\mathcal{A}$, otherwise it is rejected. In this case, a symbol a_j is defined as $(n^{SOD}_{\mathcal{C}_j}, K(\mathcal{C}_j) \cdot \delta, 1 - \Gamma(\mathcal{C}_j))$, adopting a user-defined tolerance parameter $\delta \geq 1$.

Computational Complexity. The procedure shown in Algorithm 2 consists basically in a linear scan of the set $\mathcal{N}^*$. For each $n_i \in \mathcal{N}^*$, a number of distance computations, dependent on the clusters size and on the maximum number of allowed clusters Q, must be performed to update the *MinSOD* element (line 11 or 16) and to assign the input pattern n_i to the nearest receptor (line 8). If L is the maximum length of the *n*-grams, the cost of a single Levenshtein distance computation is given by $O(L^2)$. The assignment of each n_i to a receptor has a cost bounded by $Q \cdot L^2$, and the *MinSOD* update cost can be upper bounded by $|\mathcal{C}|^2 \cdot L^2$, where $\mathcal{C} \subset \mathcal{N}^*$ is the constant-size cache of the *MinSOD* [5]. Moreover, the loop from line 21 to line 26 consists in at most Q evaluations of the receptors strength. Thus, the overall computational complexity is upper bounded by $O(|\mathcal{N}^*| \cdot (Q \cdot L^2 + |\mathcal{C}|^2 \cdot L^2 + Q)) = O(|\mathcal{N}^*| \cdot L^2 \cdot (Q + |\mathcal{C}|^2) + |\mathcal{N}^*| \cdot Q)$, that is linear in the size of the input $\mathcal{N}^*$.

3.2 Sequences Embedding Method

Both *GRADIS* and *RL-GRADIS* share the following sequences embedding procedure. The embedding space $\mathcal{E}$ is built upon a *local reference framework*, defined on the base of the symbols alphabet $\mathcal{A}$. Indeed, the set $\mathcal{A}$ can be seen as a set of prototypes of the input set $\mathcal{S}$, used to produce a *dissimilarity space* representation $\mathcal{E}$ [20]. Practically, if $|\mathcal{A}| = d$, each sequence $s_j, j = 1 \rightarrow |\mathcal{S}|$, is represented as a d-dimensional numeric vector, $\underline{\mathbf{h}}_j \in \mathbb{R}^d$, called *symbolic histogram*. In the *i*-th component of $\underline{\mathbf{h}}$ is counted the number of occurrences of the symbol a_i into s_j, evaluating the match degree using an inexact matching procedure, on the base of the adopted sequences dissimilarity (*e.g.*, Levenshtein).

Algorithm 3 shows the embedding algorithm. Let $\nu_i = \lfloor len(s_i) \cdot \lambda \rfloor$ be the length tolerance adopted in order to find matches in s_i, where λ is a user-defined parameter in $[0, 1]$. The selected set of *n*-grams $\mathcal{N}_{s_i}$ of s_i, of variable length n between $len(s_i) - \nu_i \leq n \leq len(s_i) + \nu_i$, is extracted and the inexact matching is computed against a_j and each $n_k \in \mathcal{N}_{s_i}$. If the matching value is lower or equal to the symbol-dependent value $K(\mathcal{C}_j) \cdot \delta$, the counting is incremented by one. Further post-processing techniques can be applied to each histogram in $\mathcal{E}$, such as normalizations and different (monotonic) transformations, aimed at reshaping data.

4 Optimized Classification System

A very important feature of a classification system is its generalization capability, that can be estimated evaluating a suited performance measure, related to the classification

Algorithm 3. Symbolic Histogram Representation.

```
Input: The input set $\mathcal{S}$, the symbols alphabet $\mathcal{A}$, parameters $\lambda$ and $\delta$
Output: A set of symbolic histogram representations $\mathcal{H} \subseteq \mathcal{E}$
$\mathcal{H} = \emptyset$
for all $s_i \in \mathcal{S}$ do
    Initialize the relative zero-valued symbolic histogram $\underline{\mathbf{h}}^{(i)}$
    Let $\nu_i = \lfloor len(s_i) \cdot \lambda \rfloor$
    Extract all the $n$-grams of $s_i$, denoted as $\mathcal{N}_{s_i}$, of variable length $n$ between $len(s_i) - \nu_i \leq n \leq len(s_i) + \nu_i$
    for all $a_j \in \mathcal{A}$ do
        for all $n_k \in \mathcal{N}_{s_i}$ do
            Compute the inexact matching value $d_{jk} = d(a_j, n_k)$
            if $d_{jk} \leq K(\mathcal{C}_j) \cdot \delta$ then
                $h_j^{(i)} = h_j^{(i)} + 1$
            end if
        end for
    end for
    $\mathcal{H} = \mathcal{H} \cup \{\underline{\mathbf{h}}^{(i)}\}$
end for
```

model, on the test set. The well-known *Occam's Razor* principle [31] of learning theory can be interpreted, for the purpose of classification systems, as a *law of parsimony* concerning the learned classifier. That is, under the same conditions of performance on the training set, the classification model that shows the best generalization capability is the one that is characterized by the lowest structural complexity.

The proposed classification system performs an automatic stage of genetic algorithm-based model optimization and *wrapper*-based feature selection on $\mathcal{E}$ [25], using only the information of the training set. This is done defining a fitness function with the aim of optimizing a linear convex combination of the classification accuracy and the *composite* structural complexity. The composite structural complexity is itself defined as a linear convex combination of the fraction of selected features and the model structural complexity of the classifier. Formally, if $\hat{\underline{\mathbf{h}}}$ is the reduced symbolic histogram of a given input sequence s, we compute its fitness function as

$$f(\hat{\underline{\mathbf{h}}}) = 1 - (\eta \cdot Err_{\mathcal{S}_{tr}} + (1-\eta) \cdot (\gamma \cdot FC + (1-\gamma) \cdot SC)) \tag{2}$$

In Equation 2, $Err_{\mathcal{S}_{tr}}$ is the classification error rate achieved over the training set, FC stands for the fraction of selected feature and $SC = 1 - \exp(-(\#I/\varDelta))$ is a non linear compression of the structural complexity measure. For instance, using *SVM* [3] as classifier on $\mathcal{E}$, $\#I$ is given by the number of adopted support vectors, while in the case of a neuro-fuzzy Min-Max model [26], is given by the number of hyperboxes. Of course, a wide range of feature-based classifiers are applicable to embedding space $\mathcal{E}$. The parameter $\varDelta$ is a factor used to modulate the decrease speed of the nonlinear compressing function. However, the SC factor could be defined also adopting linear functions, getting rid of the modulating parameter $\varDelta$. The η and γ parameters can be adapted to focus, during the optimization stage, on the structural complexity of the classification model and on the dimensionality of the embedding space, or to the classification error. Once the model optimization phase is terminated, the best-performing setup is considered and the classification accuracy on the test set is evaluated directly as

$$f(\hat{\underline{\mathbf{h}}}) = 1 - Err_{\mathcal{S}_{ts}} \tag{3}$$

5 Experiments

In the following two subsections, we show and discuss two experimental evaluations on a synthetically generated set of classification problem instances, and a test on a well know shared dataset of graphs belonging to different scientific contexts [21]. The methods under analysis are the following:

- k-NN classification rule using Levenshtein distance applied to sequenced graphs $\mathcal{S}$.
- The *SVM* classification system applied to the embedding space $\mathcal{E}$, employed in the optimized classification system described in Section 4.

5.1 Experiments on Synthetic Data

The aim of this test is to progressively increase the hardness of the classification task, measuring the robustness of the system. We focus on an equally distributed two-classes set of classification problem instances, using only weighted graphs. Each class of graphs, for each problem instance, has been generated using a *Markov Chains*-based approach [15]. That is, each class is entirely described by a proper transition matrix. The order of the graphs is set up to 100 and, their size is randomly determined, with 1000, 500 and 500 graphs for the training validation and test sets, respectively. To control the hardness of the classification problem, we produce a sequence of generating transition matrices with different level of randomness. A transition matrix is said to be fully random if the transition probabilities are uniform. Thus, for each classification problem instance, we generate the two transition matrices (one for each class of graphs) introducing two real parameters, $\alpha, \beta \in [0, 1]$, controlling the similarity of the transition matrices. Let $\mathbf{P}_1$ and $\mathbf{P}_2$ be two different permutation matrices, and let $\mathbf{U}$ be the uniform transition matrix, with a zero diagonal. We firstly generate two intermediate matrices, say $\mathbf{A}$ and $\mathbf{B}$, as shown in [15, Equation 2]. Finally, we obtain the two transition matrices, characterized by a desired similarity, as shown in [15, Equation 3]. In the following tests, different combinations of α and β have been used, generating 118 graph classification problem instances.

Results with Both Approaches. Tests on synthetic data demonstrate the correctness of the implementations and the robustness and stability of the employed tools. In fact, for the k-NN with Levenshtein distance method, except four cases, the classification accuracy percentage is always 100% for values of α and β in the interval $[0.1, 0.9]$. Only for extreme values of α and β (*i.e.*, 0 and 1) the behavior becomes unstable and a high number of errors is observed, approaching to the performance of the random classifier. The employed k in this case has been fixed to one. The approach based on the explicit embedding performs in a very similar way, yielding slightly worse results [15].

5.2 Experiments on the *IAM* Dataset

In this section, we provide different comparative experimental evaluations over two *IAM* datasets [21]. The *AIDS* dataset is a not-equally distributed two-class problem with 250,

Table 1. Classification Accuracy Percentages on the *AIDS* and *Mutagenicity* Datasets

System	Datasets	
	AIDS	Mutagenicity
BP-H+*k*-NN [8]	99.2	68.3
BP-V+*k*-NN [8]	98.9	67.6
GC+*k*-NN [16]	99.2	-
wBMF+*k*-NN [6]	94.0	69.1
sk+*SVM* [22]	97.4	55.4
k-NN+*Levenshtein*	99.0	71.1
RL-GRADIS+*SVM*	98.0	67.1
GRADIS+*SVM*	98.5	59.0

250 and 1500 samples for the training, validation and test set, respectively. The represented data are molecular compounds, denoting or not activity against *HIV*. The atoms are represented directly through the vertices, and covalent bonds by the edges of the graph. Vertices are labeled with the chemical symbol and edges by the valence of the linkage. The *Mutagenicity* dataset is again a two class problem, concerning the classification of chemical agents depending on their mutagenicity properties. The graphs labels and their meaning are the same as the ones of the *AIDS* dataset. The training, validation and test set contain 1500, 500 and 2337 graphs, respectively. The following results have been obtained on a machine with an Intel Core 2 Quad CPU Q6600 2.40GHz and 4 GB of main memory, running a 64-bit Linux OS.

Results with Levenshtein-Based *k*-NN. As concerns the *AIDS* dataset, the classification result is comparable with the one obtained by other state of the art competing algorithms. In fact the classification accuracy percentage of the antagonist methods varies from 94.0% to 99.2%, while the proposed method achieves 99.0%. As concerns the *Mutagenicity* dataset, the proposed method slightly outperforms the competing algorithms, achieving an accuracy of 71.1% against results which vary from 55.4% to 69.1%. The reported results in Table 1 have been obtained with $k = 1$ and $k = 3$, respectively, resulting to be the best configurations. However, we tested the method with k varying from one to seven, yielding only slightly different results.

Results with the Embedding-Based Systems. For what concerns the embedding of the *AIDS* dataset, *RL-GRADIS* employs only 0.247 seconds for the alphabet computation and 1.251 seconds for the embedding of the three datasets (*i.e.*, training, test and validation sets). The alphabet contains 86 symbols. The *RL-GRADIS* has been executed setting $\alpha = 0.01, \beta = 0.001, \epsilon = 0.001$ and $\theta = 0.2$. On the other hand *GRADIS* computes the alphabet of symbols in 1.304 seconds and performs the embedding in roughly the same computing time. The alphabet contains 100 symbols, yielding a less compact representation of the input set. The *GRADIS* procedure has been executed setting $\Theta_{\text{min}} = 0, \Theta_{\text{max}} = 0.2$ and $\tau = 0.75$. In both cases, the algorithms have been executed searching for *n*-grams of length between 3 and 5, limiting the maximum number of clusters to 100, and retaining the full retrieved *n*-grams set $\mathcal{N}$.

For what concerns the *Mutagenicity* dataset, *RL-GRADIS* performs the alphabet identification in 5.045 seconds and the embedding of the three datasets in 6.941 seconds. The alphabet contains 76 symbols. The procedure has been executed with $\alpha = 0.01, \beta = 0.003, \epsilon = 0.001$ and $\theta = 0.2$. The *GRADIS* procedure employs 19.255 seconds for the alphabet computation and roughly the same computing time for the embedding of the three datasets. The retrieved alphabet contains 100 symbols and the execution setup consists in setting $\Theta_{\text{min}} = 0, \Theta_{\text{max}} = 0.2$ and $\tau = 0.75$. Both algorithm were aligned searching for n-grams between length 3 and 6, and again limiting the maximum number of clusters to 100, retaining the full set of n-grams.

Table 1 shows the average accuracy percentages, obtained executing ten times the classification stage using different random seeds affecting the genetic algorithm behavior employed in the optimization stage. For both systems (*i.e.*, *GRADIS* and *RL-GRADIS*), *SVM* (actually, the *C-SVM* version, setting $C = 1000$) has been employed as the final classifier on the produced embeddings, performing 100 evolutions and using a population of 20 individuals for the optimization stage of the classifier (see Section 4). The synthesized classifier has been obtained, for the *AIDS* dataset, in 15.96 and 22.98 seconds, selecting 24 and 10 features, using the *RL-GRADIS* and *GRADIS* embeddings, respectively. For what concerns the *Mutagenicity* dataset, the synthesis has required 157.98 and 306.96 seconds, selecting 22 and 33 features, using *RL-GRADIS* and *GRADIS* derived embeddings. In all the configurations, the standard deviation of the results is around 1%. As a whole, the proposed two stages IGM procedure, employed in classification tasks, exhibits stable and encouraging results.

6 Conclusions and Future Directions

In this paper, graph seriation and embedding procedures based on substructure recurrence analysis have been employed in two different algorithmic arrangements for classification purposes. Tests have been carried out on synthetic data for robustness verification, and on two particular datasets from the *IAM* database. The simpler approach only based on seriation and Levenshtein distance showed better performances. This could be due to the fact that frequent subgraphs in the original graph dataset $\mathcal{G}$ are not mapped into recurrent substrings in $\mathcal{S}$ after the seriation stage. We are planning further tests in order to ascertain this hypothesis. It is important to underline that in any classification approach based on graph seriation, once obtained $\mathcal{S}$, all useful information stored in edges labels is definitively unavailable to subsequent processing stages. For this reason, this information should be used appropriately in the seriation stage. Consequently, in order to take full advantage of edge label information, as observed in Section 2.1, a meaningful norm on this labels set (*i.e.*, the set $\mathcal{L}_{\mathcal{E}}$) should be defined, in order to represent the strength of the relationship between vertices, which in turn is the very fundamental information exploited during the seriation stage. However, in both the *AIDS* and *Mutagenicity* datasets edge labels information has not taken into account in computing a weighted adjacency or transition matrix. In spite of this, the classification system based on *RL-GRADIS* achieves satisfactory classification results, not far from the other state of the art approaches.

Future works include to validate further the overall performance measures using other real world datasets with additional algorithmic improvements. Moreover, another interesting approach consists in searching frequent substructures directly in the original graph domain $\mathcal{G}$, and using them to define a suited compressed semantic representation of each graph, defining an intermediate information granulation level, in order to perform another frequent subgraphs mining step or directly the seriation stage followed by an *RL-GRADIS* embedding. Further algorithmic refinements include the automatic optimization of crucial parameters by means of improved meta-heuristic global optimization methods. Another possible improvement consists in enhancing the simple clustering algorithm adopted in *RL-GRADIS*, defining more meaningful cluster compactness costs functions, for example relying on higher order statistical descriptors (such as a joint measure of mean and variance, kurtosis, etc.) or using fuzzy sets related mathematical tools for cluster models.

References

1. Bargiela, A., Pedrycz, W.: Granular computing: an introduction. Kluwer international series in engineering and computer science, vol. (2002). Kluwer Academic Publishers (2003)
2. Borgwardt, K.M., Ong, C.S., Schönauer, S., Vishwanathan, S.V.N., Smola, A.J., Kriegel, H.P.: Protein function prediction via graph kernels. Bioinformatics 21, 47–56 (2005)
3. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT 1992, pp. 144–152. ACM, New York (1992)
4. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters 1(4), 245–253 (1983)
5. Del Vescovo, G., Livi, L., Rizzi, A., Frattale Mascioli, F.M.: Clustering structured data with the spare library. In: Proceeding of 2011 4th IEEE Int. Conf. on Computer Science and Information Technology, vol. 9, pp. 413–417 (June 2011)
6. Del Vescovo, G., Rizzi, A.: Automatic classification of graphs by symbolic histograms. In: Proceedings of the 2007 IEEE International Conference on Granular Computing, GRC 2007, pp. 410–416. IEEE Computer Society (2007)
7. Del Vescovo, G., Rizzi, A.: Online handwriting recognition by the symbolic histograms approach. In: Proceedings of the 2007 IEEE International Conference on Granular Computing, GRC 2007, pp. 686–690. IEEE Computer Society, Washington, DC (2007)
8. Fankhauser, S., Riesen, K., Bunke, H.: Speeding Up Graph Edit Distance Computation through Fast Bipartite Matching. In: Jiang, X., Ferrer, M., Torsello, A. (eds.) GbRPR 2011. LNCS, vol. 6658, pp. 102–111. Springer, Heidelberg (2011)
9. Gao, X., Xiao, B., Tao, D., Li, X.: Image categorization: Graph edit direction histogram. Pattern Recognition 41(10), 3179–3191 (2008)
10. Gao, X., Xiao, B., Tao, D., Li, X.: A survey of graph edit distance. Pattern Anal. Appl. 13(1), 113–129 (2010)
11. Jain, B.J., Obermayer, K.: Structure spaces. J. Mach. Learn. Res. 10, 2667–2714 (2009)
12. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: A survey. CoRR cs.AI/9605103 (1996)
13. Kashima, H., Tsuda, K., Inokuchi, A.: Marginalized kernels between labeled graphs. In: Proceedings of the Twentieth International Conference on Machine Learning, pp. 321–328. AAAI Press (2003)
14. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8 (1966)
15. Livi, L., Del Vescovo, G., Rizzi, A.: Graph recognition by seriation and frequent substructures mining. In: Proceeding of the First International Conference on Pattern Recognition Applications and Methods, vol. 1, pp. 186–191 (2012)

16. Livi, L., Del Vescovo, G., Rizzi, A.: Inexact graph matching through graph coverage. In: Proceeding of the First International Conference on Pattern Recognition Applications and Methods, vol. 1, pp. 269–272 (2012)
17. Livi, L., Rizzi, A.: The graph matching problem: a survey. To appear: Pattern Anal. Appl. (2012)
18. Livi, L., Rizzi, A.: Parallel algorithms for tensor product-based inexact graph matching. To be presented at IJCNN 2012 (2012)
19. Neuhaus, M., Riesen, K., Bunke, H.: Fast Suboptimal Algorithms for the Computation of Graph Edit Distance. In: Yeung, D.-Y., Kwok, J.T., Fred, A., Roli, F., de Ridder, D. (eds.) SSPR&SPR 2006. LNCS, vol. 4109, pp. 163–172. Springer, Heidelberg (2006)
20. Pękalska, E., Duin, R.: The dissimilarity representation for pattern recognition: foundations and applications. Series in machine perception and artificial intelligence. World Scientific (2005)
21. Riesen, K., Bunke, H.: IAM Graph Database Repository for Graph Based Pattern Recognition and Machine Learning. In: da Vitoria Lobo, N., Kasparis, T., Roli, F., Kwok, J.T., Georgiopoulos, M., Anagnostopoulos, G.C., Loog, M. (eds.) SSPR&SPR 2008. LNCS, vol. 5342, pp. 287–297. Springer, Heidelberg (2008)
22. Riesen, K., Bunke, H.: Graph classification by means of lipschitz embedding. Trans. Sys. Man Cyber. Part B 39, 1472–1483 (2009)
23. Riesen, K., Bunke, H.: Graph Classification and Clustering Based on Vector Space Embedding. Series in Machine Perception and Artificial Intelligence. World Scientific Pub. Co. Inc. (2010)
24. Rizzi, A., Del Vescovo, G.: Automatic image classification by a granular computing approach. In: Proceedings of the 2006 16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing, pp. 33–38 (2006)
25. Rizzi, A., Del Vescovo, G., Livi, L., Frattale Mascioli, F.M.: A new granular computing approach for sequences representation and classification. To be Presented at IJCNN 2012 (2012)
26. Rizzi, A., Panella, M., Frattale Mascioli, F.M.: Adaptive resolution min-max classifiers. IEEE Transactions on Neural Networks 13, 402–414 (2002)
27. Robles-Kelly, A., Hancock, E.R.: String Edit Distance, Random Walks and Graph Matching. In: Caelli, T.M., Amin, A., Duin, R.P.W., Kamel, M.S., de Ridder, D. (eds.) SSPR&SPR 2002. LNCS, vol. 2396, pp. 104–129. Springer, Heidelberg (2002)
28. Robles-Kelly, A., Hancock, E.R.: Graph edit distance from spectral seriation. IEEE Trans. Pattern Anal. Mach. Intell. 27, 365–378 (2005)
29. Sakoe, H.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 26, 43–49 (1978)
30. Schölkopf, B., Smola, A.: Learning with kernels: support vector machines, regularization, optimization, and beyond. Adaptive computation and machine learning. MIT Press (2002)
31. Theodoridis, S., Koutroumbas, K.: Pattern recognition. Elsevier/Academic Press (2006)
32. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: models of consensus and weak partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(12), 1866–1881 (2005)
33. Tun, K., Dhar, P., Palumbo, M., Giuliani, A.: Metabolic pathways variability and sequence/networks comparisons. BMC Bioinformatics 7(1), 24 (2006)

Segmentation of Nonstationary Time Series with Geometric Clustering

Alexei Bocharov and Bo Thiesson

Microsoft Research, One Microsoft Way, Redmond, WA 98052, U.S.A.
{alexeib,thiesson}@microsoft.com

Abstract. We introduce a non-parametric method for segmentation in regime-switching time-series models. The approach is based on spectral clustering of target-regressor tuples and derives a switching regression tree, where regime switches are modeled by oblique splits. Such models can be learned efficiently from data, where clustering is used to propose one single split candidate at each split level. We use the class of ART time series models to serve as illustration, but because of the non-parametric nature of our segmentation approach, it readily generalizes to a wide range of time-series models that go beyond the Gaussian error assumption in ART models. Experimental results on S&P 1500 financial trading data demonstrates dramatically improved predictive accuracy for the exemplifying ART models.

Keywords: Regime-switching time series, Spectral clustering, Regression tree, Oblique split, Financial markets.

1 Introduction

The analysis of time-series data is an important area of research with applications in areas such as natural sciences, economics, and finance to mention a few.

Many time series exhibit nonstationarity due to regime switching. Proper detection and modeling of this switching is a major challenge in time-series analysis. In regime-switching models, different time series regimes are described by submodels with different sets of parameters. A particular submodel may apply to multiple time ranges when the underlying time series repeatedly falls into a certain regime. For example, volatility of equity returns may change when affected by events such as earnings releases or analysts' reports, and we may see similar volatility patterns around similar events.

The intuition in this paper is to match proposed regimes with modes of the joint distribution of target-regressor tuples, which is a particular kind of mixture modeling. Prior research offers quite a variety of mixture modeling approaches to the analysis of nonstationary time series. In Markov-switching models (see, e.g., [12,13]) a Markov evolving hidden state indirectly partitions the time-series data to fit local auto-regressive models in the mixture components. Another large body of work (see, e.g., [27,28]) have adapted the hierarchical mixtures of experts in [15] to time series. In these models–also denoted as gated experts–the hierarchical gates explicitly operate on the data in order to define a partition into local regimes. In both the Markov-switching and the gated expert

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 93–107.
DOI: 10.1007/978-3-642-36530-0_8

models, the determination of the partition and the local regimes are tightly integrated in the learning algorithm and demands an iterative approach, such as the EM algorithm.

We focus on a conceptually simple direction that lends itself easier to explanatory analysis. The resulting design differs from the above work in at least three aspects: 1) we propose a modular separation of the regime partitioning and the regime learning, which makes it easy to experiment independently with different types of regime models and different separation methods, 2) in particular, this modularity allows for non-parametric as well as parametric regime models, or a mixture thereof, 3) the regime-switching conditions depend deterministically on data and are easy to interpret.

We model the actual switching conditions in a regime-switching model in the form of a regression tree and call it the *switching tree*. Typically, the construction of a regression tree is a stagewise process that involves three ingredients: 1) a *split proposer* that creates split candidates to consider for a given (leaf) node in the tree, 2) one or more *scoring criteria* for evaluating the benefit of a split candidate, and 3) a *search strategy* that decides which nodes to consider and which scoring criterion to apply at any state during the construction of the tree. Since the seminal paper [2] popularized the classic classification and regression tree (CART) algorithm, the research community has given a lot of attention to both types of decision trees. Many different algorithms have been proposed in the literature by varying specifics for the three ingredients in the construction process mentioned above.

Although there has been much research on learning regression trees, we know of only one setting, where these models have been used as switching trees in regime-switching time series models–namely the class of auto-regressive tree (ART) models in [18]. The ART models generalize classical auto-regressive (AR) models (e.g., [11]) by having a regression tree define the switching between the different AR models in the leafs. As such, the well-known threshold auto-regressive (TAR) models [25,24] can also be considered as a specialization of an ART model with the regression tree limited to a single split variable. The layout of our algorithms is strongly influenced by [18] (which we repeatedly refer to for comparison), but our premises and approach is very different.

In particular, we propose a different way to create the candidate splits during the switching tree construction. A split defines a predicate, which, given the values of regressor variables, decides on which side of the split a data case should belong.[1] A predicate may be as simple as checking if the value of a particular single regressor is below some threshold or not. We will refer to this kind of split as an *axial split*, and it is in fact the only type of splits allowed in the ART models. We make use of general multi-regressor split predicates, which in this paper we approximate with linear predicates called *oblique splits*. Importantly, we show evidence that for a broad class of time series, the best split is not likely to be axial.

It may sometimes be possible to consider and evaluate the efficacy of all feasible axial splits for the data associated with a node in the tree, but for combinatorial reasons, oblique splitting rarely enjoys this luxury. We therefore need a split proposer, which is more careful about the candidate splits it proposes. In fact, our approach is extreme in that respect by only proposing a *single* oblique split to be considered for any given node

[1] For clarity of presentation, we will focus on binary splits only. It is a trivial exercise to extend our proposed method to allow for n-ary splits.

during the construction of the tree. Our oblique split proposer involves a simple two step procedure. In the first step, we use a spectral clustering method to separate the data in a node into two classes. Having separated the data, the second step now proceeds as a simple classification problem, by using a linear discriminant method to create the best separating hyperplane for the two data classes. Any discriminant method can be used, and there is in principle no restriction on it being linear, if more complicated splits are sought.

Oblique splitting has enjoyed significant attention for the classification tree setting. See, e.g., [2,20,3,10,14]. Less attention has been given to the regression tree setting, but still a number of methods has come out of the statistics and machine learning communities. See, e.g., [7,16,4] to mention a few. Setting aside the time-series context for our switching trees, the work in [7] is in style the most similar to the oblique splitting approach that we propose in this paper. In [7], the EM algorithm for Gaussian mixtures is used to cluster the data. Having committed to Gaussian clusters it now makes sense to determine a separating hyperplane via a quadratic discriminant analysis for a projection of the data onto a vector that ensures maximum separation of the Gaussians. This vector is found by minimizing Fisher's separability criterion.

Our approach to proposing oblique split candidates is agnostic to any specific parametric assumptions on the noise distribution and therefore accommodates without change non-Gaussian or even correlated errors (thus our method is more general than ART, which relies on univariate Gaussian quantiles as split candidates). This approach allows us to use spectral clustering - a non-parametric segmentation tool, which has been shown to often outperform parametric clustering tools (see, e.g., [26]).

Spectral clustering dates back to the work in [8,9] that suggest to use the method for graph partitionings. Variations of spectral clustering have later been popularized in the machine learning community [23,19,21], and, importantly, very good progress has been made in improving an otherwise computationally expensive eigenvector computation for these methods [29]. We use a simple variation of the method in [21] to create a spectral clustering for the time series data in a node. Given this clustering, we then use a simple perceptron learning algorithm (see, e.g., [1]) to find a hyperplane that defines a good oblique split predicate for the autoregressors in the model.

Let us now turn to the possibility of splitting on the time feature in a time series. Due to the special nature of time, it does not make sense to involve this feature as an extra dimension in the spectral clustering; it would not add any discriminating power to the method. Instead, we propose a procedure for time splits, which uses the clustering in another way. The procedure identifies specific points in time, where succeeding data elements in the series cross the cluster boundary, and proposes time splits at those points. Our split proposer will in this way use the spectral clustering to produce both the oblique split candidate for the regressors, and a few very targeted (axial) split candidates for the time dimension.

The rest of the paper is organized as follows. In Section 2, we briefly review the ART models that we use as a baseline, and we define and motivate the extension that allows for oblique splits. Section 3 reviews the general learning framework for ART models. Section 4 contains the details for both aspects of our proposed spectral splitting method– the oblique splitting and the time splitting. In Sections 5 and 6 we describe experiments and provide experimental evidence demonstrating that our proposed spectral splitting method dramatically improves the quality of the learned ART models over the current approach. We will conclude in Section 7.

2 Standard and Oblique ART Models

We begin by introducing some notation. We denote a temporal sequence of variables by $X = (X_1, X_2, \ldots, X_T)$, and we denote a sub-sequence consisting of the i'th through the j'th element by $X_i^j = (X_i, X_{i+1}, \ldots, X_j)$, $i < j$. Time-series data is a sequence of values for these variables denoted by $x = (x_1, x_2, \ldots, x_T)$. We assume continuous values, obtained at discrete, equispaced intervals of time.

An autoregressive (AR) model of length p, is simply a p-order Markov model that imposes a linear regression for the current value of the time series given the immediate past of p previous values. That is,

$$p(x_t|x_1^{t-1}) = p(x_t|x_{t-p}^{t-1}) \sim \mathcal{N}(m + \sum_{j=1}^{p} b_j x_{t-j}, \sigma^2)$$

where $\mathcal{N}(\mu, \sigma^2)$ is a conditional normal distribution with mean μ and variance σ^2, and $\theta = (m, b_1, \ldots, b_p, \sigma^2)$ are the model parameters (e.g., [6, page 55]).

The ART models is a regime-switching generalization of the AR models, where a switching regression tree determines which AR model to apply at each time step. The autoregressors therefore have two purposes: as input for a classification that determines a particular regime, and as predictor variables in the linear regression for the specific AR model in that regime.

As a second generalization[2], ART models may allow exogenous variables, such as past observations from related time series, as regressors in the model. Time (or time-step) is a special exogenous variable, only allowed in a split condition, and is therefore only used for modeling change points in the series.

2.1 Axial and Oblique Splits

Different types of switching regression trees can be characterized by the kind of predicates they allow for splits in the tree. The ART models allow only a simple form of binary splits, where a predicate tests the value of a single regressor. The models handle continuous variables, and a split predicate is therefore of the form

$$X_i \leq c$$

where c is a constant value and X_i is any one of the regressors in the model or a variable representing time. A simple split of this type is also called *axial*, because the predicate that splits the data at a node can be considered as a hyperplane that is orthogonal to the axis for one of the regressor variables or the time variable.

[2] The class of ART models with exogenous variables has not been documented in any paper. We have learned about this generalization from communications with the authors of [18].

The best split for a node in the tree can be learned by considering all possible partitionings of the data according to each of the individual regressors in the model, and then picking the highest scoring split for these candidates according to some criterion. It can, however, be computationally demanding to evaluate scores for that many split candidates, and for that reason, [5] investigated a Gaussian quantile approach that proposes only 15 split points for each regressor. They found that this approach is competitive to the more exhaustive approach. A commercial implementation for ART models uses the Gaussian quantile approach and we will compare our alternative to this approach.

We propose a solution, which will only produce a single split candidate to be considered for the entire set of regressors. In this solution we extend the class of ART models to allow for a more general split predicate of the form

$$\sum_i a_i X_i \le c \tag{1}$$

where the sum is over all the regressors in the model and a_i are corresponding coefficients. Splits of this type are in [20] called *oblique* due to the fact that a hyperplane that splits data according to the linear predicate is oblique with respect to the regressor axes. We will in Section 4 describe the details behind the method that we use to produce an oblique split candidate.

2.2 Motivation for Oblique Splits

There are general statistical reasons why, in many situations, oblique splits are preferable over axial splits. In fact, for a broad class of time series, the best splitting hyperplane turns out to be approximately orthogonal to the principal diagonal $d = (\frac{1}{\sqrt{p}}, \ldots, \frac{1}{\sqrt{p}})$. To qualify this fact, consider two pre-defined classes of segments $x^{(c)}, c = 1, 2$ for the time-series data x. Let $\mu^{(c)}$ and $\Sigma^{(c)}$ denote the mean vector and covariance matrix for the sample joint distribution of X_{t-p}^{t-1}, computed for observations on p regressors for targets $x_t \in x^{(c)}$.

Let us define the moving average $A_t = \frac{1}{p} \sum_{i=1}^{p} X_{t-i}$. We show in the Appendix that in the context where $X_{t-i} - A_t$ is weakly correlated with A_t, while its variance is comparable with that of A_t, the angle between the principal diagonal and one of the principal axes of $\Sigma^{(c)}, c = 1, 2$ is small. This would certainly be the case with a broad range of financial data, where increments in price curves have notoriously low correlations with price values [22,17], while seldom overwhelming the averages in magnitude. With one of the principal axes being approximately aligned with the principal diagonal d for both $\Sigma^{(1)}$ and $\Sigma^{(2)}$ it is unlikely that a cut orthogonal to either of the coordinate axes $X_{t-1}, \ldots, X_{t-p}$ can provide optimal separation of the two classes.

3 The Learning Procedure

An ART model is typically learned in a stagewise fashion. The learning process starts from the trivial model without any regressors and then greedily evaluates regressors one at a time and adds the ones that improve a chosen scoring criterion to model, while scoring criterion keeps improving.

The task of learning a specific autoregressive model considered at any stage in this process can be cast into a standard task of learning a linear regression tree. It is done by a trivial transformation of the time-series data into multivariate data cases for the regressor and target variables in the model. For example, when learning an ART model of length p with an exogenous regressor, say z_{t-q}, from a related time series, the transformation creates the set of $T - \max(p, q)$ cases of the type (x_{t-p}^{t}, z_{t-q}), where $\max(p, q) + 1 < t \leq T$. We will in the following denote this transformation as the *phase view*, due to a vague analogy to the phase trajectory in the theory of dynamical systems.

Most regression tree learning algorithms construct a tree in two stages (see, e.g., [2]): First, in a growing stage, the learning algorithm will maximize a scoring criterion by recursively trying to replace leaf nodes by better scoring splits. A least-squares deviation criterion is often used for scoring splits in a regression tree. Typically the chosen criterion will cause the selection of an overly large tree with poor generalization. In a pruning stage, the tree is therefore pruned back by greedily eliminating leaves using a second criterion–such as the holdout score on a validation data set–with the goal of minimizing the error on unseen data.

In contrast, [18] suggests a learning algorithm that uses a Bayesian scoring criterion, described in detail in that paper. This criterion avoids over-fitting by penalizing for the complexity of the model, and consequently, the pruning stage is not needed. We use this Bayesian criterion in our experimental section.

In the next section, we describe the details of the algorithm we propose for producing the candidate splits that are considered during the recursive construction of a regression tree. Going from axial to oblique splits adds complexity to the proposal of candidate splits. However, our split proposer dramatically reduces the number of proposed split candidates for the nodes evaluated during the construction of the tree, and by virtue of that fact spends much less time evaluating scores of the candidates.

4 Spectral Splitting

This section will start with a brief description of spectral clustering, followed by details about how we apply this method to produce candidate splits for an ART time-series model. A good tutorial treatment and an extensive list of references for spectral clustering can be found in [26].

The spectral splitting method that we propose constructs two types of split candidates–oblique and time–both relying on spectral clustering. Based on this clustering, the method applies two different views on the data–phase and trace–according to the type of splits we want to identify. The algorithm will only propose a *single* oblique split candidate and possibly a few time split candidates for any node evaluated during the construction of the regression tree.

4.1 Spectral Clustering

Given a set of n multi-dimensional data points $(x_1, \ldots, x_n)$, we let $a_{ij} = a(x_i, x_j)$ denote the affinity between the i'th and j'th data point, according to some symmetric and non-negative measure. The corresponding affinity matrix is denoted by$A =$

$(a_{ij})_{i,j=1,\dots,n}$, and we let D denote the diagonal matrix with values $\sum_{j=1}^{n} a_{ij}$, $i = 1,\dots,n$ on the diagonal.

Spectral clustering is a non-parametric clustering method that uses the pairwise proximity between data points as a basis of the criterion that the clustering must optimize. The trick in spectral clustering is to enhance the cluster properties in the data by changing the representation of the multi-dimensional data into a (possibly one-dimensional) representation based on eigenvalues for the so-called Laplacian.

$$L = D - A$$

Two different normalizations for the Laplacian have been proposed in [23] and [21], leading to two slightly different spectral clustering algorithms. We will follow a simplified version of the latter. Let I denote the identity matrix. We will cluster the data according to the second smallest eigenvector–the so-called Fiedler vector [9]–of the normalized Laplacian

$$L_{norm} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}$$

The algorithm is illustrated in Figure 1. Notice that we replace L_{norm} with

$$L'_{norm} = I - L_{norm}$$

which changes eigenvalues from λ_i to $1 - \lambda_i$ and leaves eigenvectors unchanged. We therefore find the eigenvector for the second-largest and not the second-smallest eigenvector. We prefer this interpretation of the algorithm for reasons that become clear when we discuss iterative methods for finding eigenvalues in Section 4.2.

1. Construct the matrix L'_{norm}.
2. Find the second-largest eigenvector $e = (e_1,\dots,e_n)$ of L'_{norm}.
3. Cluster the elements in the eigenvector (e.g. by the largest gap in values).
4. Assign the original data point x_i to the cluster assigned to e_i.

Fig. 1. Simple normalized spectral clustering algorithm

Readers familiar with the original algorithm in [21] may notice the following simplifications: First, we only consider a binary clustering problem, and second, we only use the two largest eigenvectors for the clustering, and not the k largest eigenvectors in their algorithm. (The elements in the first eigenvector always have the same value and will therefore not contribute to the clustering.) Due to the second simplification, the step in their algorithm that normalizes rows of stacked eigenvectors can be avoided, because the constant nature of the first eigenvector leaves the transformation of the second eigenvector monotone.

4.2 Oblique Splits

Oblique splits are based on a particular view of the time series data that we call the *phase view*, as defined in Section 3. Importantly, a data case in the phase view involves

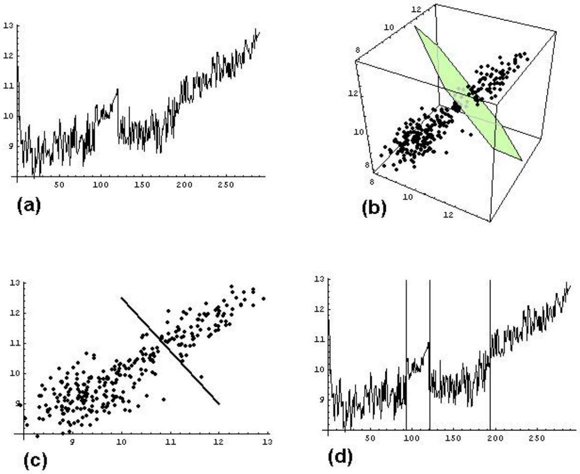


Fig. 2. Oblique split candidate for ART model with two autoregressors. (a) The original time series. (b) The spectral clustering of phase-view data. The polygon separating the upper and lower parts is a segment of a separating hyperplane for the spectral clusters (c) The phase view projection to regressor plane and the separating hyperplane learned by the perceptron algorithm. (d) The effect of the oblique split on the original time series: a regime consisting of the slightly less upward trending and more volatile first and third data segments is separated from the regime with more upward trending and less volatile second and fourth segments.

values for both the target and regressors, which imply that our oblique split proposals may capture regression structures that show up in the data–as opposed to many standard methods for axial splits that are ignorant to the target when determining split candidates for the regressors.

It should also be noted that because the phase view has no notion of time, similar patterns from entirely different segments of time may end up on the same side of an oblique split. This property can at times result in a great advantage over splitting the time series into chronological segments. First of all, splitting on time imposes a severe constraint on predictions, because splits in time restrict the prediction model to information from the segment latest in time. Information from similar segments earlier in the time series are not integrated into the prediction model in this case. Second, we may need multiple time splits to mimic the segments of one oblique split, which may not be obtainable due to the degradation of the statistical power from the smaller segments of data. Figure 2(d) shows an example, where a single oblique split separates the regime with the less upward trending and slightly more volatile first and third data segments of the time series from the regime consisting of the less volatile and more upward trending second and fourth segments. In contrast, we would have needed three time splits to properly divide the segments and these splits would therefore have resulted in four different regimes.

Our split proposer produces a single oblique split candidate in a two step procedure. In the first step, we strive to separate two modes that relates the target and regressors for the model in the best possible way. To accomplish this task, we apply the affinity based spectral clustering algorithm, described in Section 4.1, to the phase view of the time series data. For the experiments reported later in this paper, we use an affinity measure proportional to

$$\frac{1}{1 + ||p_1 - p_2||^2}$$

where $||p_1 - p_2||^2$ is the L2-norm between two phases. We do not consider exogenous regressors from related time series in these experiments. All variables in the phase view are therefore on the same scale, making the inverse distance a good measure of proximity. With exogenous regressors, more care should be taken with respect to the scaling of variables in the proximity measure, or the time series should be standardized. Figure 2(b) demonstrates the spectral clustering for the phase view of the time-series data in Figure 2(a), where this phase view has been constructed for an ART model with two autoregressors.

The oblique split predicate in (1) defines an inequality that only involves the regressors in the model. The second step of the oblique split proposer therefore projects the clustering of the phase view data to the space of the regressors, where the hyperplane separating the clusters is now constructed. While this can be done with a variety of linear discrimination methods, we decided to use a simple single-layer perceptron optimizing the total misclassification count. Such perceptron will be relatively insensitive to outliers, compared to, for example, Fisher's linear discriminant.

The computational complexity of an oblique split proposal is dominated by the cost of computing the full affinity matrix, the second largest eigenvector for the normalized Laplacian, and finding the separating hyperplane for the spectral clusters. Recall that n denotes the number of cases in the phase view of the data. The cost of computing the full affinity matrix is therefore $O(n^2)$ affinity computations. Direct methods for computing the second largest eigenvector is $O(n^3)$. A complexity of $O(n^3)$ may be prohibitive for series of substantial length. Fortunately, there are approximate iterative methods, which in practice are much faster with tolerant error. For example, the Implicitly Restarted Lanczos Method (IRLM) has complexity $O(mh + nh)$, where m is the number of nonzero affinities in the affinity matrix and h is the number of iterations required until convergence [29]. With a full affinity matrix $m = n^2$, but a significant speedup can be accomplished by only recording affinities above a certain threshold in the affinity matrix. Finally, the perceptron algorithm has complexity $O(nh)$.

4.3 Time Splits

A simple but computationally expensive way of determining a good time split is to let the split proposer nominate all possible splits in time for the further evaluation. The commercial implementation of the ART models relies on an approximation to this approach that proposes a smaller set of equispaced points on the time axis.

We suggest a data driven approximation, which will more precisely target the change points in the time series. Our approach is based on another view of the time series data that we call the *trace view*. In the trace view we use the additional time information to label the phase view data in the spectral clustering. The trace view, now traces the clustered data through time and proposes a split point each time the trace jumps across clusters. The rationale behind our approach is that data in the same cluster will behave in a similar way, and we can therefore significantly reduce the number of time-split proposals by only proposing the cluster jumps. As an example, the thin lines orthogonal to the time axis in Figure 2(d) shows the few time splits proposed by our approach. Getting close to a good approximation for the equispaced approach would have demanded far more proposed split points.

Turning now to the computational complexity. Assuming that spectral clustering has already been performed for the oblique split proposal, the additional overhead for the trace through data is $O(n)$.

5 Evaluation

In this section, we provide an empirical evaluation for our spectral splitting methods. We use a large collection of financial trading data. The collection contains the daily closing prices for 1495 stocks from Standard & Poor's 1500 index[3] as of January 1, 2008. Each time series spans across approximately 150 trading days ending on February 1, 2008. (Rotation of stocks in the S&P 1500 lead to the exclusion of 5 stocks with insuffient data.) The historic price data is available from Yahoo!, and can be downloaded with queries of format http://finance.yahoo.com/q/hp?s=SYMBOL, where SYMBOL is the symbol for the stock in the index. We divide each data set into a training set, used as input to the learning method, and a holdout set, used to evaluate the models. We use the last five observations as the holdout set, knowing that the data are daily with trading weeks of five days.

In our experiments, we learn ART models with an arbitrary number of autoregressors and we allow time as an exogenous split variable. We do not complicate the experiments with the use of exogenous regressors from related time series, as this complication is irrelevant to the objective for this paper. For all the models that we learn, we use the same Bayesian scoring criterion, the same greedy search strategy for finding the number of autoregressors, and the same method for constructing a regression tree – except that different alternative split candidates are considered for the different splitting algorithms that we consider.

We evaluate two different types of splitting with respect to the autoregressors in the model: *AxialGaussian* and *ObliqueSpectral*. The AxialGaussian method is the standard method used to propose multiple axial candidates for each split in an ART model, as described in Section 2.1. The ObliqueSpectral method is our proposed method, which for a split considers only a single oblique candidate involving all regressors. In combination with the two split proposer methods for autoregressors, we also evaluate three types of time splitting: *NoSplit*, *Fixed*, and *TimeSpectral*. The NoSplit method does not allow any time splits. The Fixed method is the simple standard method for learning splits on time in an ART model, as described in Section 4.3. The TimeSpectral method is our spectral clustering-based alternative. In order to provide context for the numbers in the evaluation of these methods, we will also evaluate a very weak baseline, namely the method not allowing any splits. We call this method the *Baseline* method.

We evaluate the quality of a learned model by computing the *sequential predictive score* for the holdout data set corresponding to the training data from which the model

[3] standardandpoors.com

was learned. The sequential predictive score for a model is simply the average log-likelihood obtained by a one-step forecast for each of the observations in the holdout set. To evaluate the quality of a learning method, we compute the average of the sequential predictive scores obtained for each of the time series in the collection. Note that the use of the log-likelihood to measure performance simultaneously evaluates both the accuracy of the estimate and the accuracy of the uncertainty of the estimate. Finally, we use a (one-sided) sign test to evaluate if one method is significantly better than another. To form the sign test, we count the number of times one method improves the predictive score over the other for each individual time series in the collection. Excluding ties, we seek to reject the hypothesis of equality, where the test statistic for the sign test follows a binomial distribution with probability parameter 0.5.

6 Results

To make sure that the results reported here are not an artifact of sub-optimal axial splitting for the AxialGaussian method, we first verified the claim from [5] that the Gaussian quantiles is a sufficient substitute for the exhaustive set of possible axial splits. We compared the sequential predictive scores on 10% of the time series in our collection and did not find a significant difference.

Table 1 shows the average sequential predictive scores across the series in our collection for each combination of autoregressor and time-split proposer methods. First of all, for splits on autoregressors, we see a large improvement in score with our ObliqueSpectral method over the standard AxialGaussian method. Even with the weak baseline–namely the method not allowing any splits–the relative improvement from AxialGaussian to ObliqueSpectral over the improvement from the baseline to AxialGaussian is still above 20%, which is quite impressive.

The fractions in Table 2 report the number of times one method has higher score than another method for all the time series in our collection. Notice that the numbers in a fraction do not necessarily sum to 1495, because we are not counting ties. We particularly

Table 1. Average sequential predictive scores for each combination of autoregressor and time split proposer methods

Regressor splits	Time splits	Ave. score
Baseline	Baseline	-3.07
AxialGaussian	NoSplit	-1.73
AxialGaussian	Fixed	-1.72
AxialGaussian	TimeSpectral	-1.74
ObliqueSpectral	NoSplit	-1.45
ObliqueSpectral	Fixed	-1.46
ObliqueSpectral	TimeSpectral	-1.44

Table 2. Pairwise comparisons of sequential predictive scores. The fractions show the number of time series, where one method has higher score than the other. The column labels denote the autoregressor split proposers being compared.

	Baseline / AxialGaussian	Baseline / ObliqueSpectral	AxialGaussian / ObliqueSpectral
NoSplit	118 / 959	74 / 1168	462 / 615
Fixed	114 / 990	79 / 1182	226 / 418
SpectralTime	122 / 955	71 / 1171	473 / 604

notice that the ObliqueSpectral method is significantly better than the standard AxialGaussian method for all three combinations with time-split proposer methods. In fact, the sign test rejects the hypothesis of equality at a significance level $< 10^{-5}$ in all cases. Combining the results from Tables 1 and 2, we can conclude that the large improvement in the sequential predictive scores for our ObliqueSpectral method over the standard AxialGaussian method is due to a general trend in scores across individual time series, and not just a few outliers.

We now turn to the surprising observation that adding time-split proposals to either of the AxialGaussian and the ObliqueSpectral autoregressor proposals does not improve the quality over models learned without time splits–neither for the Fixed nor the TimeSpectral method. Apparently, the axial and oblique splitting on autoregressors are flexible enough to cover the time splits in our analysis. We do not necessarily expect this finding to generalize beyond series that behave like stock data, due to the fact that it is a relatively easy exercise to construct an artificial example that will challenge this finding.

Finally, the oblique splits proposed by our method involve *all* regressors in a model, and therefore rely on our spectral splitting method to be smart enough to ignore noise that might be introduced by irrelevant regressors. Although efficient, such parsimonious split proposal may appear overly restrictive compared to the possibility of proposing split candidates for all possible subsets of regressors. However, an additional set of experiments have shown that the exhaustive approach in general only leads to insignificant improvements in predictive scores. We conjecture that the stagewise inclusion of regressors in the overall learning procedure for an ART model (see Section 3) is a main reason for irrelevant regressors to not pose much of a problem for our approach.

7 Conclusions and Future Work

We have presented a method for building regime-switching trees for nonstationary time series. The method is based on geometric clustering. More specifically, spectral clustering has been used in this paper. As such, our method does not rely on any parametric assumptions with regards to the distributions that best describe individual regimes. The clustering-based split proposer is used to propose a single oblique split candidate at each node level in the switching tree, which makes the method computationally efficient.

In the evaluation part of the paper we limited ourselves to an extension of ART models that are built under the assumption of uncorrelated Gaussian error. The joint target-regressor distribution for a regime-switching time series can be modeled as a mixture of Gaussians in this case, and we were able to motivate and then prove empirically that oblique splits are better at learning the mixtures than combinations of axial splits. In fact, the experimental evidence we have collected shows that our approach when used to extend the ART models, dramatically improves predictive accuracy over the current approach. We still experimented under the assumption of Gaussianity. An important future experiment should allow non-Gaussian models in the oblique switching trees.

The focus in this paper has been on learning regime-switching time-series models that will easily lend themselves to explanatory analysis and interpretation. In future experiments we also plan to evaluate the potential tradeoff in modularity, interpretability, and computational efficiency with forecast precision for our simple learning approach compared to more complicated approaches that integrates learning of soft regime switching and the local regimes in the models, such as the learning of Markov-switching (e.g., [12,13]) and gated experts (e.g., [27,28]) models.

References

1. Bishop, C.M.: Neural Networks for Pattern Recognition. Oxford University Press, Oxford (1995)
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group, Belmont, California (1984)
3. Brodley, C.E., Utgoff, P.E.: Multivariate decision trees. Machine Learning 19(1), 45–77 (1995)
4. Chaudhuri, P., Huang, M., Loh, W.Y., Yao, R.: Piecewise polynomial regression trees. Statistica Sinica 4, 143–167 (1994)
5. Chickering, D., Meek, C., Rounthwaite, R.: Efficient determination of dynamic split points in a decision tree. In: Proc. of the 2001 IEEE International Conference on Data Mining, pp. 91–98. IEEE Computer Society (November 2001)
6. DeGroot, M.: Optimal Statistical Decisions. McGraw-Hill, New York (1970)
7. Dobra, A., Gehrke, J.: Secret: A scalable linear regression tree algorithm. In: Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 481–487. ACM Press (2002)
8. Donath, W.E., Hoffman, A.J.: Lower bounds for the partitioning of graphs. IBM Journal of Research and Development 17, 420–425 (1973)
9. Fiedler, M.: Algebraic connectivity of graphs. Czechoslovak Mathematical Journal 23, 298–305 (1973)
10. Gama, J.: Oblique linear tree. In: Proc. of the Second International Symposium on Intelligent Data Analysis, pp. 187–198 (1997)
11. Hamilton, J.D.: Time Series Analysis. Princeton University Press, Princeton (1994)
12. Hamilton, J.D.: A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica 57(2), 357–384 (1989)
13. Hamilton, J.D.: Analysis of time series subject to changes in regime. Journal of Econometrics 45, 39–70 (1990)
14. Iyengar, V.S.: Hot: Heuristics for oblique trees. In: Proc. of the 11th IEEE International Conference on Tools with Artificial Intelligence, pp. 91–98. IEEE Computer Society, Washington, DC (1999)

15. Jordan, M.I., Jacobs, R.A.: Hierarchical mixtures of experts and the EM algorithm. Neural Computation 6, 181–214 (1994)
16. Li, K.C., Lue, H.H., Chen, C.H.: Interactive tree-structured regression via principal Hessian directions. Journal of the American Statistical Association 95, 547–560 (2000)
17. Mandelbrot, B.: Forecasts of future prices, unbiased markets, and martingale models. Journal of Business 39, 242–255 (1966)
18. Meek, C., Chickering, D.M., Heckerman, D.: Autoregressive tree models for time-series analysis. In: Proc. of the Second International SIAM Conference on Data Mining, pp. 229–244. SIAM (April 2002)
19. Meilă, M., Shi, J.: Learning segmentation by random walks. In: Advances in Neural Information Processing Systems 13, pp. 873–879. MIT Press (2001)
20. Murthy, S.K., Kasif, S., Salzberg, S.: A system for induction of oblique decision trees. Journal of Artificial Intelligence Research 2, 1–32 (1994)
21. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14, pp. 849–856. MIT Press (2002)
22. Samuelson, P.: Proof that properly anticipated prices fluctuate randomly. Industrial Management Review 6, 41–49 (1965)
23. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
24. Tong, H.: Threshold models in non-linear time series analysis. Lecture Notes in Statistics, vol. 21. Springer (1983)
25. Tong, H., Lim, K.S.: Threshold autoregression, limit cycles and cyclical data- with discussion. Journal of the Royal Statistical Society, Series B 42(3), 245–292 (1980)
26. von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
27. Waterhouse, S., Robinson, A.: Non-linear prediction of acoustic vectors using hierarchical mixtures of experts. In: Advances in Neural Information Processing Systems 7, pp. 835–842. MIT Press (1995)
28. Weigend, A.S., Mangeas, M., Srivastava, A.N.: Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. International Journal of Neural Systems 6(4), 373–399 (1995)
29. White, S., Smyth, P.: A spectral clustering approach to finding communities in graphs. In: Proc. of the 5th SIAM International Conference on Data Mining. SIAM (2005)

Appendix

Lemma 1. *Let Σ be a non-singular sample auto-covariance matrix for X_{t-p}^{t-1} defined on the p-dimensional space with principal diagonal direction $d = (\frac{1}{\sqrt{p}}, \ldots, \frac{1}{\sqrt{p}})$, and let $A_t = \frac{1}{p}\sum_{i=1}^{p} X_{t-i}$. Then*

$$\sin^2(\Sigma d, d) = \frac{\sum_{i=1}^{p} cov(X_{t-i} - A_t, A_t)^2}{\sum_{i=1}^{p} cov(X_{t-i}, A_t)^2}. \tag{2}$$

Proof. Introduce $S_t = \sum_{i=1}^{p} X_{t-i}$. As per bi-linear property of covariance, $(\Sigma d)_i = \frac{1}{\sqrt{p}} cov(X_{t-i}, S_t), i = 1, \ldots, p$ and $(\Sigma d)d = \frac{1}{p}\sum_{i=1}^{p} cov(X_{t-i}, S_t) = cov(A_t, S_t)$. Non-singularity of Σ implies that the vector $\Sigma d \neq 0$. Hence, $|\Sigma d|^2 \neq 0$ and

$$\cos^2(\Sigma d, d) = \frac{((\Sigma d)d)^2}{|(\Sigma d)|^2} = \frac{p\, cov(A_t, S_t)^2}{\sum_{i=1}^{p} cov(X_{t-i}, S_t)^2}.$$

It follows that

$$\begin{aligned}
&\sin^2(\Sigma d, d)\\
&= 1 - \cos^2(\Sigma d, d)\\
&= \frac{p\left(\frac{1}{p}\sum_{i=1}^{p} cov(X_{t-i}, S_t)^2 - cov(A_t, S_t)^2\right)}{\sum_{i=1}^{p} cov(X_{t-i}, S_t)^2}\\
&= \frac{\sum_{i=1}^{p} cov(X_{t-i} - A_t, S_t)^2}{\sum_{i=1}^{p} cov(X_{t-i}, S_t)^2}
\end{aligned}$$

Dividing the numerator and denominator of the last fraction by p^2 amounts to replacing S_t by A_t, which concludes the proof. □

Corollary 1. *When $X_{t-i} - A_t$ and A_t are weakly correlated, and the variance of $X_{t-i} - A_t$ is comparable to that of A_t, $i = 1, \ldots, p$, then $\sin^2(\Sigma d, d)$ is small.*

Specifically, let σ and ρ denote respectively standard deviation and correlation, and introduce $\Delta_i = \frac{cov(X_{t-i} - A_t, A_t)}{\sigma(A_t)} = \rho(X_{t-i} - A_t, A_t)\sigma(X_{t-i} - A_t)$. We quantify both assumptions in Corollary 1 by positing that $|\Delta_i| < \epsilon\sigma(A_t), i = 1, \ldots, p$, where $0 < \epsilon \ll 1$. Easy algebra on Equation (2) yields

$$\begin{aligned}
\sin^2(\Sigma d, d) &= \frac{\Sigma \Delta_i^2}{\Sigma(\sigma(A_t) + \Delta_i)^2}\\
&< \frac{p\epsilon^2\sigma(A_t)^2}{p(1-\epsilon)^2\sigma(A_t)^2}\\
&= \frac{\epsilon^2}{(1-\epsilon)^2} \qquad (3)
\end{aligned}$$

Under the assumptions of Corollary 1, we can now show that d is geometrically close to an eigenvector of Σ. Indeed, by inserting (3) into the Pythagorean identity we derive that $|\cos(\Sigma d, d)| > \frac{\sqrt{1-2\epsilon}}{1-\epsilon}$ and close to 1. Now, given a vector v for which $|v| = 1$, $|\cos(\Sigma v, v)|$ reaches the maximum of 1 iff v is an eigenvector of Σ. When the eigenvalues of Σ are distinct, d must therefore be at a small angle with one of the p principal axes for Σ.

A Novel Framework for Nontechnical Losses Detection in Electricity Companies

Matías Di Martino, Federico Decia, Juan Molinelli, and Alicia Fernández

Instituto de Ingeniería Eléctrica,
Facultad de Ingeniería Universidad de la República Montevideo, Uruguay
{matiasdm,alicia}@fing.edu.uy,
{federicodecia,jmolinelli}@gmail.com

Abstract. Nontechnical losses represent a very high cost to power supply companies, who aims to improve fraud detection in order to reduce this losses. The great number of clients and the diversity of different types of fraud makes this a very complex task. In this paper we present a combined strategy based on measures and methods adequate to deal with class imbalance problems. We also describe the features proposed, the selection process and results. Analysis over consumers historical kWh load profile data from Uruguayan Electricity Utility (UTE) shows that using combination and balancing techniques improves automatic detection performance.

Keywords: Electricity theft, Support vector machine, Optimum path forest, Unbalance class problem, Combining classifier, UTE.

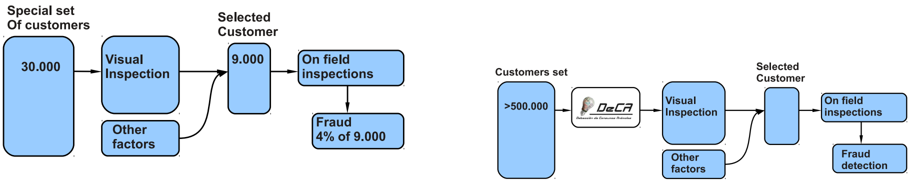
1 Introduction

Improving nontechnical loss detection is a huge challenge for electric companies. Research in pattern classification field has been made to tackle this problem [25], [21], [20], [17]

In Uruguay the national electric power utility (henceforth call UTE) faces the problem by manually monitoring a group of customers. The procedure is ilustrated in the figure 1(a). Agroup of experts looks at the monthly consumption curve of each customer and indicates those with some kind of suspicious behavior. This set of customers, initially classified as suspects are then analyzed taking into account other factors (such as fraud history, counter type etc.). Finally a subset of customers is selected to be inspected by an UTE employee, who confirms (or not) the irregularity. The procedure described before, has major drawbacks, mainly, the number of costumers that can be manually controlled is small compared with the total amount of costumer (around 500.000 only in Montevideo). To improve the efficiency of fraud detection and resource utilization, we implemented a tool that automatically detects suspicious behavior analyzing customers historical consumption curve. Thus, UTE's experts only need to look to a reduced number of costumers and then select those who need to be inspected, as is ilustrated in the figure 1(b)

Due to the applications nature there is a great imbalance between "normal" and "fraud/suspicious" classes. The class imbalance problem in general and fraud detection in particular have received considerable attention in recent years. Garcia et al. and

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 109–120.
DOI: 10.1007/978-3-642-36530-0_9



(a) Previous fraud detection scheme

(b) Actual fraud detection scheme

Fig. 1.

Guo and Zhou review main topics in the field of the class imbalance problem [15], [14], [16]. These include: resampling methods for balancing data sets [3],[2], [7], [8], [18], feature extraction and selection techniques -wrapper [10], and choose of F-value as performance measure.

In addition, it is generally accepted that combination of diverse classifiers can improve performance. A difficult task is to choose the combination strategy for a diverse set of classifiers. Kuncheva found the optimum set of weights for the majority weight vote combiner when the performance metrics is accuracy and with independent base classifiers [19]. Further analysis has been done on the relationship between diversity and the majority rules performance [4], [28], [9]. In this paper we propose a combination function adapted to the imbalance between classes, using F-value as the performance measurement and some well-known pattern recognition techniques such as SVM (Support Vector Machine) [27], [26], Tree classifiers and more recent algorithms such as Optimum Path Forest [22],[24] as base classifiers.

Performance evaluation using test dataset shows very good results on suspicious profiles selection. Also, on field evaluation of fraud detection using our automatic system shows similar results to manual experts' method.

This paper is an extension of our previous work presented in the International Conference on Pattern Recognition Application and Methods (ICPRAM 2012) [11], including some new and deeper analysis and some suggestions received in the conference presentation. The paper is organized as follows. Section 2 describes general aspects of the class imbalance problem, section 3 describes different strategies proposed, section 4 presents the results obtained, and, finally, section 5 concludes the work.

2 The Class Imbalance Problem

When working on the fraud detection problem, one can not assume that the number of people who commit fraud are the same than those who do not, usually there are fewers elements from the class who commit fraud. This situation is known as the problem of class imbalance, and it is particularly important in real world applications where it is costly to misclassify examples from the minority class. In this cases, standard classifiers tend to be overwhelmed by the majority class and ignore the minority class, hence obtaining suboptimal classification performance. Having to confront this type of problem, we decided to use three different strategies on different levels, changing class distribution by resampling, manipulating classifiers, and on the ensemble of them.

The first consists mainly in resampling techniques such as under-sampling the majority class or over-sampling the minority one. Random under-sampling aims at balancing the data set through random removal of majority class examples. The major problem of this technique is that it can discard potentially important data for the classification process. On the other hand, the simplest over-sampling method is to increase the size of the minority class by random replication of those samples. The main drawback of over-sampling is the likelihood of over-fitting, since it makes exact copies of the minority class instances As a way of facing the problems of resampling techniques discussed before, different proposals address the imbalance problem by adapting existing algorithms to the special characteristics of the imbalanced data sets. One approach is one-class classifiers, which tries to describe one class of objects (target class) and distinguish it from all other objects (outliers). In this paper, the performance of One-Class SVM, adaptation of the popular SVM algorithm, will be analyzed. Another technique is cost-sensitive learning, where the cost of a particular kind of error can be different from others, for example by assigning a high cost to mislabeling a sample from the minority class.

Another problem which arises when working with imbalanced classes is that the most widely used metrics for measuring the performance of learning systems, such as accuracy and error rate, are not appropriate because they do not take into account misclassification costs, since they are strongly biased to favor the majority class ([14]) . In the past few years, several new metrics which measure the classification performance on majority and minority classes independently, hence taking into account the class imbalance, have been proposed [5].

- $Recall^p = \dfrac{TP}{TP + FN}$
- $Recall^n = \dfrac{TN}{TN + FP}$
- $Precision = \dfrac{TP}{TP + FP}$
- $F_{value} = \dfrac{(1+\beta^2) Recall^p \times Precision}{\beta^2 \; Recall^p + Precision}$

Table 1. Confusion matrix

	Labeled as	
	Positive	Negative
Positive	TP (True Positive)	FN (False Negative)
Negative	FP (False Positive)	TN (True Negative)

$Recall^p$ is the percentage of correctly classified positive instances, in this case, the fraud samples. Precision is defined as the proportion of labeled as positive instances that are actually positive. The combination of this two measurements, the F-value, represents the geometric mean between them, weighted by the parameter β. Depending on the value of β we can prioritize Recall or Precision. For example, if we have few resources to perform inspections, it can be useful to prioritize Precision, so the set of samples labeled as positive has high density of true positive.

3 Strategy Proposed

The system presented consists of basically on three modules: Pre-Processing and Normalization, Feature selection and extraction and, finally, Classification. Figure 2 shows the system configuration. The system input corresponds to the last three years of the

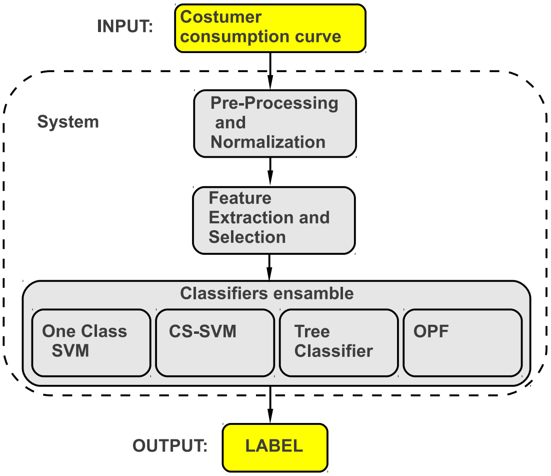

INPUT:
Costumer consumption curve
System
Pre-Processing and Normalization
Feature Extraction and Selection
Classifiers ensamble
One Class SVM
CS-SVM
Tree Classifier
OPF
OUTPUT:
LABEL

Fig. 2. Block Diagram

monthly consumption curve of each costumer, here called $X^m = \{x_1^m, \ \ldots \ x_n^m\}$, where x_i^m is the consumption of the m costumer during the i-th month. The first module called Pre-Processing and Normalization, normalizes the input data so that they all have unitary mean and implements some filters to avoid peaks from billing errors.

The proposed methodology was developed as GUI software in Matlab using PRTOOLS [13], LibOPF [23] and LibSVM [6].

3.1 Features

A feature set was proposed taking into account UTEs technician experts in fraud detection by manual inspection and recent papers on non technical loss detection [1], [20], [21]. To represents samples in some convenient space we meet several times UTEs experts in order to understand what they look for, when inspecting some customer consumption curve.

Below a list of the proposed features:

- Consumption ratio for the last 3, 6 and 12 months and the average consumption.
- Norm of the difference between the expected consumption and the actual consumption. The expected consumption value, is calculated taking into account the same month of the previous year multiplied by the ratio between the mean consumption of each year.
- Difference between Fourier coefficients from the last and previous years.
- Difference between Wavelet coefficients from the last and previous years.
- Difference in the coefficients of the polynomial that best fits the consumption curve.

All the above features compare the actual behavior with the past behavior for each customer. The idea is to identify changes in the behavior that could be associated to irregular situations. But, imagine that some customer is stealing since long time ago, then above features, will not show any change in the behavior. For these reason, we consider more features that compare customers curves with the other customers in the data set as:

- Euclidean distance of each customer to the *mean customer*, where the *mean customer* is calculated by taking the mean for each month between all the customers.
- Variance of the consumption curve.
- Module of the first five Fourier coefficients.
- Slope of the straight line that fits the consumption curve.

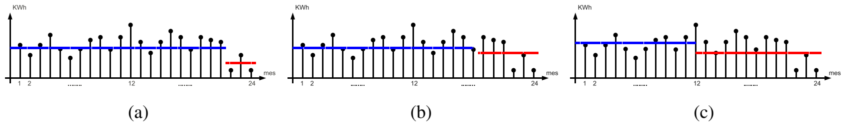
3.2 Features Selection

It is well known that when thinking about the features to use, large number of attributes do not imply better performances. The important thing is their relevance and the relationship between the number of these and the number of elements. This is why we implemented a feature selection stage. We implemented some algorithms for feature selection and look for those subsets of features that bests better for each classifer algorithm.

Evaluation Methods Used

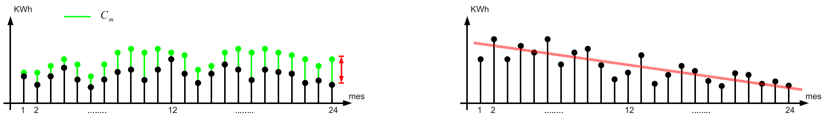
We used two types of evaluation methods: filter and wrapper. Filters methods looks for subsets of features with low correlation between them and high correlation with the labels, while wrapper methods evaluate the performance of a given classifier for the given subset of features.
In the wrapper methods, we used as performance measure the F-value, also, the evaluations were performed using 10 fold cross validation over the training set.
As searching method, we used *Bestfirt*, for which we found in this application a good balance between performance and computational costs.
Some of the features purposed, were selected most of the times for all the classifiers, for example:

1. Consumption ratio for the last 3 months and the average consumption (illustrated in Figured 3(a)).
2. Consumption ratio for the last 6 months and the average consumption (illustrated in Figured 3(b)).
3. Consumption ratio for the last 12 months and the average consumption (illustrated in Figured 3(c)).



(a) (b) (c)

Fig. 3. Features Selected



(a) (b)

Fig. 4. Features Selected

4. Euclidean distance of each customer to the *mean customer* (illustrated in Figured 4(a)).
5. Slope of the straight line that fits the consumption curve (illustrated in Figured 4(b)).
6. Some of the wavelets coefficients considered.

3.3 Classifiers

SVM is an algorithm frequently used in pattern recognition and fraud detection. The main purpose of the binary SVM algorithm is to construct an optimal decision function $f(x)$ that predicts unseen data into two classes and minimizes the classification error. In order to obtain this, one looks to maximize the separation margin between the two classes and hence classify correctly unseen data [21]. This can be formulated as a quadratic programming optimization problem

$$\Phi(\omega, \zeta_i) = \min \left\{ \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{n} \zeta_i \right\} \tag{1}$$

subjected to the constraint that all the training samples are correctly classified, that is

$$y_i(\langle \omega, x \rangle \ + b) \geq 1 - \zeta_i, \quad i = 1, 2, ..., n \tag{2}$$

where ζ_i for $i = 1, 2, ..., n$ are nonnegative slack variables. C is a regularization parameter and is selected to be the tradeoff between the two terms in 1.

CS-SVM and One-Class SVM. Two different approaches where introduced when describing the class imbalance problem, one-class classifiers and cost-sensitive learning. When applying this two approaches on SVM, we talk about One-Class SVM and CS-SVM.

In One-Class SVM equation 1 becomes,

$$\min_{\omega \in \mathcal{H}, \, \zeta_i \in \mathbb{R}, \, \rho \in \mathbb{R}} \frac{1}{2} \|\omega\|^2 + \frac{1}{\nu \, l} \sum_{i=1}^{n} \zeta_i - \rho \tag{3}$$

while in CS-SVM it becomes:

$$\Phi(\omega, \zeta_i) = \min \left\{ \frac{1}{2} \|\omega\|^2 + \sum_{i/y_i=1} C^+ \zeta_i + \sum_{i/y_i=-1} C^- \zeta_i \right\} \tag{4}$$

Both the kernel parameter K and the values of C^+, C^- and ω are often chosen using cross validation. The method consists in splitting the data set into p parts of equal size, and perform p training runs. Each time, leaving out one of the p parts and use it as an independent validation set for optimizing the parameters. Usually, the parameters which work best on average over the p runs are chosen. Finally, these average parameters are used to train the complete training set. There are some problems with this, as can be seen on [26].

Having said this, the method used to determine the optimum parameters for CS-SVM was:

1. Determine sets $C = [C_1, C_2, ..., C_n]$ and $\gamma = [\gamma_1, \gamma_2, ..., \gamma_m]$.
2. Select $C_i \in C$ and $\gamma_j \in \gamma$, split the training set into p parts of equal size and perform p training runs. Each set is called B_i with $i = \{1, 2, ..., p\}$.
3. Use $B_{te} = B_1$ as the test set and $B_{tr} = B_2 \cup B_3 \cup ... \cup B_p$ as the training set.
4. Determine a classifier model for B_{tr}, C_i and γ_j. As the ratio between the two classes is unbalanced, when determining the CS-SVM classifier two parameters are defined, C^+ and C^- using class weights defined by calculating the sample ratio for each class. This was achieved by dividing the total number of classifier samples with the individual class samples. In addition, class weights were multiplied by a factor of 100 to achieve satisfactory weight ratios [21].
5. Classify the samples from the training set B_{te} and compare the results with the labels predetermined. From these comparison, obtain the estimated F_{value} for C_i and γ_j called $F_{value_1}(C_i, \gamma_j)$.
6. Repeat these procedure for $B_{te} = B_2$ and the combination of the reaming sets as B_{tr} getting $e_2(C_i, \gamma_j)$, then for $B_{te} = B_3$ and so on until completing the p iterations.
7. For each pair of (C_i, γ_j) there's an estimation of the classification error for each cross validation. The classification error for this pair (C_i, γ_j) is the average value of the classification errors obtained in each cross validation, $e(C_i, \gamma_j) = \frac{1}{p} \sum e_l(C_i, \gamma_j)$.
8. This method is repeated combining all the values from the sets C and γ.
9. The values of C_{opt} and γ_{opt} are the ones for which the smallest classification error is obtained.

The metric used for measuring the classification error for this method was the F_{value}. For One-Class SVM, the method was the same but with the main objective of finding $\sigma \in S = \{\sigma_1, \sigma_2 ..... \sigma_l\}$.

OPF. In [25] a new approach, Optimum Path Forest (OPF), is applied to fraud detection in electricity consumption. The work shows good results in a problem similar to the targeted. OPF creates a graph with training dataset elements. A cost is associated to each path between two elements, based on the distance of the intermediate elements belonging to the path. It is assumed, that elements of the same class will have a lower path cost, than elements of different classes. The next step is to choose representatives from each class, called prototypes. Classifying a new element implies to find the prototype with lowest path cost. Since OPF is very sensitive to class imbalance, we under-sampled the majority class. Best performance was obtained while using a training data set with 40% of the elements from the minority class.

C4.5. The fourth classifier used is a decision tree proposed by Ross Quinlan: C4.5. Trees are a method widely used in pattern recognition problems due to its simplicity and good results. To classify, a sequence of simple questions is done. It begins with an initial question, and depending on the answer, the procedure continues until reaching a conclusion about the label to be assigned. The disadvantage of these methods is that they are very unstable and highly dependent on the training set. To fix this, in C4.5 a later stage of AdaBoost was implemented. It generates multiple instances of the tree with different portions of the training set and then combines them achieving a more robust result. As in OPF, sensitivity to class imbalance has led to sub-sampling the majority class. Again, we found that the best results was obtained while using a training data set with 40% of the elements from the minority class.

3.4 Combining Classifiers

The next step after selecting feature sets and adjusting classification algorithms to the training set, is to decide how to combine the information provided by each classifier. There are several reasons to combine classifiers, for example, to obtain a more robust and general solution and improve the final performance [12].

After labels have been assigned by each individual classifier, a decision rule is build as:

$$\begin{aligned} g_p(x) = {} & \lambda^p_{O-SVM}\, d^p_{O-SVM} + \lambda^p_{CS-SVM}\, d^p_{CS-SVM} \\ & + \lambda^p_{OPF}\, d^p_{OPF} + \lambda^p_{Tree}\, d^p_{Tree} \end{aligned} \tag{5}$$

$$\begin{aligned} g_n(x) = {} & \lambda^n_{O-SVM}\, d^n_{O-SVM} + \lambda^n_{CS-SVM}\, d^n_{CS-SVM} \\ & + \lambda^n_{OPF}\, d^n_{OPF} + \lambda^n_{Tree}\, d^n_{Tree} \end{aligned} \tag{6}$$

where $d^i_j(x) = 1$ if the classifier j labels the sample as i and 0 otherwise. Then if $g_p(x) > g_n(x)$ the sample is assigned to the positive class, if $g_n(x) > g_p(x)$ the sample is assigned to the negative class.

In [19], the weighted majority vote rule is analyzed and optimum weights are found for maximum overall accuracy, assuming independence between classifiers: $\lambda^i_j = log\left(\frac{Accuracy_j}{1-Accuracy_j}\right)$, where $Accuracy_j$ represents the ratio of correctly classified samples for the classifier j, (in [19] priors are also consider on the $g_{\{p,n\}}(x)$ construction adding $log(P(\omega_{\{p,n\}})))$

Inspired in this result, but taking into account that we want to find a solution with good balance between Recall and Precision, several weights $\lambda^{p,n}_j$ were proposed:

- $\lambda^i_j = log\left(\frac{Recall^p_j+1}{Recall^p_j-1}\right)$
- $\lambda^i_j = log\left(\frac{F_{value_j}+1}{F_{value_j}-1}\right)$
- $\lambda^i_j = log\left(\frac{Accuracy_j}{1-Accuracy_j}\right)$
- $\lambda^p_j = Recall^n_j$ and $\lambda^n_j = Recall^p_j$

Also the *optimal* multipliers were found by exhaustive search over a predefined grid, looking for those which maximize the classification F_{value}. Search was made by looking for all the possibilities with $\lambda^i_j \in [0 : 0.05 : 1]$ and was evaluated with a 10-fold cross validation.

All of the proposed combined classifiers improved individual classifiers performance. In Table 2 we present the performance results using *optimal* multipliers, found by exhaustive search.

4 Results

4.1 Data

For this paper we used a data set of 1504 industrial profiles (October 2004- September 2009) obtained from the Uruguayan electric power company (DATASET 1). Each profile is represented by the customers monthly consumption. UTE technicians make random profile selection and data labeling. Training and performance evaluation shown in Table 2 was done with DATASET 1. Another independent dataset (DATASET 2) of 3338 industrial profiles with contemporary data (January 2008-2011) was used for on field evaluation.

4.2 Labeling Results

Table 2 shows performance for individual classifiers and for the combination of them, results shown here were achieved by using a 10-fold cross validation using DATASET1. CS-SVM presented the best F_{value}, followed by One class SVM. We saw that combination improved performance achieving better results than those of the the best individual classifier.

Table 2. Data Set 1 labeling results

Description	$Acc.$ (%)	$Rec^p.$ (%)	$Pre.$ (%)	$Fval.$ (%)$[\beta = 1]$
O-SVM	84,9	54,9	50,8	52,8
CS-SVM	84,5	62,8	49,7	55,5
OPF	80,1	62,2	40,5	49
Tree (C4.5)	79	64,6	39	48,6
Combination	86,2	64	54,4	58,8

4.3 On Field Results

After all the proposed alternatives were evaluated (on DATASET 1), comparing automatic labelling with *manual labelling* performed by UTE's experts, we tested data labels with on field evaluation.

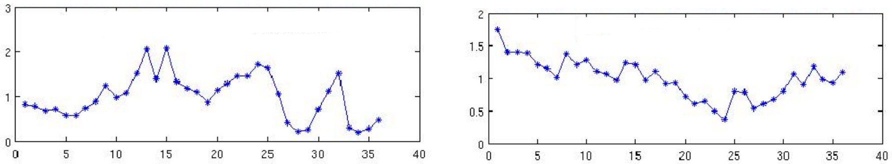


Fig. 5. Consumption Profiles

This test were done in the following way:

1. Train the classification algorithm using DATASET 1.
2. Classify samples from DATASET 2. Lets call DATASET 2P the samples of DATASET 2 labelled as positive (associated to abnormal consumption behavior).
3. Inspect customers on DATASET 2P

533 samples of DATASET 2 labelled as positive, were inspected by UTE's team. The inspections yielded 22 irregular situations. This results show that the automatic framework has a hit rate of 4.12%. Manual fraud detection performed by UTE's experts during 2010 had a hit rate of about 4%, so results are promising, specially taking into account that manual detection considers more information than just the consumption curve, such as fraud history, surface dimension and contracted power, among others.

Figures 5(a) and 5(b) show some examples of customers classified as suspicious by our automatic system. Once inspected, illegal activities were detected in these cases.

5 Conclusions

We developed a framework able to detect customers whose consumption behaviour show some kind of irregularities. UTE is beginning to incorporate the system proposed and first results showed that it is useful and can lead to important savings, both time and money. We will continue working with UTE's collaboration, focusing our investigation on the lines of:

- Improving final performance and monitor bigger customer sets aiming to reach all customers in Montevideo (Uruguayan capital city).
- Analyze existence of data clusters, i.e. to allow making more specific soluitions for the consumer with a similar kind of "normal" behavior. This has importance for the automatic analysis and also for the manual analysis.
- Add more features to our learning algorithm, such as: counter type (digital or analog), customer type (dwelling or industrial) and contracted power, among others.

We introduce different classifiers suitable for this type of problems (with unbalanced classes), comparing performance results for each of them. Innovative combination strategies are also proposed, all of them showing better results (using F-value as performance measurement) than the best individual classifier.

Acknowledgements. The authors would like to thank UTE, especially Juan Pablo Kosut, for providing datasets and share fraud detection expertise. We also want to thank Pablo Muse, Pablo Cancela and Martin Rocamora for fruitfull discussion.

References

1. Alcetegaray, D., Kosut, J.: One class svm para la detección de fraudes en el uso de energía eléctrica. Trabajo Final Curso de Reconocimiento de Patrones, Dictado por el IIE- Facultad de Ingeniería- UdelaR (2008)
2. Barandela, R., Garcia, V.: Strategies for learning in class imbalance problems. Pattern Recognition, 849–851 (2003)
3. Batista, G., Pratti, R., Monard, M.: A study of the behavior of several methods for balancing machine learning training data. SIGKDD Explorations 6, 20–29 (2004)
4. Brown, G., Kuncheva, L.: "Good" and "Bad" Diversity in Majority Vote Ensembles (2010)
5. Manning, C., Raghavan, P., Schutze, H.: An Introduction to Information Retrival, 1st edn. Cambridge University Press, Cambridge (2009)
6. Chang, C., Lin, C.: LIBSVM: a library for support vector machines (2001)
7. Chawla, N., Bowyer, K., Hall, L.: Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research (2002)
8. Chawla, N., Lazarevic, A., Hall, L.: Smoteboost: impoving prediction of the minority class in boosting. In: European Conf. of Principles and Practice of Knowledge Discovery in Databases (2003)
9. Chawla, N., Sylvester, J.: Exploiting diversity in ensembles: Improving the performance on unbalanced datasets. Departament of Computer Science and Engineering (2007)
10. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1, 131–156 (1997)
11. Di Martino, J., Decia, F., Molinelli, J., Fernámdez, A.: Improving electric fraud detection using class imbalance strategies. In: 1st International Conference in Pattern Recognition Aplications and Methods, vol. 2, pp. 135–141 (2012)
12. Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000)
13. Duin, R.: PRTools Version 3.0: A Matlab Toolbox for Pattern Recognition (2000)
14. Garcia, V., Sanchez, J., Mollineda, R.: On the suitability if numerical performance evaluation measures for class imbalance problems. In: 1st International Conference in Pattern Recognition Aplications and Methods, vol. 2, pp. 310–313 (2012)
15. Garcia, V., Sanchez, J., Mollineda, R., Alejo, R., Sotoca, J.: The class imbalance problem in pattern classification and learning (2007)
16. Guo, X., Zhou, G.: On the class imbalance problem. IIE - Computer Society 1, 192 (2008)
17. Jiang, R., Tagaris, H., Laschusz, A.: Wavelets based feature extraction and multiple cassifiers for electricity fraud detection (2000)
18. Kolez, A., Chowdhury, A., Alspector, J.: Data duplication: an imbalance problem? In: Proc. Proc. Intl. Conf. on Machine Learning, Workshop on Learning with Imbalanced Data Sets II (2003)
19. Kuncheva, L.: Combining Pattern Classifiers: Methods and Algorithms. Wiley-Interscience (2004)
20. Muniz, C., Vellasco, M., Tanscheit, R., Figueiredo, K.: Ifsa-eusflat 2009 a neuro-fuzzy system for fraud detection in electricity distribution (2009)
21. Nagi, J., Mohamad, M.: Nontechnical loss detection for metered customers in power utility using support vector machines. IEEE Transactions on Power Delivery 25(2) (2010)
22. Papa, J., Falcao, A.: Optimum-path forest: A novel and powerful framework for supervised graph-based pattern recognition techniques. Institute of Computing University of Campinas (2010)
23. Papa, J., Falcao, A., Suzuki, C.: LibOPF: a library for Opthimum Path Forets (2008)

24. Papa, J., Falcao, A., Miranda, P., Suzuki, C., Mascarenhas, N.: Design of robust pattern classifiers based on optimum-path forests. In: 8th International Symposium on Mathematical Morphology Rio de Janeiro Brazil, pp. 337–348 (October 2007)
25. Ramos, C., de Sousa, A.N., Papa, J., Falcao, A.: A new approach for nontechnical losses detection based on optimum-path forest. IEEE Transactions on Power Systems (2010)
26. Scholkopf, B., Smola, A.: Learning with Kernels, 2nd edn. The MIT Press, London (2002)
27. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
28. Wang, S., Yao, X.: Theoretical study of the relationship between diversity and single-class measures for class imbalance learning (2009)

Adaptive Features for Object Classification

Heydar Maboudi Afkham, Stefan Carlsson, and Josephine Sullivan

Computer Vision and Active Perception Lab., KTH, Stockholm, Sweden
{heydarma,stefanc,sullivan}@csc.kth.se
http://csc.kth.se/cvap

Abstract. This work presents a method for building adaptive local/semi-global features using a set of already extracted features. While for most methods local features are extracted independently of the task in hand, these features tend to change their representations in favor of different hypotheses to find the best possible representation. The features introduced in this paper take advantage of the part-based models at the feature level by combining the near by local features. This combination can either be local, which results in a more generic set of features suitable for bag-of-visual-words (BOVW) models or be semi-global, which results in a set of more object dependent features which are referred as parts. These representations capture the local variations around the local feature. At classification time, the best possible representation of these features is found and used in the calculations. This selection is done based on a latency defined at the feature level. The goal of this paper is to test how the adaptive features can improve the feature level likelihoods. The focus of the experiments of this paper is showing 1) how adaptive feature perform in BOVW scenarios and 2) how replacing single features with equivalent adaptive features improves the likelihoods obtained from them. The experiments of this paper are done on several classes of MSRCv1 and v2 datasets and it is shown that the method outperforms the baselines in all cases and the results are comparable to the state-of-the-art methods using BOVW models.

Keywords: Feature inference, Latent models, Clustering.

1 Introduction

Local features are considered to be the building blocks of many computer vision and machine learning methods and their quality highly effects the method's outcome. There are many popular methods for extracting local features from images. Among them one can name sift [10], hog[2] and haar[16] features, which are widely used for object detection and recognition [4,7] and texture features such as maximum response filter-banks [14] and MRF[13] which are used for texture recognition and classification. For most methods feature extraction is done independently from the method's task. For example in a normal inference problem a model tries to decide between two different classes. Usually for both hypothesises the same feature vector is fed to the model. In other words once the features are extracted, they remain constant through the whole process.

Computer vision methods deal with local features in different ways. Some such as boosting based object detectors [7,16] and markov random fields [6], depend on how

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 121–133.
DOI: 10.1007/978-3-642-36530-0_10

discriminative single features are and some, such as bag-of-visual-words [12], demand less discriminative features and depend on groups of features seen in a specific region on the image. For the methods that depend on the discriminative properties of local features the inference done at the feature level plays a critical role in the outcome of the method. Improving the quality of feature level inference can highly improve the quality of the object level inference done by most of such methods.

Many studies have shown that use of higher order statistics, *ex.* the joint relation between the features, can highly improve the quality of the features. Capturing joint relations is popular with the bag-of-words methods [9,8] since they deals with modeling joint relation between a finite number of data clusters. Unfortunately not many studies have focused on modeling joint relations in non-discretized data to create features that capture joint relations. A recent study on this matter is done by Morioka *et al.* [11]. In their study they introduced a mechanism for pairing two sift feature vectors together and creating a Local Pairwise Codebook by clustering them. As shown in their work the clusters produced using these joint features are more informative than the clusters produced using single features. The idea behind their work is similar to the work in this paper while the methodology of this work does not limit the number of feature vectors used in creating more complex features.

The method in this paper uses the assumption that a set of features are extracted from the image and a relation is known between them that can be captured by a graph. For example the features can come from several patches in the image and their spatial relation can be presented as a graph. These features can be extracted using any feature extraction method. The basic idea behind this paper is to use local features and their relations to introduce a new set of dynamic and changeable intermediate semi-global features in terms of latent variables. These intermediate features will be referred as feature clouds. The latent variables enable the feature to change its representation in different scenarios and their value is determined by an optimization procedure to make it more discriminative for learning algorithms. This dynamic property of feature cloud provides a good ground for introducing more discriminative features than the ones previously extracted from the image. The performance of these features is analyzed in two different experiments on *MSRC v1 and v2* [17] datasets. The first experiment deals with discriminative analysis of feature clouds and their inference at the feature level. These analysis show how more complex feature have an easier time locally identifying object regions in comparison more simple features. In the second experiment the features are employed in bag-of-visual-words model and it is shown that they have a better performance than the existing methods.

The outline of this paper is as follow. The related works to this paper are discussed in section 2. The feature clouds are discussed in detail in sections 3, 4 and 5. Finally section 6 discusses the behaviour of the feature clouds in some different scenarios.

2 Related Works

The goal of this paper is to present a method that takes advantage of part-based models at the feature level to come up with a set of intermediate features with discriminative properties. In this section a brief review of part based methods is provided and their

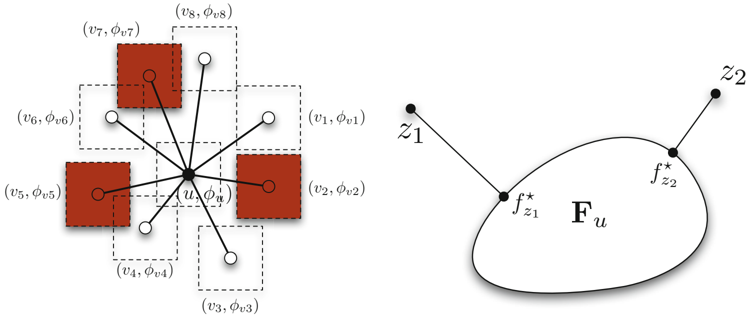


Fig. 1. (Left) A node u (a patch from an image) is connected to its neighboring nodes (close by patches). Here tree node are selected (red patches) as a latent configuration. The quantitative value of this configuration is calculated as $(\phi_u, \phi_{v_2}, \phi_{v_5}, \phi_{v_7})$.**(Right)** For any given point z in the feature space the closest vector within $\mathbf{F}_u$ is selected as $f^\star_z$ through an optimization process. This shows how z can influence the value of $\mathbf{F}_u$.

differences with the presented method is pointed out. Later an example of studies that show how local features can effect the overall inference is discussed.

Part-based models have been widely used in object detection applications. A good example of such application can be found in the work of Felzenszwalb *et al.*[3]. These models consist of a fixed root feature and several part feature with their position as latent variables in relation to the root feature. The part features are learnt either using exact annotation [5] or as the result of an optimization problem [3]. Because of this latency the model can have many configurations an usually the best configuration is chosen among many due to the task in hand. In these models the part features are used to estimate a better confidence for the root feature.

Taking the part-based models to the feature level comes with several difficulties. To begin with there is larger variation at the feature level compared to the object level. Here each local feature can play the role of a root feature and completely different features can be equally good representatives for an object class. As an example consider the features obtained from the wheel and the door of a car, one wishes for a car model to return a high likelihood for both features despite their differences. Also there is no right or wrong way to look at local features. In this work the root features and their parts are calculated as the result of a clustering process. Since there is no best configuration for these features each root feature can have several good part configurations. These configurations will capture the local variation around the local feature and will later be used for training non-linear discriminative classifiers.

As mentioned in section 1 many methods benefit from discriminative behaviours of local features. A good example of such benefit can be seen in [6] where the authors Sanjiv Kumar *et al.* show how replacing generative models with discriminative models benefits the MRF solvers and improves their final results. Similar examples can be widely found in numerous computer vision studies. The key difference between these works and this method is the fact the features can change their value to result in a more discriminative behaviour.

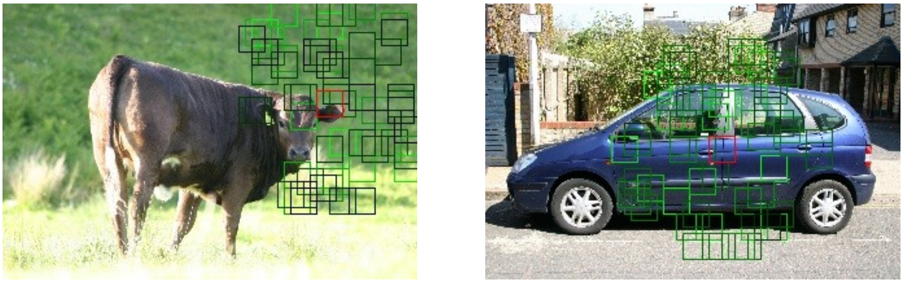


Fig. 2. This figure shows how a car representative anchor point pulls out features from two different images. It can be seen that the features are avoiding cow texture since this texture is not a good candidate for supporting being a car hypothesis.

3 Feature Clouds

To define the feature clouds, let $G(V, E)$ be a graph with the extracted features as its nodes and the relation between them encoded as its edges. Also for each node, say u, let ϕ_u denote the feature vector associated with this node and $\mathbf{N}_u$ denote the its neighbors. A cloud feature, with its root at node u and m latent parts, is the set of all possible vectors that are created by concatenating the feature vector of node u and the feature vectors of m nodes selected from $\mathbf{N}_u$. This set is formally defined as

$$\mathbf{F}_u = \{(\phi_u, \phi_{v_1}, ..., \phi_{v_m}) | v_i \in \mathbf{N}_u\}, \tag{1}$$

where $(\phi_u, \phi_{v_1}, ..., \phi_{v_m})$ is the concatenation of the feature vector of the nodes $u, v_1, ...,$ v_m. In other words all possible configurations that can be made using u and its parts exist in the set $\mathbf{F}_u$. This can also be seen as the space of all variations around node u. These configurations are shown in figure 1 (Left). In this figure a node u (a patch extracted from an image) is connected to its neighbours (its close by patches). In this configuration the three selected neighbors v_2, v_5, v_7 are shown using solid line edges. and the resulting feature vector for this configuration is $(\phi_u, \phi_{v_2}, \phi_{v_5}, \phi_{v_7})$. Here the set $\mathbf{F}_u$ will contain all possible similar feature vectors made by selecting three nodes among the eight neighbors.

In practice only one of the vectors within $\mathbf{F}_u$ is selected and used as its quantitative value. Since the size of this set can grow large this value is selected in an optimization process. This value is determined in relation to a fixed target point in the feature space. For any arbitrary point z in the feature space, the value of $\mathbf{F}_u$ is fixed as the best fitting vector in $\mathbf{F}_u$ to this point. This can be written as

$$f^{\star}_{(z,u)} = \arg\min_{f \in \mathbf{F}_u} \{d(f, z)\}. \tag{2}$$

Here $d(.)$ is the euclidean distance. Here the point z is used as an anchor point in the feature space for fixing latent variables of the feature cloud. Figure 1 (Right) illustrates how $f^{\star}_z$ is selected. In this work z plays an important role in the classification process. Since for every given z the value of $\mathbf{F}_u$ changes, z can be seen as a tool for pulling out

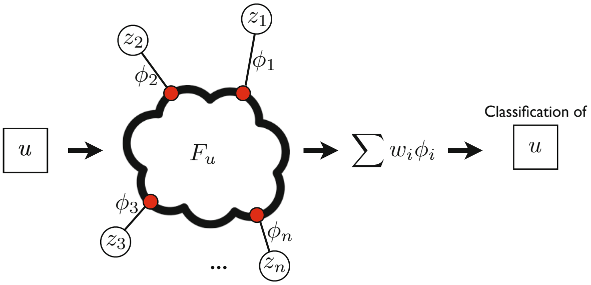


Fig. 3. This figure shows a summery of the classification process. Initially a node u is selected and its corresponding cloud is calculated. With the known anchor points $z_1, \ldots, z_n$, the value for the basis functions ϕ_i are calculated and using them a classification for the node u is achieved.

different properties of $\mathbf{F}_u$. In the classification stage each $\mathbf{F}_u$ will be fit to different z values, learnt during the training process, to verify whether $\mathbf{F}_u$ contains configurations that belong to the object or not. The result of this selection can be seen in figure 2. In this figure a car related region is being extracted from a car image and a cow image. It can be seen that the positioning of the features completely differ in the two images and more importantly the features seem to be avoiding the texture of the cow. This is done because the optimization process tends to find the best available matches in the image to support a certain hypothesis (being on a car) and regions on the cow don't seem to be good candidates.

By having a prior knowledge of how the nodes are distributed in $\mathbf{N}_u$ a partitioning can be imposed on this set. This partitioning will later be referred as the *architecture* of the cloud feature. This partitioning is designed in a way that each latent part comes from one partition. This partitioning can slightly reduce the complexity of the optimization.

Efficient solving of the optimization problem 2 can have a large effect on the performance and the running time of the methods using feaure. Assuming that the size of the extracted features is fixed and distance is measured using euclidean distance, the complexity of implemented method is calculated as $O(|\mathbf{N}_u|m)$, where is m is the number of latent parts. By introducing an architecture and partitioning $\mathbf{N}_u$, this complexity will be reduced to $O(|\mathbf{N}_u|)$. In other words this problem can be solved by visiting the neighbors of each node at most m times. It is possible to design more efficient algorithms for solving this optimization problem and this will be in the focus of the future works of this paper.

4 Latent Classifiers

In this problem the task of a classifier is to take a feature cloud and classify it into either being from the object or not. For a set of labeled features, $\{(\mathbf{F}_{u_1}, y_1), \ldots, (\mathbf{F}_{u_N}, y_N)\}$ gathered from the training set, the goal is to design a function c that uses one or several configurations within cloud features to minimize the cost function

$$\sum_{i=1}^{N} |c(\mathbf{F}_{u_i}) - y_i|. \tag{3}$$

A key difference between this approach and other available approaches is the fact that the local variations around the local feature are also modeled with optimization of c. This means that not only the model uses the value of the root feature but it also uses the dominant features appearing around the root feature regardless of their spatial position.

The optimization problem 3 is approximated by defining the function c as a linear basis regression function with model parameters $z_1, z_2, ..., z_M, W$. The values z_1, z_2, ..., z_M are M anchor points in the feature space for fixing the latent variables capturing different configurations of the features and $W = (w_0, ..., w_M)$ contains the regression weights, these parameters are learnt during the training stage. Using these parameters, the regression function c is defined as

$$c(\mathbf{F}_u; z_1, \ldots, z_M, W) = \sum_{m=1}^{M} w_m \Phi_{z_m}(\mathbf{F}_u) + w_0. \tag{4}$$

Here the basis function Φ_{z_m} measures how good $\mathbf{F}_u$ can be fit to z_m. This basis function can be written as any basis function for example a Gaussian basis function is defined as

$$\Phi_{z_m}(\mathbf{F}_u; s) = \exp(-\frac{d(f^{\star}_{(z_m,u)}, z_m)^2}{2s^2}). \tag{5}$$

Equations 4 and 5 clearly show that the decision made for $\mathbf{F}_u$ depend both on the different values in $\mathbf{F}_u$ and how good it can be fit to z_m values. Due to the dynamic section, $\mathbf{F}_u$ can be fit to different z_m values which makes the scoring processes harder for negative samples. The summery of this classification process can be seen in figure 3. During the training stage solving equation 4 to obtain W is straight forward once the z_m values are known.

The local features come from different regions of an object and these regions are no visually similar. This fact results in a large variation on the data used for training. This variation can not be captured by only one set of M configurations. To solve this problem a mixture model of K sets each with M different configurations is considered and each set is associated with a different regression model. When classifying a cloud feature, it is initially assigned to the model that minimizes the over all fitting cost, defined as

$$\arg\min_{k \in \{1 \ldots K\}} \{\sum_{m=1}^{M} d(f^{\star}_{(z_m^{(k)},u)}, z_m^{(k)})\}. \tag{6}$$

Here the model is chosen based on how good the feature $\mathbf{F}_u$ is fit to all the configurations within the model.

5 Learning the Parameters

The goal of the learning algorithms is to determine the K configurations sets together with the regression function. It is possible to design an optimization method to estimate all configurations together with the regression function, but such optimization is more suitable for an object level classification since the data contains less variation at that level. In this work the optimization is done two separate steps. The first step uses a

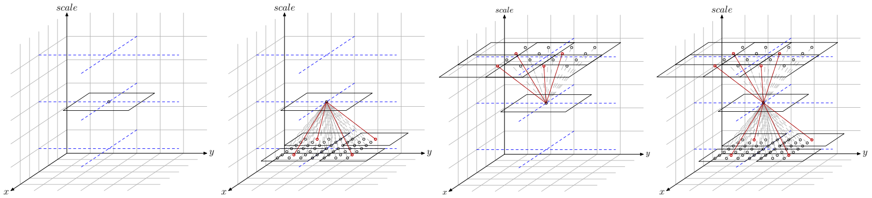


Fig. 4. (Left) The **Arc1** only depends on the patch itself. (Middle) The **Arc2** (**Arc3**) Depends on the central patch together with a selection of patches in the scale details (context) of the central patch. (Right) The **Arc4** Depends on the central patch and a selection of patches from from details and the context of the central patch.

generative method for finding the K sets of configurations. This generative method is an adaptation k-means algorithm with the cost function

$$\arg\min_{S} \sum_{k=1}^{K} \sum_{\mathbf{F}_u \in S_k} \left(\sum_{m=1}^{M} d(f^{\star}_{z_m^{(k)}}, z_m^{(k)}) \right). \tag{7}$$

This adaptation of k-means divides the features into K clusters and the elements of each cluster have a strong connection by sharing the M different configurations. This optimization can be solved using iterative methods used for solving the k-means problem. This method will be referred as *L-KMEANS*(K,M). To optimize this cost function initially the feature clouds are partitioned into K subsets based on how close they are to the M anchor points. After this partitioning the obtained values from the clouds of each partition are used for updating the anchor points. This process is continued until a convergence is achieved.

To determine the parameters initially *L-KMEANS*(K,M) procedure is ran over the positive features. This way the strong configurations appearing in the training set are formulated in terms of cluster centers resulted by the procedure. These cluster centers can be used for labeling all features. For each cluster the variations in the negative features is captured by running *L-KMEANS*(1,M) on the negative features assigned to that cluster. Finally, after identifying both positive and negative configurations, these configurations are used to fix the values of cloud features and the and regression function 4 is optimized to separate the positive features from the negative features.

6 Experiments and Results

The aim of the proposed methodology is to define a level of latency at the feature level to extract more discriminative local/semi-global features. This latency in selection of features can benefit many different pattern recognition tasks. To analyze the effect of this latency two different scenarios are considered. A scenario is to employ the feature clouds in a bag-of-visual-words (BOVW) model. The idea behind the bag of visual words model is the fact that objects are built using a series of local structures that are shared between different objects and a histogram of such local structures can lead to the identification of the objects. Meanwhile, another challenge is to identify unique

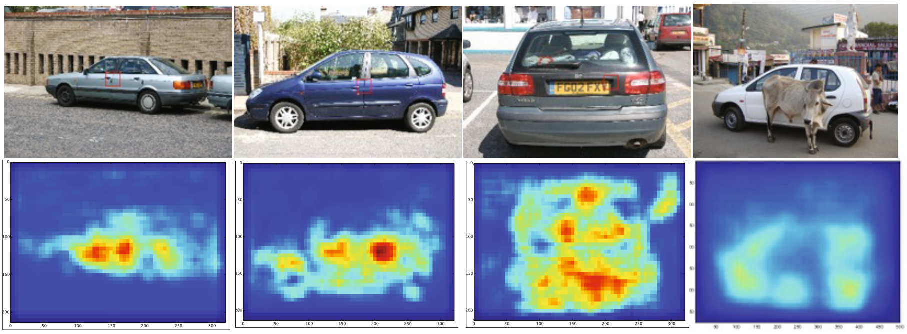


Fig. 5. The heat-maps shown in this figure are produced by mapping the scores of all the individual features on a plane. These show how the models locally learn the structures of the object.

structures (parts) on the objects and use them for the classification of the object. The fact that the parts are defined in a more global sense and are more related to the object class, makes them bad candidates for BOVW models. Instead the quality of these parts can be evaluated based on how *good* they appear on the object. Both these scenarios are discussed in this section.

When defining the feature clouds in section 3, each cloud is defined based on a set of neighboring features. On an image the neighboring features can be located as features withing a spatial radius of δ around the root feature. In this context δ will be referred as the flexibility parameter. This parameter can be used to build clouds for different purposes. If the value of δ is low then the joint representation become more local therefor more suited for BOVW models. A disadvantage of building such local joint presentations is the fact that their performance of becomes close to the performance of single features upon which the joint features were built. When the value of δ grows large, the joint representations can be selected from a larger range of features on the object, which makes the selection more dependent on object class and the features become more class and viewpoint dependent. Having a semi-global feature provides a series of features that are independent and are rarely detected off the object.

The experiments conducted in this paper are not designed to be compared with state of the art object detectors but to test the hypothesis proposed in the paper. The main idea behind the experiments is to evaluate these local features with different architectures and compare them with the baseline which only contains the root feature. The evaluation is straight forward, each feature is scored using the equations 4 and 6. Figure 5 shows how the heat-map of this score looks like for different images. For these figures the score was calculated for individual nodes and mapped on a plane. In the feature inference problem the results are presented in terms of different precision recall curves of these values. The experiments are conducted on several classes of *MSRC v1* and *v2* [17] datasets.

6.1 Feature Clouds as Semi-global Features

Several parameters control the behaviour of the cloud features. As mentioned in section 3 the architecture of the cloud features imposes a strong prior on how the latent parts are placed together. The architecture controls the complexity of the feature by controlling the number of its latent parts. The flexibility of these features is controlled by the number of neighbors each node has in graph G. The larger the size of the neighbours the wider the search space is for the latent parts. The goal of the experiments to analyze how the complexity and flexibility of the cloud features effects their discriminative behavior. Unfortunately, as the flexibility and the complexity of the features increase the optimization processes in equations 7 and 4 become computationally expensive. Therefore the results are only provided on a few classes of this dataset.

The graph used in the experiments is build over the fixed size patches with 30 pixel side extracted from an image pyramid and described using a PHOG descriptor [1]. The choice of this graph is due to the future works of this paper when these features are used to build object level classifiers.

Let $G = (V, E)$ denote the graph built over the patches extracted from the image pyramid with V containing all extracted patches and for two patches in V, say u and v, the edge uv belongs to E iff $|x_u - x_v| < t$ and $|s_u - s_v| < 1$. Here x_u is the spatial position, s_u is the scale level of node u and t is a given threshold which controls how patches are connected to each other.

In this work four different architectures are considered. These architectures are considered as prior information and are hard coded in the method. Although it is possible to learn the architectures, learning them requires more tools which are not in the scope of this paper. As mentioned in section 3 the architecture is imposed by partitioning the set $\mathbf{N}_u$. In this problem the neighbors of each node come from three different scale levels. There are, of course, many different ways that this set can be partitioned into m subsets. To reduce the number of possibilities only partitions with simple fixed scale and spatial relations to the central node are considered. Twelve subsets are formed by dividing the nodes (patches) in each scale into four quadrants. The scale levels and quadrants can be seen in the features shown in figure 4. A number of these subsets are selected to form different features architectures. Let this selection be denoted by $\overline{\mathbf{P}}_u$. Using a subset of the twelve partitions, the four architectures defined in this figure are,

- **Arc 1:** This architecture is created by having $\overline{\mathbf{P}}_u = \emptyset$. This architecture uses the descriptor of the central node as the descriptor. This feature will be used as a benchmark for analyzing dynamic architectures.
- **Arc 2:** Let $\overline{\mathbf{P}}_u$ partition the scale level below the scale level of u into four spatial quadrants. This architecture contains the data from node u and additional information about the details in this region.
- **Arc 3:** Let $\overline{\mathbf{P}}_u$ partition the scale level above the scale level of u into four spatial quadrants. This architecture contains the data from node u and additional information about the context in this region.
- **Arc 4:** Let $\overline{\mathbf{P}}_u$ partition the scale levels both above and below the scale level of u. This architecture contains the data from u together with information about details and the context of the region u has appeared in.

Here each node can be described using each of the four architectures and the goal is to verify the most suitable architecture for the object region. To train the classifiers any feature from the positive regions is considered positive and the rest of the extracted features are considered as negative features. This should be kept in mind that the problem

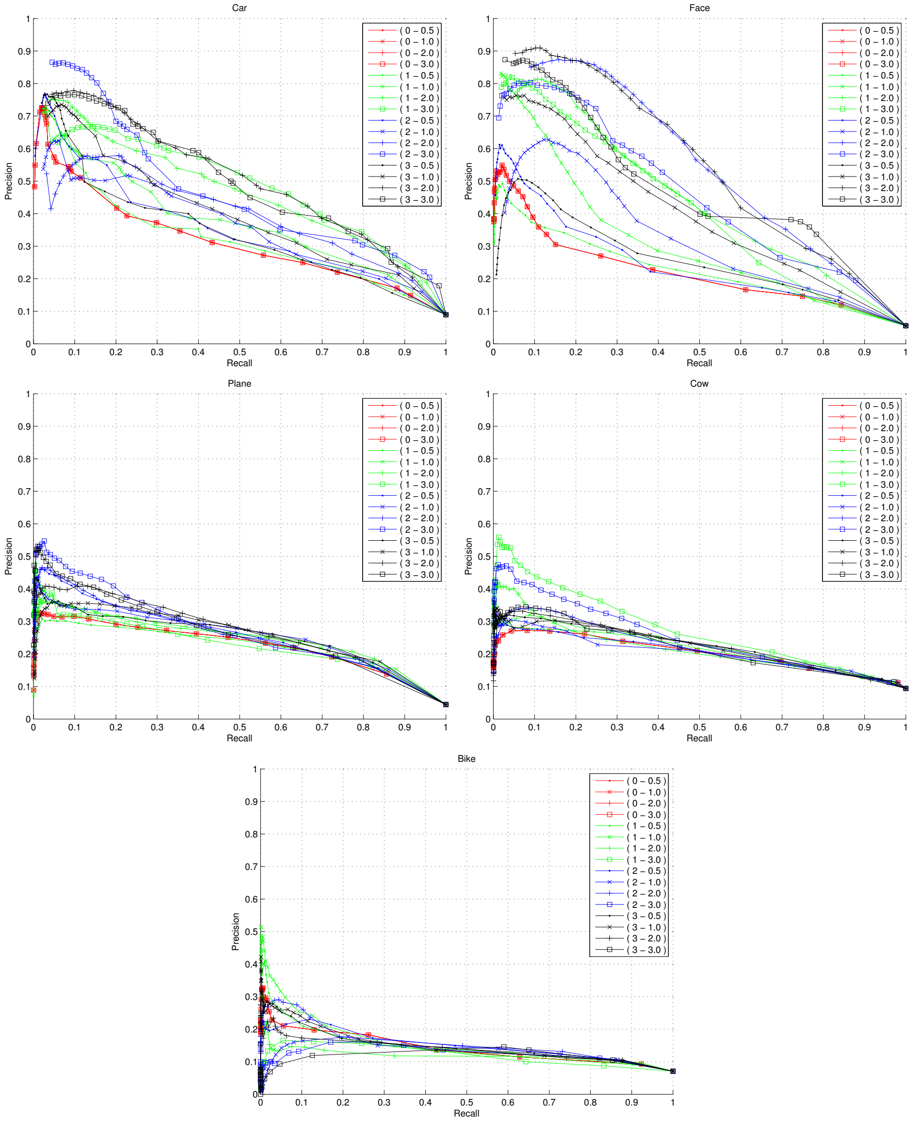


Fig. 6. Results from the *MSRCv1* dataset. Five classes $\{car, face, plane, cow, bike\}$ were considered from this dataset. In this experiment the value of t controlling number of neighbors was varied from the value equal to half the patch size to three times larger than the patch size.

being solved here is equivalent taking an arbitrary patch from an arbitrary scale and location and asking whether it belongs to the object or not. Due to the noise at the feature level this problem is a hard problem to solve by nature.

Classes $\{Car, Face, Plane, Cow, Bike\}$ are chosen from this dataset. In this experiment all four architectures are used with varying t value to increase the flexibility of features.

Here 256 models are used and each with 5 positive and 5 negative latent configurations. The results of this experiment can be seen in figure 6. These experiments reveal several properties about the cloud features. The first property is in fact that the improvement obtained on the feature likelihood level depend on both the base feature and the architecture. At it can be seen in this figure the base feature (red curves) has an easier time capturing the properties of the *car* and *face* classes in comparison with the rest of the classes. Also between these two classes the architectures have an easier time capturing the relations in face region. Meanwhile it is clearly visible that for the *bike* class both the features and architectures are failing to capture the local properties. This figure also shows that there is no best architecture for the all the object classes and the choice of the architecture is completely object dependent. This can be seen in the likelihoods obtained from the *plane* and the *cow* classes, where the base likelihoods are similar but the responses obtained from the different architectures are different.

6.2 Feature Clouds in BOVW Models

This experiment is conducted on 9 classes of *MSRC v2* dataset following the experiment setting presented by Morioka *et al.* [11] for building local pairwise codebook (LPC). In their setting sift features were sampled at every 8 pixels from the images and LPC was build by clustering them. In their framework features with distance equal or less than 8 pixels are merged to build joint features. To adapt this scenario the graph $G(V, E)$ from section 3 is constructed over such sift features. Here V contains all the sampled sift features and for every two features, say u and v, $uv \in E$ iff $|x_u - x_v| \leq \delta = 8$. In the concept of feature clouds each the anchor points is optimized for each for each class. Here a number of anchor points are calculated for the classes (together with the background class) using the cost function 7 and put together as N anchor points $\{c_1, ..., c_N\}$. In this experiment, equation 2 was used to build a histogram for each image. To define this formally let $\{\mathbf{F}_{u_1}, \ldots, \mathbf{F}_{u_M}\}$ be M feature clouds extracted from an image and H be a N bin histogram with $H[n]$ representing its n^{th} bin. The value of $H[n]$ is determined as

$$H[n] = \#\{\mathbf{F}_{u_i} : \forall m \neq n, d(f^\star_{(c_n,u_i)}, c_n) < d(f^\star_{(c_m,u_i)}, c_m)\}. \tag{8}$$

Here $\mathbf{F}_{u_i}$ is assigned to most fitting anchor point. Similar to [11] the histograms where classified using non-linear SVM with histogram intersection kernel. In this experiment beside the appearance of latent parts their position was also modeled. This modeling was by changing the optimization 2 to

$$f^\star_z = \operatorname*{arg\,min}_{f=(f_A,f_x)\in\mathbf{F}_u} \{\alpha d(f_A, z_A) + (1-\alpha) d(f_x, z_x)\}. \tag{9}$$

Here f_A and z_A contain the appearance information of the configuration and the anchor point, while f_x and z_x contain the information about the relative position of the latent parts with respect to the root feature. The value alpha was considered as a constant value equal to 0.75.

For this experiment half of the images in each class were used as training images and the other half was used for testing. The SIFT features were calculated on a dense grid after smoothing the image using VLFeat's [15] Matlab API. In this experiment feature clouds with accuracy of $84.19 \pm 2.75\%$, having 820 words in the dictionary, have performed better than the best baseline accuracy (single features) with accuracy of $83.0 \pm 2.0\%$. Also since the leading dimensions of the words, calculated for the feature clouds, corresponding to the root features can be used for labeling root features, a secondary histogram can be calculated for these features. The performance of the concatenated histogram of cloud features and root features classified the images in the dataset with the accuracy of $85.6 \pm 1.4\%$. Our results are higher than the results reported for LPC[11] $83.9 \pm 2.9\%$ and [18] $78.3 \pm 2.6\%$ for 2^{nd} order and $80.4 \pm 2.5\%$ for 10^{th} order spatial features.

7 Conclusions and Future Works

The main objective of this work has been to investigate the improvement in discriminability obtained by substituting simple local features with more adaptive composite hierarchical structures that are computed at recognition time from a set of potential structures denoted as feature clouds. This is motivated by the fact that even at local feature level, intra class object variation is very large, implying that generic single feature classifiers that try to capture this variation will be very difficult to design. In our approach this difficulty is circumvented by the introduction of the cloud features that capture the intra class variation an feature level. The price paid is of course a more complex process for the extraction of local features that are computed in an optimization process in order to yield maximally efficient features. We believe however that this process can be made efficient by considering the dependencies and similarities between local feature variations that are induced by the global intra class object variation.

There are many ways to improve the performance and accuracy of the feature clouds and investigate their applications. As mentioned in the text, coming up with better optimization algorithms will decrease the usage cost of these features. Meanwhile designing algorithms for learning the architecture rather than hard-coding them will increase the accuracy of these features. As for the applications, these features can be used in different object detection and recognition platforms. A direct follow up of this work is using these features to build more robust object detectors for detecting object classes. Since the cloud features are results of clustering process rather than discriminative analysis, they can also be used in bag-of-words models and will result in more discriminative words and smoothed labeled regions.

Acknowledgements. This work was supported by The Swedish Foundation for Strategic Research in the project Wearable Visual Information Systems.

References

1. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: CIVR, pp. 401–408. Association for Computing Machinery (July 2007)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, pp. 886–893 (2005)

3. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. Technical report (2009)
4. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1627–1645 (2010)
5. Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. IJCV 61, 55–79 (2005)
6. Kumar, S., Hebert, M.: Discriminative random fields. IJCV 68, 179–201 (2006)
7. Laptev, I.: Improvements of object detection using boosted histograms. In: BMVC, pp. 949–958 (2006)
8. Ling, H., Soatto, S.: Proximity distribution kernels for geometric context in category recognition. In: ICCV (2007)
9. Liu, D., Hua, G., Viola, P., Chen, T.: Integrated feature selection and higher-order spatial feature extraction for object categorization. In: CVPR, pp. 1–8 (2008)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints (2003)
11. Morioka, N., Satoh, S.: Building Compact Local Pairwise Codebook with Joint Feature Space Clustering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 692–705. Springer, Heidelberg (2010)
12. Savarese, S., Winn, J., Criminisi, A.: Discriminative object class models of appearance and shape by correlatons. In: CVPR, vol. 2, pp. 2033–2040 (2006)
13. Varma, M., Zisserman, A.: Texture classification: Are filter banks necessary. In: CVPR (2003)
14. Varma, M., Zisserman, A.: Classifying Images of Materials: Achieving Viewpoint and Illumination Independence. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part III. LNCS, vol. 2352, pp. 255–271. Springer, Heidelberg (2002)
15. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), http://www.vlfeat.org/
16. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: CVPR, vol. 1, p. 511 (2001)
17. Winn, J., Criminisi, A., Minka, T.: Object categorization by learned universal visual dictionary. In: ICCV 2005, pp. 1800–1807. IEEE Computer Society, Washington, DC (2005)
18. Zhang, Y., Chen, T.: Efficient kernels for identifying unbounded-order spatial features. In: IEEE Computer Society Conference on CVPR, pp. 1762–1769 (2009)

SVM-Based Feature Selection and Classification for Email Filtering

Sebastían Maldonado[1] and Gaston L'Huillier[2]

[1] Universidad de los Andes, School of Engineering and Applied Sciences,
Av. San Carlos de Apoquindo 2200, Las Condes, Santiago, Chile
[2] Groupon, Inc., 3101 Park Blvd., Palo Alto, CA. 94301, U.S.A.
smaldonado@uandes.cl, gaston@groupon.com

Abstract. The email inbox is indeed a dangerous place, but using pattern recognition tools it is possible to filter most wasteful elements that may cause damage to end users. Furthermore, as phishing and spam strategies have shown an adversarial and dynamic behavior, the number of variables to be considered for a proper email classification has increased substantially over time. For many years these elements have driven pattern recognition and machine learning communities to keep improving email filtering techniques. This work presents an embedded feature selection approach that determines a non-linear decision boundary with minimal error and a reduced number of features by penalizing their use in the dual formulation of binary Support Vector Machines (SVMs). The proposed method optimizes the width of an anisotropic RBF Kernel via successive gradient descent steps, eliminating those features that have low relevance for the model. Experiments with two real-world spam and phishing data sets demonstrate that our approach has a better performance than well-known feature selection algorithms while consistently using a smaller number of variables.

Keywords: Spam and phishing filtering, Support vector machines, Feature selection, Embedded methods.

1 Introduction

One particular domain for which machine learning has been considered a key component is cyber-security. Specifically, for the correct identification of the large number of spam messages, web spam, and spam servers which inundate Internet resources every day. It is likely that spam messages will continue to be one of the most wasteful, dangerous and infectious elements on the Web as new campaigns are occasionally instigated by spam senders [29].

One of the main reasons for which spam classification is relevant, is that unsolicited email leads to several threats, as they are not just unwanted product advertisements. Through spam mechanisms, different types of viruses, Trojans, and links to phishing fraud websites are spread on mass. Among the reasons for which spam classification is important, its carbon footprint has gain relevance over the last years. It has been reported that the carbon footprint of the 62 trillion spam emails sent each year is equivalent to 3.1 million cars on the road using at least 2 billion gallons of gasoline [19].

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 135–148.
DOI: 10.1007/978-3-642-36530-0_11

In the *cyber-crime* context, one of the most common social engineering threats is phishing fraud. This malicious activity consists of sending email scams, asking for personal information to break into any virtual or physical location where victims may store useful private information, such as financial institutions, *e*-Commerce, among other locations. Using phishing, millions of dollars are stolen every year[1], and this number is likely to keep raising as the Internet penetration in our everyday life increases.

Identifying malicious emails such as spam or phishing can be considered as a task of binary classification where the goal is to discriminate between the two classes of "desired" and "undesired" emails. Support Vector Machine [31] is an effective classification method and provides several advantages such as absence of local minima, adequate generalization to new objects, and representation that depends on few parameters. Furthermore, this method has proved to be very effective for spam classification [27] and Phishing [15]. However, this approach does not determine the importance of the features used by a classifier [17]. In this paper we present a feature selection approach for binary classification using SVM, showing its potential for spam and phishing classification.

This paper is organized as follows. In Section 2 we briefly introduce spam and phishing classification. Recent developments for feature selection using SVM are reviewed in Section 3. Section 4 presents the proposed feature selection method based on SVM. Experimental results using real-world data sets for spam and phishing classification are given in Section 5. A summary of this paper can be found in Section 6, where we provide its main conclusions and address future developments.

2 Spam and Phishing Classification

Among all counter-measures used against spam and phishing, there are two main alternatives [4]: content-based classification methods and network-based strategies. In the following, the main approaches for these alternatives are briefly reviewed.

2.1 Content-Based Classification

Spam filtering is a classical problem in machine learning, and many filtering techniques have been described [12]. However, in terms of content-based classification, phishing differs in many aspects from the spam case. While most of spam emails are intended to spread information about products and web sites, in phishing, the interaction between a message and the receiver is more complex. End users are usually involved in a third step of interaction, such as following malicious links, filling deceptive forms, or replying with useful information which are relevant for the fraud message to succeed.

Also, there is a clear difference among many phishing techniques, classified into two main categories, known as *deceptive phishing* and *malware phishing* [4]. While *malware phishing* has been used to spread malicious software installed on victim's machines, *deceptive phishing*, according to [4], can be categorized in six categories: *Social*

[1] Reports and statistics are kept by the Antiphishing Working Group, `www.apwg.org` [Online: accessed May 13, 2012].

engineering, Mimicry, Email spoofing, URL hiding, Invisible content, and *Image content*. For each one of these subcategories, content-based features have been proposed by [4] to enhance phishing classifiers.

Previous works on content-based filtering of deceptive spam or phishing emails have focused on the extraction of a large number of features used in popular machine learning techniques for its classification [5,4]. In [1], logistic regression, SVMs, and random forests were used to construct classifiers to correctly label email messages, obtaining the best results with an F-measure of 0.9. In [11], using a list of improved features extracted directly from email messages, the author proposed an SVM-based model which obtained an F-measure of 0.9764 in a different phishing and spam corpus data set.

2.2 Network-Based Classification

Real Time Blacklists (RBLs) have been considered as an efficient alternative to filtering spam messages, just by considering server-side features for spam sender detection. These services can be queried over the Domain Name System (DNS) protocol, which provides a powerful tool for email servers to decide whether or not to accept messages from a given host [26]. Furthermore, different machine learning classifiers have been proposed to classify spam senders, such as the basic formulation of SVMs [27], a modified extension for SVMs especially designed for imbalanced datasets, called Granular SVM with Boundary Alignment (GSVM-BA) [28], and an improvement of previous method, called Granular SVM with Random granulation (GSVM-RAND) [25].

To date, few experiments have been documented in terms of large scale spam server classification in different contexts. One approach, introduced by [16], describes how to use online learning algorithms to classify suspicious URLs, which could be related to phishing fraud and spam activities. Furthermore, [30] designed and evaluated a real-time malicious URLs classification strategy using a distributed approach for the logistic regression binary classification algorithm.

The latter approaches are based on features extracted from network properties and not from content-based characteristics, hence the dimensionality of the classification problem is considerably low and the features' properties are different than in content-based approaches. For this reason, these approaches were not considered in this paper.

3 Feature Selection for SVMs

In this section we recall the classification method Support Vector Machine (SVM) developed by [31]. Additionally, we present the main strategies for feature selection with SVMs.

3.1 Support Vector Classification

Given training points $\mathbf{x}_i \in \mathbb{R}^n$, $i \in \{1,\dots,m\}$ and binary labels $\mathbf{y} \in \mathbb{R}^m$, $y_i \in \{-1,+1\}$, SVM provides the optimal hyperplane $f(\mathbf{x}) = \mathbf{w}^T \cdot \mathbf{x} + b$ that aims to separate the training examples by maximizing the *margin*, which is equivalent to minimizing the norm of coefficients $\mathbf{w}$ [31]. A set of slack variables ξ is also introduced for each training vector, considering a penalty parameter C, which helps to control the degree of misclassification.

For a non-linear classifier, the solution will be given in a form of a Kernel machine, where training data are mapped to the higher dimensional space $\mathscr{H}$ by the function $\mathbf{x} \to \phi(\mathbf{x}) \in \mathscr{H}$. The mapping is performed by a kernel function $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{y})$ which defines an inner product in $\mathscr{H}$ [24].

The optimal hyperplane is thus the one with maximal distance (in $\mathscr{H}$) to the closest image $\phi(\mathbf{x}_i)$ from the training data. The dual formulation of SVM for binary classification can be stated as follows:

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s) \tag{1}$$

subject to

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

$$0 \le \alpha_i \le C \qquad i \in \{1, \ldots, m\}.$$

From a variety of available kernel functions, the linear, polynomial, and the Gaussian kernel are chosen in many applications:

1. Linear Kernel: $K(\mathbf{x}_i, \mathbf{x}_s) = \mathbf{x}_i \cdot \mathbf{x}_s$.
2. Polynomial Kernel: $K(\mathbf{x}_i, \mathbf{x}_s) = (\mathbf{x}_i \cdot \mathbf{x}_s + 1)^d$, where $d \in \mathbb{N}$ is the degree of the polynomial.
3. Gaussian Kernel: $K(\mathbf{x}_i, \mathbf{x}_s) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_s||^2}{2\rho^2}\right)$, where $\rho > 0$ is the parameter controlling the width of the kernel.

The selection of the best kernel function is still a matter of research [2,24]. Empirically, best classification performance is usually achieved with the Gaussian Kernel [2].

3.2 Feature Selection with SVMs

There are different strategies for embedded feature selection. First, feature selection can be seen as an optimization problem. For example, the methods presented in [21] add an extra term that penalizes the cardinality of the selected feature subset to the standard cost function of SVM. By optimizing this modified cost function features are selected simultaneously to model construction. Another embedded approach is the Feature Selection ConcaVe (FSV) [7], based on the minimization of the "zero norm" : $\|\mathbf{w}\|_0 = |\{i : w_i \neq 0\}|$. Note that $\|\cdot\|_0$ is not a norm because the triangle inequality does not hold [7], unlike l_p-norms with $p > 0$. Since the l_0-"norm" is non-smooth, it was approximated by a concave function:

$$\|\mathbf{w}\|_0 \approx \mathbf{e}^T(\mathbf{e} - \exp(-\beta|\mathbf{w}|) \tag{2}$$

with an approximation parameter $\beta \in \mathbb{R}_+$ and $\mathbf{e} = (\mathbf{1}, \ldots, \mathbf{1})^{\mathbf{T}}$. The problem is finally solved by using an iterative method called Successive Linearization Algorithm (SLA) for FSV [7]. [33] proposed an alternative approach for zero-"norm" minimization (l_0-SVM) by iteratively scaling the variables, multiplying them by the absolute value of the weight vector $\mathbf{w}$. [22] consider simultaneously the three objectives goodness-of-fit, a regularization parameter for structural risk minimization, and feature penalization, considering a sequential forward selection strategy. An important drawback of these methods is that they are limited to linear classification functions [13].

Several embedded approaches consider backward feature elimination in order to establish a ranking of features, using SVM-based contribution measures to evaluate their relevance. One popular method is known as Recursive Feature Elimination (SVM-RFE) [14]. The goal of this approach is to find a subset of size r among n variables ($r < n$) which maximizes the classifier's performance. The feature to be removed in each iteration is the one whose removal minimizes the variation of $W^2(\boldsymbol{\alpha})$:

$$W^2(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i,\mathbf{x}_s) \tag{3}$$

The scalar $W^2(\boldsymbol{\alpha})$ is a measure of the model's predictive ability and is inversely proportional to the margin. Features are eliminated applying the following procedure:

1. Given a solution $\boldsymbol{\alpha}$, for each feature p calculate:

$$W^2_{(-p)}(\boldsymbol{\alpha}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i^{(-p)},\mathbf{x}_s^{(-p)}) \tag{4}$$

   where $\mathbf{x}_i^{(-p)}$ represents the training object i with feature p removed.

2. Eliminate the feature with smallest value of $|W^2(\boldsymbol{\alpha}) - W^2_{(-p)}(\boldsymbol{\alpha})|$.

Another ranking method that allows kernel functions was proposed in [23], which considers a *leave-one-out* error bound for SVM, the *radius margin bound* [31] $LOO \leq 4R^2||\mathbf{w}||^2$, where R denotes the radius of the smallest sphere that contains the training data. This bound is also used in [34] through the *scaling factors* strategy. Feature selection is performed by scaling the input parameters by a vector $\boldsymbol{\sigma} \in [0,1]^n$. Large values of σ_j indicate more useful features. The problem consists in choosing the best kernel of the form:

$$K_{\boldsymbol{\sigma}}(\mathbf{x}_i, \mathbf{x}_s) \equiv K(\boldsymbol{\sigma} * \mathbf{x}_i, \boldsymbol{\sigma} * \mathbf{x}_s) \tag{5}$$

where $*$ is the component-wise multiplication operator. The method presented by [34] considers the gradient descent algorithm for updating $\boldsymbol{\sigma}$. [8] propose to limit the use of the attributes by constraining the scaling factors using a parameter σ_0, which controls the norm of $\boldsymbol{\sigma}$.

4 The Proposed Method for Embedded Feature Selection

An embedded method for feature selection using SVMs is proposed in this section. The reasoning behind this approach is that we can improve classification performance by eliminating the features that affect on the generalization of the classifier by optimizing the Kernel function. The main idea is to penalize the use of features in the dual formulation of SVMs using a gradient descent approximation for Kernel optimization and feature elimination. The proposed method attempts to find the best suitable RBF-type Kernel function for each problem with a minimal dimension by combining the parameters of generalization (using the 2-norm), goodness of fit, and feature selection (using a 0-"norm" approximation).

For this approach we use the anisotropic Gaussian Kernel:

$$K(\mathbf{x}_i, \mathbf{x}_s, \boldsymbol{\sigma}) = exp\left(-\frac{||\boldsymbol{\sigma} * \mathbf{x}_i - \boldsymbol{\sigma} * \mathbf{x}_s||^2}{2}\right) \tag{6}$$

where $*$ denotes the component-wise vector product operator.

The proposed approach (Kernel-Penalized SVM) incorporates feature selection in the dual formulation of SVMs. The formulation includes a penalization function $f(\boldsymbol{\sigma})$ based on the 0-"norm" approximation (2) described in Section 3 and modifying the Gaussian Kernel using an (anisotropic) width vector $\boldsymbol{\sigma}$ as a decision variable. The feature penalization should be negative since the dual SVM is a maximization problem. The following embedded formulation of SVMs for feature selection is proposed:

$$\underset{\boldsymbol{\alpha}, \boldsymbol{\sigma}}{\text{Max}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s, \boldsymbol{\sigma}) - C_2 f(\boldsymbol{\sigma}) \tag{7}$$

subject to

$$\begin{aligned} &\sum_{i=1}^{m} \alpha_i y_i = 0 \\ &0 \le \alpha_i \le C \qquad i \in \{1, \ldots, m\}. \\ &\sigma_j \ge 0 \qquad\quad\;\; j \in \{1, \ldots, n\}. \end{aligned}$$

Notice that the values of $\boldsymbol{\sigma}$ are always considered to be positive, in contrast to the weight vector $\mathbf{w}$ in formulation (2), since it is desirable that the kernel widths be positive values [18]. Considering the "zero norm" approximation described in (2), $\|\boldsymbol{\sigma}\|_0 \approx \mathbf{e}^T(\mathbf{e} - \exp(-\beta|\boldsymbol{\sigma}|)$, and since $|\sigma_j| = \sigma_j \; \forall j$, it is not necessary to use the 1-norm in the approximation.

The following feature penalization function is proposed, where the approximation parameter β is also considered. In [7], the authors suggest setting β to 5:

$$f(\boldsymbol{\sigma}) = \mathbf{e}^T(\mathbf{e} - \exp(-\beta\boldsymbol{\sigma})) = \sum_{j=1}^{n} [1 - exp\,(-\beta\sigma_j)] \tag{8}$$

Since the formulation (7) is non-convex, we develop an iterative algorithm for its approximation. A 2-step methodology is proposed: first we solve the traditional dual formulation of SVM for a fixed anisotropic kernel width $\boldsymbol{\sigma}$:

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s, \boldsymbol{\sigma}) \tag{9}$$

subject to

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

$$0 \leq \alpha_i \leq C \qquad i \in \{1, \ldots, m\}.$$

In the second step the algorithm solves, for a given solution $\boldsymbol{\alpha}$, the following non-linear formulation:

$$\underset{\boldsymbol{\sigma}}{\text{Min}} \quad F(\boldsymbol{\sigma}) = \sum_{i,s=1}^{m} \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s, \boldsymbol{\sigma}) + C_2 f(\boldsymbol{\sigma}) \tag{10}$$

subject to

$$\sigma_j \geq 0 \qquad j \in \{1, \ldots, n\}.$$

The goal of formulation (10) is to find a sparse solution, making zero as many components of $\boldsymbol{\sigma}$ as possible. We propose an iterative algorithm that updates the anisotropic kernel variable $\boldsymbol{\sigma}$, using the gradient of the objective function, and eliminates the features that are close to zero (below a given threshold ϵ). The algorithm solves successive gradient descent steps until one particular scaling factor σ_j drops below a threshold ϵ, starting with one initial solution $\boldsymbol{\sigma}_0$. When this happens, attribute j is eliminated by setting $\sigma_j = 0$. Then the algorithm returns to formulation (9) until convergence. It is also possible that several variables become zero in one iteration. The algorithm Kernel Width Updating and Feature Elimination follows:

Algorithm 1. Kernel Width Updating and Feature Elimination.

```
Start with $\boldsymbol{\sigma} = \boldsymbol{\sigma}_0$;
flag=true; flag2=true;
while(flag==true) do
  train SVM (formulation (9));
  $t = 0$;
   while(flag2==true) do
     $\boldsymbol{\sigma}^{t+1} = \boldsymbol{\sigma}^t - \gamma \Delta F(\boldsymbol{\sigma}^t)$;
     if ($||\boldsymbol{\sigma}^{t+1} - \boldsymbol{\sigma}^t||_1 < \epsilon'$) then
        flag2==false, flag==false;
     else
       if ($\exists j \mid \sigma_j^{t+1} > 0 \wedge \sigma_j^{t+1} < \epsilon, \forall j$) then
         for all ($\sigma_j^{t+1} < \epsilon$) do $\sigma_j^{t+1} = 0$;
         flag2==false;
       end if
     end if
      $t = t + 1$;
     end while;
   end while;
```

In the seventh line the algorithm adjusts the Kernel variables by using the gradient descent procedure, incorporating a gradient parameter γ. In this step the algorithm computes the gradient of the objective function in formulation (10) for a given solution of SVMs $\boldsymbol{\alpha}$, obtained by training an SVM classifier using formulation (9). For a given feature j, the gradient of formulation (10) is:

$$\Delta_j F(\boldsymbol{\nu}) = C_2 \beta exp\left(-\beta\sigma_j\right) + \sum_{i,s=1}^{m} \sigma_j (x_{i,j} - x_{s,j})^2 \alpha_i \alpha_s y_i y_s K(\mathbf{x}_i, \mathbf{x}_s, \boldsymbol{\sigma}) \qquad (11)$$

Lines 11 to 14 of the algorithm represent the feature elimination step. When a Kernel variable σ_j in iteration $t+1$ is below a threshold ϵ, this feature is considered as irrelevant and eliminated by setting $\sigma_j = 0$. This variable will not be included in subsequent iterations of the algorithm.

Lines 8 and 9 of the algorithm represent the stopping criterion, which is reached when $\boldsymbol{\sigma}^{t+1} \approx \boldsymbol{\sigma}^t$. It is also possible to monitor the convergence by considering the measure $||\boldsymbol{\sigma}^{t+1} - \boldsymbol{\sigma}^t||_1$, which represents the variation of the Kernel width between two consecutive iterations t and $t+1$.

5 Results for Spam and Phishing Data Sets

We applied the proposed approach for feature selection to two data sets. We consider the following procedure for model comparison: First, model selection is performed before feature selection, obtaining the kernel parameters d, ρ and penalty parameter C. The best combination is selected via 10-fold cross-validation. For the methods RFE-SVM, FSV-SVM and Fisher Filtering a ranking is first obtained with the training data, and model performance is then obtained using 10-fold cross-validation for specific numbers of attributes, depending on the size of the data set, considering the hyper-parameters obtained during the model selection procedure. For KP-SVM, instead, the algorithm runs using initial hyper-parameters and automatically obtains the desired number of features and the Kernel shape when convergence is reached we compute also the average cross-validation performance in intermediate steps for comparison purposes. The parameters for KP-SVM were selected previously according to the following values:

- Parameter C_2 represents the penalty for the feature usage and is strongly related to C, the original regularization parameter. C_2 is considered the most important parameter for KP-SVM, since classification results change significantly varying its values. We try the values $C_2 = \{0, 0.5C, C, 2C\}$, monitoring both classification accuracy and feature usage.
- The initial (isotropic) kernel width $\boldsymbol{\sigma}_0$, the threshold ϵ and the gradient parameter γ are considered less influential in the final solution, according to our empirical results. We set $\boldsymbol{\sigma}_0 = \frac{1}{\rho^2} \cdot \mathbf{e}$, where ρ is the isotropic kernel width obtained in a previous step for model selection considering all features, and $\mathbf{e}$ is a vector of ones of the size of the number of current features in the solution; $\epsilon = \frac{1}{100\rho^2}$ and $\gamma = 0.1\epsilon||\Delta F(\boldsymbol{\sigma}^0)||$, where $||\Delta F(\boldsymbol{\sigma}^0)||$ represents the Euclidean norm of the first computed gradient vector. This combination of parameters guarantees both a sufficiently small ϵ that avoids

the removal of relevant features and an adequate update of the kernel variables, controlled by the magnitude of the components of $\Delta F(\boldsymbol{\sigma})$. This parameter avoids a strong fluctuation of the kernel variables and negative widths, especially at the first iterations of the algorithm.

5.1 Description of Data Sets

In this subsection we briefly describe the different data sets mentioned above.

Spambase Data Set (Spam)

The Spambase Data set from the UCI data repository [3] presents 57 features and 4,601 instances (2,788 emails labeled as spam and 1,813 ham[2] emails). The data set was created by Mark Hopkins, Erik Reeber, George Forman and Jaap Suermondt from the Hewlett Packard Labs.

Most of the features indicate whether a particular word or character was frequently occurring in the email. The data set presents 48 continuous attributes representing the percentage of words in the email that match a particular word, 6 continuous attributes representing the percentage of characters in the email that match a particular character, the average length of uninterrupted sequences of capital letters, the length of the longest uninterrupted sequences of capital letters and the total number of capital letters in the email. The predictive variables were scaled between 0 and 1.

Phishing Data Set (Phishing)

The phishing corpus used to test the proposed methodology, was an English language phishing email corpus built using Jose Nazario's phishing corpus [20] and the SPAMASSASSIN ham collection. The phishing corpus[3] consists of 4,450 emails manually retrieved from November 27, 2004 to August 7, 2007.

The ham corpus was built using the Spamassassin collection, from the Apache SPAMASSASSIN Project[4], based on a collection of 6,951 ham email messages. Both phishing and ham messages are available in UNIX" mbox format.

All features were extracted according to [15], where first documents are tokenized in order to extract all words in messages. Then, a stopword removal process and stemming of messages' words is realized. When all messages are pre-processed, different feature extraction methodologies are executed, such as structural features [11], keyword extraction [32], singular value decomposition [10], and latent Dirichlet allocation [6]. Finally, all extracted features are combined in order to extract a final set of features that fully characterize a given phishing message.

Table 1 summarizes the relevant information for each spam data set, considering the number of original variables, number of instances and the predominant class proportion (PCP), which is obtain by dividing the number of examples of the predominant class (ham) with the total number of instances.

[2] "Ham" is the name used to describe regular messages that are neither spam nor phishing.

[3] Available at http://bit.ly/jnazariophishing [Online: accesses May 13, 2012].

[4] Available at http://spamassassin.apache.org/publiccorpus/ [Online: accessed May 13, 2012].

Table 1. Descriptive information for each data set

	Variables	Examples	PCP
Spam	57	4,601	0.61
Phishing	273	9,794	0.61

5.2 Results Using Kernel-Penalized Feature Selection

First we compare the results of the best model found using the described model selection procedure for the three different kernel functions presented in Section 3: linear, polynomial, and Gaussian kernel. Table 2 presents the mean classification accuracy and its standard deviation using 10-fold cross-validation. The following set of values for the parameters (penalty parameter C, degree of the polynomial function d and Gaussian Kernel width σ) were used:

$$C = \{0.1, 0.5, 1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 1000\}$$
$$d = \{2, 3, 4, 5, 6, 7, 8, 9\}$$
$$\rho = \{0.1, 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 100\}.$$

Table 2. Mean classification accuracy for three different kernel functions

	N	SVM linear	SVM poly	SVM RBF
Spam	57	93.24±0.4	93.37±0.3	**93.87±0.4**
Phishing	273	98.50±0.1	98.52±0.1	**98.79±0.1**

Best results were achieved for both data sets using the Gaussian Kernel, but this difference is statistically significant only on the second data set. Notice that the Kernel function (6) can be easily modified by incorporating the component-wise product to any suitable Kernel if the best Kernel is not the Gaussian.

In order to study the classification performance of KP-SVM we compared the results for a given number of features (determined by the stopping criterion of our approach) with different feature selection algorithms for SVMs presented before in this paper (SVM-RFE, FSV and Fisher Criterion Score). The results of the mean test accuracy using 10-fold cross-validation are shown in Table 3, where n is the number of features determined by KP-SVM.

Table 3. Mean classification accuracy for four different feature selection strategies

	n	Fisher+SVM	SVM-FSV	RFE-SVM	KP-SVM
Spam	26	92.15±0.3	85.85±0.8	93.18±0.5	**93.52±0.3**
Phishing	30	98.44±0.1	96.93±0.2	98.27±0.2	**98.68±0.1**

The proposed method outperforms all other approaches in terms of classification error for the given number of features obtained by the convergence of KP-SVM, as can be observed from the data in Table 3. The gain in terms of effectiveness is significant in both data sets, with the only exception of RFE-SVM in the first data set, whose performance is lower but not significantly worse than KP-SVM.

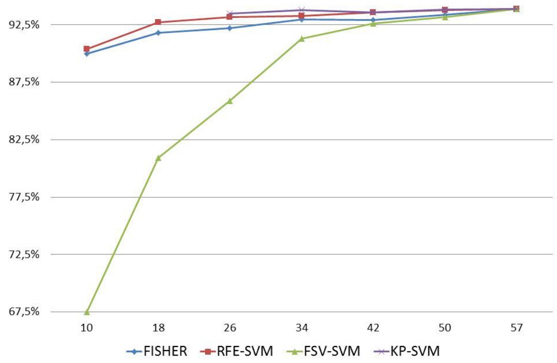


Fig. 1. Mean of test accuracy for Spam vs. the number of ranked variables used for training

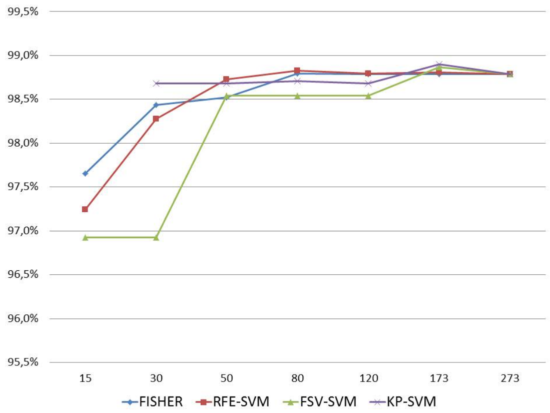


Fig. 2. Mean of test accuracy for Phishing vs. the number of ranked variables used for training

Then we compared the classification performance of the different ranking criteria for feature selection by plotting the mean test accuracy for an increasing number of ranked features used for learning. Figures 1 and 2 show the results for each data set respectively. The proposed KP-SVM approach provides only the information until the stopping criterion is reached.

These experiments underline that the proposed approach, KP-SVM, outperforms other feature selection methods in terms of classification performance for a small number of features in both data sets used. Another important remark is that best classification performance is achieved for KP-SVM considering $C_2 = C$ for the Spam data set and $C_2 = 0.5C$ for the Phishing data set. For both data sets the use of feature penalization outperforms the model obtained using $C_2 = 0$, which can be considered a variant of the ARD model presented in [9]. This fact proves the importance of feature selection in relatively high dimensional data sets, such as the ones presented in this work.

6 Conclusions

In this work we present an embedded approach for feature selection using SVM applied to phishing and spam classification. A comparison with other feature selection methods and classification shows the advantages of our approach:

- It outperforms other techniques in terms of classification accuracy, based on its ability to adjust better to a data set by optimizing the kernel function and simultaneously selecting an optimal feature subset.
- It is not necessary to set *a priori* the number of features to be selected, unlike other feature selection approaches. The algorithm determines the optimal feature number according to the regularization parameter C_2.
- It can be used other kernel functions, such as linear and polynomial kernels.

Even if several parameters have to be tuned, the computational effort can be reduced since the search for an optimal feature subset can be obtained automatically, reducing computational time by avoiding a validation step on finding an adequate number of ranked features. The model selection procedure presented in Section 5 reduces both the computational effort of setting several parameters via cross-validation and the risk of over-fitting. Our empirical results demonstrate the importance of feature penalty to achieve best classification performance under the presented model selection procedure.

Future work has to be done in various directions. First, we consider the extension to highly imbalanced data sets, a very relevant topic in phishing and spam classification, and in pattern recognition in general. Furthermore, the current scenario for spam and phishing classification suggests the extension of the proposed embedded feature selection technique to very large databases as an important research opportunity.

Acknowledgements. Support from the Chilean "Instituto Sistemas Complejos de Ingeniería" (ICM: P-05-004-F, CONICYT: FBO16, www.sistemasdeingenieria.cl) is greatly acknowledged.

References

1. Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S.: A comparison of machine learning techniques for phishing detection. In: eCrime 2007: Proceedings of the Anti-Phishing Working Groups 2nd Annual eCrime Researchers Summit, pp. 60–69. ACM, New York (2007)
2. Ali, S., Smith-Miles, K.A.: A meta-learning approach to automatic kernel selection for support vector machines. Neurocomputing 20(1-3), 173–186 (2006)
3. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
4. Bergholz, A., De Beer, J., Glahn, S., Moens, M.-F., Paass, G., Strobel, S.: New filtering approaches for phishing email. Journal of Computer Security 18(1), 7–35 (2010)
5. Bergholz, A., Chang, J.-H., Paass, G., Reichartz, F., Strobel, S.: Improved phishing detection using model-based features. In: CEAS 2008: Fifth Conference on Email and Anti-Spam, Mountain View, CA, USA (2008)
6. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. J. Mach. Learn. Res. 3, 993–1022 (2003)

7. Bradley, P., Mangasarian, O.: Feature selection via concave minimization and support vector machines. In: Int. Conference on Machine Learning, pp. 82–90 (1998)
8. Canu, S., Grandvalet, Y.: Adaptive scaling for feature selection in SVMs. In: Advances in Neural Information Processing Systems 15, pp. 553–560. MIT Press (2002)
9. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning 46, 131–159 (2002)
10. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41, 391–407 (1990)
11. Fette, I., Sadeh, N., Tomasic, A.: Learning to detect phishing emails. In: WWW 2007: Proceedings of the 16th International Conference on World Wide Web, pp. 649–656. ACM, New York (2007)
12. Goodman, J., Cormack, G.V., Heckerman, D.: Spam and the ongoing battle for the inbox. Commun. ACM 50(2), 24–33 (2007)
13. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature extraction, foundations and applications. Springer, Berlin (2006)
14. Guyon, I., Saffari, A., Dror, G., Cawley, G.: Model selection: Beyond the bayesian frequentist divide. Journal of Machine Learning Research 11, 61–87 (2009)
15. L'Huillier, G., Hevia, A., Weber, R., Rios, S.: Latent semantic analysis and keyword extraction for phishing classification. In: ISI 2010: Proceedings of the IEEE International Conference on Intelligence and Security Informatics, pp. 129–131. IEEE, Vancouver (2010)
16. Ma, J., Saul, L.K., Savage, S., Voelker, G.M.: Identifying suspicious urls: an application of large-scale online learning. In: Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, pp. 681–688. ACM, New York (2009)
17. Maldonado, S., Weber, R.: A wrapper method for feature selection using support vector machines. Information Sciences 179, 2208–2217 (2009)
18. Maldonado, S., Weber, R., Basak, J.: Kernel-penalized SVM for feature selection. Information Sciences 181(1), 115–128 (2011)
19. Inc. McAfee. The carbon footprint of email spam report. Technical report (2008)
20. Nazario, J.: Phishing corpus (2004-2007)
21. Neumann, J., Schnörr, C., Steidl, G.: Combined svm-based feature selection and classification. Machine Learning 61, 129–150 (2005)
22. Perkins, S., Lacker, K., Theiler, J.: Grafting: Fast incremental feature selection by gradient descent in function space. Journal of Machine Learning research 3, 1333–1356 (2003)
23. Rakotomamonjy, A.: Variable selection using SVM-based criteria. Journal of Machine Learning Research 3, 1357–1370 (2003)
24. Schölkopf, B., Smola, A.J.: Learning with Kernels. MIT Press, Cambridge (2002)
25. Tang, Y., He, Y., Krasser, S.: Highly scalable svm modeling with random granulation for spam sender detection. In: Proceedings of the 8th International Conference in Machine Learning and Applications, ICMLA 2008, pp. 659–664. IEEE Computer Society (2008)
26. Tang, Y., Krasser, S., Alperovitch, D., Judge, P.: Spam sender detection with classification modeling on highly imbalanced mail server behavior data. In: Proceedings of the International Conference on Artificial Intelligence and Pattern Recognition, AIPR 2008, pp. 174–180. ISRST (2008)
27. Tang, Y., Krasser, S., He, Y., Yang, W., Alperovitch, D.: Support vector machines and random forests modeling for spam senders behavior analysis. In: Proceedings of the Global Telecommunications Conference, GLOBECOM 2008, pp. 2174–2178. IEEE Computer Society (2008)

28. Tang, Y., Krasser, S., Judge, P., Zhang, Y.-Q.: Fast and effective spam sender detection with granular svm on highly imbalanced server behavior data. In: Proceedings of the 2nd International Conference on Collaborative Computing: Networking, Applications and Worksharing, COLCOM 2006, pp. 1–6. IEEE Computer Society (2006)
29. Taylor, B., Fingal, D., Aberdeen, D.: The war against spam: A report from the front line. In: NIPS 2007 Workshop on Machine Learning in Adversarial Environments for Computer Security (2007)
30. Thomas, K., Grier, C., Ma, J., Paxson, V., Song, D.: Design and evaluation of a real-time url spam filtering service. In: IEEE Symposium on Security and Privacy 2011, pp. 1–16. IEEE Press (2011)
31. Vapnik, V.: Statistical Learning Theory. John Wiley and Sons (1998)
32. Velásquez, J.D., Rios, S.A., Bassi, A., Yasuda, H., Aoki, T.: Towards the identification of keywords in the web site text content: A methodological approach. International Journal of Web Information Systems 1(1), 53–57 (2005)
33. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: The use of zero-norm with linear models and kernel methods. Journal of Machine Learning Research 3, 1439–1461 (2003)
34. Weston, J., Mukherjee, S., Chapelle, O., Ponntil, M., Poggio, T., Vapnik, V.: Feature selection for SVMs. In: Advances in Neural Information Processing Systems 13, vol. 13 (2001)

Instance Selection Methods and Resampling Techniques for Dissimilarity Representation with Imbalanced Data Sets

M. Millán-Giraldo, V. García, and J.S. Sánchez

Institute of New Imaging Technologies
Department of Computer Languages and Systems, Universitat Jaume I
Av. Vicent Sos Baynat s/n, 12071 Castellón de la Plana, Spain
{mmillan,jimenezv,sanchez}@uji.es
http://www.init.uji.es

Abstract. In the dissimilarity representation approach, the dimension reduction of the dissimilarity space is addressed by using instance selection methods. Several studies have shown that these methods work well on small data sets. Also, the uniformity of the instances distribution can be obtained when the classes are evenly spread and balanced. However, many real-world problems are characterized by an imbalanced class distribution. In this paper, we address the problem of instance selection for constructing the dissimilarity space in the imbalanced data context. Class imbalance is handled by resampling the data set, whereas instance selection is applied to find a small representation set. Experimental results demonstrate the significance of the joint use of resampling techniques and instance selection methods to improve the performance of classifiers trained on dissimilarity representation.

Keywords: Instance selection, Dissimilarity representation, Resampling techniques, Imbalanced data sets.

1 Introduction

The statistical pattern recognition approach traditionally represents the objects in vector spaces by a set of measurable features. However, this approach presents some drawbacks: (i) objects of different classes may be represented by the same feature vectors and (ii) the classifiers could be affected by the variation of feature sets [1]. An alternative approach to the feature-based representation that overcomes these problems is the dissimilarity representation paradigm proposed by Pekaslka and Duin [2]. Here, the objects are represented by their dissimilarity or distance values to the other objects in the set.

The construction of a new vector space from a dissimilarity representation is carried out in two ways [3]: (i) Euclidean embedding and (ii) the dissimilarity space. The former case is based on embedding the given non-Euclidean dissimilarity data into a vector space preserving the distances between objects as good as possible in comparison to the original dissimilarities. The second way postulates an Euclidean vector space defined by the dissimilarities vectors. This method considers the dissimilarity matrix

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 149–160.
DOI: 10.1007/978-3-642-36530-0_12

as a new training data set, where the set of rows (dissimilarities) vectors (one for each object) represents individual training samples and the columns form the dimensions of the so-called dissimilarity space. For its construction, the pairwise dissimilarities are computed between a given object and objects from the representation set R. In general, a representation set is a set of chosen prototypes of the training set T. Sometimes, R can be chosen as the whole training set.

In the dissimilarity space, the dimensionality is determined by the size of the representation set. When all training objects are used to build the representation set, the dimension of the dissimilarity space is equal to $|T|$, which may impose a computational burden on the classifier. To overcome this problem, numerous works have proposed to use and develop instance selection methods (instance selection methods) for finding a small reduced representation set (from the training data) capable of achieving a good trade-off between classification accuracy and computational efficiency [4,5,6,7,8,9,10]. Results using instance selection methods have shown a good performance for small training sets. Likewise, when the classes are evenly spread and balanced, it is possible to gain a uniform prototypes distribution. However, in many real-world problems, there exists an extremely skewed difference between the class ratios of prior probabilities. This data complexity, known as the class imbalance problem [11], may affect the instance selection process to obtain reduced representation sets that does not reflect the true distribution [9].

The class imbalance problem occurs when one class vastly outnumbers the other class, which is usually the most important one and with the highest misclassification costs. Instances from the minority and majority classes are often referred to as positive and negative, respectively. Several solutions have been proposed to deal with this data complexity. One of the most investigated is resampling, which aims at balancing the original data set, either by over-sampling the minority class [12,13] and/or by under-sampling the majority class [14,15], until the classes are approximately equally represented.

Although class imbalance has been extensively studied for binary classification problems, very few approaches explore the class imbalance problem in the dissimilarity space [16,17,18]. Besides, to the best of our knowledge, no work has been carried out on how to select a small representation set for constructing the dissimilarity space on imbalanced data sets.

This paper investigates some strategies to select a reduced representation set and manage the class imbalance for dissimilarity representation. In order to face such a problem, this work focuses on the joint use of instance selection methods and resampling techniques. To this end, we will carry out experiments over real data sets, employing four renowned instance selection methods and two resampling algorithms. All techniques are evaluated in terms of their geometric mean of accuracies, and then compared for statistical differences using the Friedman's average rank test and the Nemenyi's post hoc test.

The rest of the paper is outlined as follows. Section 2 provides a summary of the classification problem in dissimilarity representation. Section 3 presents a brief overview of instance selection methods. An introduction to resampling algorithms is provided in Section 4. In Section 5, the experimental setup is described. Next, in Section 6, the results are showed and discussed. Finally, Section 7 concludes the present study.

2 Dissimilarity Space

In traditional pattern recognition algorithms, objects are represented by a vector of features, in which the dimensionality of the feature space is given by the number of features employed to describe the objects. On the contrary, in the dissimilarity space, objects are represented by dissimilarity vectors, where each element of a vector relates an object with other objects [2].

Given a training set of n objects, $T = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, a new set of r representative objects of the problem, called *prototypes*, is obtained from T. This set of prototypes, which contains information of all classes in T, is known as *representation set*, $R = \{\mathbf{p}_1, \ldots, \mathbf{p}_r\}$. The amount of prototypes (r) in R determines the dimension of the dissimilarity space. Several methods have been proposed in the literature to select this set of prototypes; for example, Pekalska et al. studied the random and systematic selection procedures for the normal density-based quadratic classifier [8].

In dissimilarity-based classification, some dissimilarity measure d has to be employed to compute the proximity between objects. Given the pair of objects $\mathbf{x} = (x_1, x_2, \ldots, x_m)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_m)$, the dissimilarity measure d must satisfy one or more of the usual conditions for a metric [19]: non-negativity, identity of indiscernibles, symmetry and triangle inequality.

Usually, the dissimilarity measure used to represent objects by proximities corresponds to the Euclidean distance between two objects $\mathbf{x}$ and $\mathbf{y}$, that is,

$$d(\mathbf{x}, \mathbf{y}) = (\sum_{j=1}^{m} (x_j - y_j)^2)^{1/2} \tag{1}$$

where m is the number of features. Then, the proximity between the i-th object in T, $\mathbf{x}_i$, and all prototypes in R is

$$D(\mathbf{x}_i, R) = \{d(\mathbf{x}_i, \mathbf{p}_1), \ldots, d(\mathbf{x}_i, \mathbf{p}_r)\} \tag{2}$$

which is a vector with r distances that associates $\mathbf{x}_i$ with all objects in R. By doing $D(T, R)$, a $n \times r$ dissimilarity matrix is obtained, which refers to the distances from the objects in the training set to all objects in the representation set and it will be further used to built the classifier in the dissimilarity space.

In this paper, we will use the Euclidean distance measure. Given a test set S, the proximity between objects in S and prototypes in R is also computed, giving a dissimilarity matrix $D(R, S)$. Thus, the test set S can be evaluated with the classifier built in the dissimilarity space.

3 Instance Selection Methods

In the framework of the dissimilarity representation, the instance selection methods are used to find a small representation set for reducing the computational effort, while preserving the classification accuracy. Research on this topic has proposed solutions to be applied in the dissimilarity space [7,10] and/or in the original feature space [4,6]. In this work, we are interested in techniques that fall into the second group. A full review of instance selection methods used in the dissimilarity space can be found in the work by Plasencia-Calaña et al. [9].

A straightforward instance selection method is the random selection (RS) which seeks k prototypes randomly from the training set without taking into account the class labels. This method can be applied in a stratified fashion, where k prototypes from each class are selected. This allows to produce a more uniform reduced data set with respect to the class distribution.

Other more "intelligent" method seeks to retain points that are closer to the decision boundaries, while removing internal points. One of the earliest methods is the Condensed Nearest Neighbour (CNN) proposed by Hart [20]. This algorithm finds a condensed subset CS from the training set T that correctly classifies every prototype in T using the nearest neighbour (1-NN) rule. This approach starts by randomly selecting one pattern belonging to each class from T and putting them into CS. Each remaining sample in T is then classified using the objects in the current CS. If a sample in T is misclassified, it is added to CS. This process ends when no sample in T is misclassified by CS. Nevertheless, this algorithm does not guarantee minimality and both the quality and size of the condensed subset depend on the order in which the training objects are presented to the algorithm.

To overcome the aforementioned issues, Barandela et al. [21] proposed the Modified Selective Subset (MSS) method, which reduces the training set size while preserving the original decision boundaries as much as possible.

4 Resampling Techniques

Resampling consists of artificially balancing the original data set, either by over-sampling the minority class and/or by under-sampling the majority class, until the problem classes are approximately equally represented. Both strategies can be applied in any learning system, since they act as a preprocessing phase, allowing the learning system to receive the training objects as if they belonged to a well-balanced data set. Thus, any bias of the system towards the majority class due to the different proportion of examples per class would be expected to be suppressed. The simplest method to increase/reduce of the minority/majority class corresponds to non-heuristic methods that aim to balance the class distribution through the random replication/elimination of positive/negative objects. Nevertheless, these methods have shown important drawbacks. Random over-sampling may increase the likelihood of overfitting, since it makes exact copies of the minority class objects. On the other hand, random under-sampling may discard data potentially important for the classification process. Despite this problem, it has empirically been shown to be one of the most effective resampling methods. In order to overcome these drawbacks, several authors have developed *focused resampling* algorithms that create balanced data sets in an intelligent way.

Chawla et al. [12] proposed an over-sampling technique that generates new synthetic minority objects by interpolating between several positive examples that lie close together. This method, called SMOTE (Synthetic Minority Oversampling TEchnique), allows the classifier to build larger decision regions that contain nearby objects from the minority class. From the original SMOTE algorithm, several modifications have been proposed in the literature, most of them pursuing to determine the region in which the positive examples should be generated. For instance, Borderline-SMOTE [13] consists of using only positive examples close to the decision boundary, since these are more likely to be misclassified.

Unlike the random method, many proposals are based on a more intelligent selection of the majority class examples to eliminate. For example, Kubat and Matwin [22] proposed an under-sampling technique named one-sided selection, that selectively removes only those negative instances that are "redundant" or that "border" the minority class objects (they assume that these bordering cases are noise). In contrast to the one-sided selection technique, the so-called neighborhood cleaning rule emphasizes more data cleaning than data reduction. To this end, Wilson's editing is used to identify and remove noisy negative objects. Similarly, Barandela et al. [14] introduced a method that eliminates not only noisy examples of the majority class by means of Wilsons editing, but also redundant examples through the MSS condensing algorithm.

5 Experimental Setup

Experiments have been carried out over 13 data sets taken from the UCI Machine Learning Database Repository [23] and a private library (http://www.vision.uji.es/∼sanchez/Databases/). All data sets have been transformed into two-class problems by keeping one original class (the minority class) and joining the objects of the remaining classes (giving the majority class). For example, in Segmentation database the objects of classes 1, 2, 3, 4 and 6 have been joined to shape a unique majority class and the original class 5 has been left as the minority class (see a summary in Table 1).

A stratified five-fold cross validation method has been adopted for the present experiments: each original data set has been randomly divided into five parts or equal (or approximately equal) size. For each fold, four of the parts have been pooled as the training data, and the remaining block has been employed as an independent set. The training sets (in the feature space) have been preprocessed by SMOTE and random under-sampling (RUS) to handle the class imbalance problem. Also, three instance selection methods previously described have been applied over the original training sets (without any preprocessing) and the balanced data sets to gain a reduced representation set: random selection (R), the Condensed Nearest Neighbour (CNN) and the Modified Selective Subset method (MSS). In the case of the random selection method, we have selected 50% (R50) and 100% (R100) of objects from each class. Next, we have computed the dissimilarity matrix $D(T, R)$ by using either imbalanced and balanced training sets with their respective representative sets. Finally, two learners, Fisher and 1-NN classifiers, were applied on the dissimilarity space.

5.1 Performance Evaluation in Class Imbalance Problems

Evaluation of classification performance plays a critical role in the design of a learning system and therefore, the use of an appropriate measure becomes as important as the

Table 1. Data sets used in the experiments

Data Set	Positive Examples	Negative Examples	Classes	Majority Class
Breast	81	196	2	1
Ecoli	35	301	8	1,2,3,5,6,7,8
German	300	700	2	1
Glass	17	197	9	1,2,4,5,6,7,8,9
Haberman	81	225	2	1
Laryngeal$_2$	53	639	2	1
Phoneme	1586	3818	2	1
Pima	268	500	2	1
Scrapie	531	2582	2	1
Segmentation	330	1980	6	1,2,3,4,6
Spambase	1813	2788	2	1
Vehicle	212	634	4	2,3,4
Yeast	429	1055	10	1,3,4,5,6,7,8,9,10

Table 2. Confusion matrix for a two-class decision problem

	Predicted positive	Predicted negative
Actual positive	True Positive (TP)	False Negative (FN)
Actual negative	False Positive (FP)	True Negative (TN)

selection of a good algorithm to successfully tackle a given problem. Traditionally, standard performance metrics have been classification accuracy and/or error rates. For a two-class problem, these can be easily derived from a 2×2 confusion matrix as that given in Table 2.

The classification accuracy (Acc) evaluates the effectiveness of the learner by its percentage of correct predictions,

$$Acc = \frac{TP + TN}{TP + FN + TN + FP} \tag{3}$$

The counterpart of accuracy is the error rate, which evaluates a classifier by its percentage of incorrect predictions.

$$Err = \frac{FP + FN}{TP + FN + TN + FP} = 1 - Acc \tag{4}$$

Empirical and theoretical evidences show that these measures are strongly biased with respect to data imbalance and proportions of correct and incorrect classifications [24,25,26,27,28]. In a binary decision problem, a learner predicts objects as either positive or negative; if very few examples belong to the positive class, a naive learning system could obtain a very high accuracy by just classifying all objects as negative. However, this is useless in most real domains because the class of interest is generally the positive one. Therefore, evaluators such as accuracy or error rate appear to be inappropriate for class imbalanced data, thus motivating the search for other measures based on some straightforward indexes, which have also been formulated from a 2×2 confusion matrix as that in Table 2. For example, Kubat and Matwin [22] use the geometric mean of accuracies measured separately on each class,

$$Gmean = \sqrt{TPr \cdot TNr} \tag{5}$$

where $TPr = TP/(TP+FN)$ is the percentage of positive examples that are correctly classified, while, $TNr = TN/(TN + FP)$ is defined as the proportion of negative examples that are correctly classified.

The $Gmean$ is associated to a point on the ROC curve, and the idea is to maximize the accuracies of both classes while keeping them balanced. It can be interpreted as a kind of good trade-off between both rates because a high value occurs when they both are also high, whereas a low value is related to at least one low rate.

5.2 Statistical Significance Tests

A common way to compare two classifiers over a set of problems is the Student's paired t-test. However, this appears to be conceptually inappropriate and statistically unsafe because parametric tests are based on a variety of assumptions (independence, normality and homoscedasticity) that are often violated due to the nature of the problems [29]. In general, the non-parametric tests (e.g., Wilcoxon and Friedman tests) should be preferred over the parametric ones, especially in multi-problem analysis, because they do not assume normal distributions or homogeneity of variance [29,30].

The Friedman test is based on the average ranked performances of a collection of techniques on each data set separately. Under the null-hypothesis, which states that all the algorithms are equivalent, the Friedman statistic can be computed as follows:

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_j R_j^2 - \frac{K(K+1)^2}{4} \right] \tag{6}$$

where N denotes the number of data sets, K is the total number of algorithms and R_j is the average ranks of algorithms. The χ_F^2 is distributed according to the Chi-square distribution with $K - 1$ degrees of freedom, when N (number of data sets) and K (number of algorithms) are big enough. However, it has been demonstrated that the Friedman statistic produces an undesirably conservative effect. In order to overcome the conservativeness, Iman and Davenport [31] proposed a better statistic distributed according to the $F-$distribution with $K - 1$ and $(K - 1)(N - 1)$ degrees of freedom,

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1) - \chi_F^2} \tag{7}$$

When the null-hypothesis is rejected, we can use post-hoc tests in order to find the particular pairwise comparisons that produce statistical significant differences. A post-hoc test compares a control algorithm opposite to the remainder techniques, making possible to define a collection of hypothesis around the control method. The Nemenyi post-hoc test, which is analogous to the Tukey test for ANOVA, states that the performances of two or more algorithms are significantly different if their average ranks are at least as great as their critical difference (CD) with a certain level of significance:

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}} \tag{8}$$

where q_α is a critical value based on the studentised range statistic divided by $\sqrt{2}$. For the present set-up, the corresponding critical values are $q_{0.05} = 3.268$ and $q_{0.10} = 3.030$, for $\alpha = 0.05$ and $\alpha = 0.10$, respectively.

6 Results and Discussion

Table 3 reports the results, in terms of $Gmean$, given by Fisher and 1-NN classifiers on the 13 data sets. For each strategy here proposed, the average Friedman is also shown. The technique achieving the best $Gmean$ on each data set as well as the average Friedman's ranking is highlighted in bold. From these results, several comments can be drawn:

- In general, when dissimilarity spaces are constructed on balanced datasets (for any instance selection method), the $Gmean$ values are significantly better than those obtained by using the original training set (without preprocessing).
- The benefits of resampling are much more obvious in the Glass data set, where $Gmean$ increases from 0.00 (all minority objects were misclassified) to 0.686 with the Fisher classifier.
- Paradoxically, for both 1-NN and Fisher classifiers, the random selection method achieves the best classification results.
- The RUS+R100 strategy has the best Friedman ranking in the case of Fisher, whereas SMOTE with both versions of random selection provide the best average ranking in 1-NN. As claimed by Pekalska and Duin [2], the nearest neighbours classifiers may require a much larger representation set to generate a higher accuracy.
- Although R100 seems a good strategy, it is important to remark that this technique may produce an increase in the computational cost. This problem might grow up if it is combined with an oversampling technique.

In order to check whether there are significantly differences in the results, we computed the Iman-Davenport's statistic using the Eq. 7 described above. The computation yields $F_F = 9.585$ and $F_F = 4.486$ for 1-NN and Fisher classifiers, respectively. The critical value for the F distribution with 12-1=11 and (12-1)(13-1)=132 degrees of freedom considering two levels of confidence, $\alpha = 0.05$ and $\alpha = 0.05$, are $F(11{,}132)_{0.05} = 1.86$ and $F(11{,}132)_{0.10} = 1.62$, so the null hypothesis that all strategies here explored perform equally well can be rejected. Therefore, we can apply Nemenyi's post hoc test in order to detect the set of strategies that are significantly worse than the control method (the method with the best Friedman's rank).

The results of the Nemenyi's post hoc test can be found in Fig. 1. For each classifier and level of confidence, the plot shows the strategies here proposed, which have been

Table 3. Average *Gmean* results obtained with Fisher and 1-NN classifiers (for each data set, the best case is highlighted in bold

	Fisher											
	Original				SMOTE				RUS			
	R50	R100	CNN	MSS	R50	R100	CNN	MSS	R50	R100	CNN	MSS
Breast	0.580	0.591	0.578	0.610	0.628	0.586	0.614	0.615	0.632	**0.637**	0.569	0.603
German	0.645	0.652	0.634	0.666	0.695	0.691	0.626	0.650	**0.697**	0.694	0.662	0.662
Laryngeal2	0.838	0.884	0.883	0.818	0.910	0.905	0.668	0.768	**0.920**	0.916	0.796	0.730
Pima	0.669	0.657	0.656	0.660	0.676	0.672	0.592	0.676	**0.697**	0.681	0.680	0.682
Scrapie	0.415	0.442	0.442	0.371	0.516	0.516	0.441	0.452	0.582	**0.592**	0.403	0.380
Spambase	0.885	0.897	0.895	0.897	0.892	0.898	0.862	0.890	0.880	0.890	**0.900**	0.897
Vehicle	0.630	0.626	0.624	0.626	0.694	0.661	0.637	0.671	0.750	**0.758**	0.627	0.624
Ecoli	0.773	0.676	0.695	0.651	0.845	0.834	0.547	0.707	0.851	**0.855**	0.757	0.774
Glass	0.000	0.000	0.000	0.000	0.635	**0.686**	0.000	0.569	0.624	0.614	0.000	0.000
Haberman	0.533	0.534	0.518	0.572	**0.610**	0.573	0.517	0.567	0.607	0.606	0.533	0.523
Segmentation	0.907	0.936	0.936	0.928	0.938	**0.949**	0.772	0.924	0.804	0.810	0.858	0.789
Yeast	0.667	0.671	0.669	0.672	0.717	0.701	0.619	0.681	0.723	**0.725**	0.678	0.669
Phoneme	0.871	0.884	0.883	0.882	0.879	**0.890**	0.710	0.871	0.870	0.885	0.880	0.875
Avg. Ranking	8.62	7.04	8.38	7.35	3.46	3.81	10.46	6.58	3.85	**3.04**	7.31	8.12
	1-NN											
	Original				SMOTE				RUS			
	R50	R100	CNN	MSS	RS50	R100	CNN	MSS	R50	R100	CNN	MSS
Breast	0.510	0.533	0.561	0.544	0.561	0.537	0.520	0.534	**0.592**	0.585	0.532	0.504
German	0.528	0.527	0.527	0.520	0.543	0.549	0.525	0.524	0.563	**0.564**	0.522	0.515
Laryngeal2	0.711	0.738	0.708	0.681	0.918	**0.907**	0.681	0.678	0.844	0.864	0.740	0.686
Pima	0.605	0.596	0.595	0.597	0.621	0.610	0.592	0.603	0.626	**0.631**	0.595	0.596
Scrapie	0.509	0.510	0.508	0.507	0.504	0.502	0.511	**0.515**	0.464	0.457	0.506	0.514
Spambase	0.732	0.734	0.733	0.734	0.732	0.732	0.733	0.731	0.733	0.732	**0.735**	0.732
Vehicle	0.561	0.557	0.557	0.567	0.604	0.607	0.555	0.579	0.611	**0.629**	0.551	0.555
Ecoli	0.715	0.716	0.708	0.687	0.793	**0.812**	0.697	0.712	0.747	0.764	0.681	0.695
Glass	0.541	0.541	0.000	0.000	0.732	**0.737**	0.594	0.555	0.647	0.662	0.000	0.000
Haberman	0.571	0.579	0.575	0.578	0.587	**0.606**	0.588	0.576	0.577	0.585	0.586	0.580
Segmentation	0.894	0.894	0.891	0.881	**0.911**	**0.911**	0.883	0.883	0.852	0.852	0.887	0.883
Yeast	0.643	0.635	0.638	0.645	0.676	0.674	0.642	0.640	**0.682**	**0.682**	0.632	0.637
Phoneme	0.843	0.844	0.844	0.840	0.861	0.861	0.844	0.839	0.850	**0.851**	0.845	0.836
Avg. Ranking	7.2	6.5	7.6	8.0	**3.6**	**3.6**	7.5	7.8	4.7	4.3	8.1	9.2

listed in ascending order based in their ranking values (on the $y-$axis), and the ranking obtained by the Friedman test is displayed on the $x-$axis. A horizontal dashed line is drawn to represent the end of the best performing technique (the control method). All methods which are on the right side of this line belong to the strategies whose performance is significantly worse than the control method. From these results, in the case of Fisher classifier, the strategies RUS+MSS, CNN, R50 and SMOTE+CNN perform

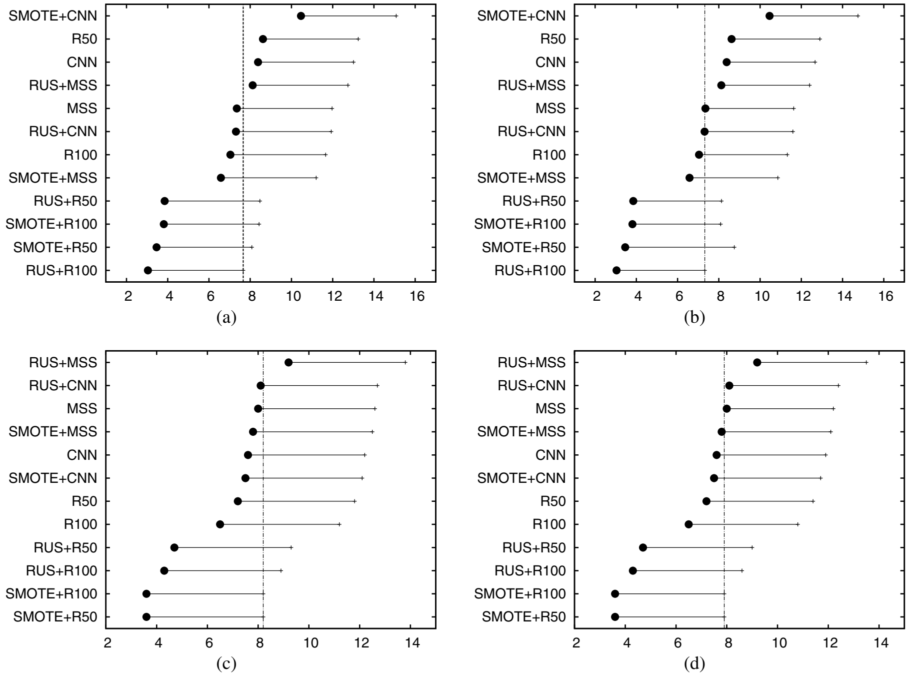


Fig. 1. Nemenyi's critical-difference diagram using Fisher (upper) and 1-NN (below) classifiers with two levels of confidence: (a) (c) $\alpha = 0.05$ and (b) (d) $\alpha = 0.10$

significantly worse than the RUS+R100 technique. For 1-NN classifier, RUS+CNN and RUS+CNN perform even worse than SMOTE with R50 and R100.

7 Conclusions

In this paper, we have analyzed the effect of the representation set in the dissimilarity space when data are imbalanced. For this purpose, we have evaluated four prototype selection methods and two resampling techniques (one corresponding to under-sampling and one to over-sampling). All these algorithms have also been applied to the data sets before representing them by dissimilarities, with the aim to analyze the influence of having a balanced representation set on the classification performance.

Using the Fisher and 1-NN classifiers, it has been observed that in general, the best classification results in terms of the geometric mean of accuracies are obtained when the training data sets have previously been preprocessed by using some resampling algorithm.

Acknowledgements. This work has partially been supported by the Spanish Ministry of Education and Science under grants CSD2007–00018, AYA2008–05965–0596 and TIN2009–14205, the Fundació Caixa Castelló-Bancaixa under grant P1–1B2009–04, and the Generalitat Valenciana under grant PROMETEO/2010/028.

References

1. Duin, R.P.W., Pekalska, E.: The dissimilarity space: Bridging structural and statistical pattern recognition. Pattern Recognition Letters 33, 826–832 (2012)
2. Pękalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. Pattern Recognition Letters 23, 943–956 (2002)
3. Paclik, P., Duin, R.P.W.: Dissimilarity-based classification of spectra: computational issues. Real-Time Imaging 9, 237–244 (2003)
4. Kim, S.W., Oommen, B.J.: On using prototype reduction schemes to optimize dissimilarity-based classification. Pattern Recognition 40, 2946–2957 (2007)
5. Kim, S.W.: An empirical evaluation on dimensionality reduction schemes for dissimilarity-based classifications. Pattern Recognition Letters 32, 816–823 (2011)
6. Lozano, M., Sotoca, J.M., Sánchez, J.S., Pla, F., Pkalska, E., Duin, R.P.W.: Experimental study on prototype optimisation algorithms for prototype-based classification in vector spaces. Pattern Recognition 39, 1827–1838 (2006)
7. Pekalska, E., Duin, R.P.W.: Prototype selection for finding efficient representations of dissimilarity data. In: Proc. 16th International Conference on Pattern Recognition, vol. 3, pp. 37–40 (2002)
8. Pekalska, E., Duin, R.P.W., Paclik, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition 39, 189–208 (2006)
9. Plasencia-Calaña, Y., García-Reyes, E., Duin, R.P.W.: Prototype selection methods for dissimilarity space classification. Technical report, Advanced Technologies Application Center CENATAV (2010)
10. Plasencia-Calaña, Y., García-Reyes, E., Orozco-Alzate, M., Duin, R.P.W.: Prototype selection for dissimilarity representation by a genetic algorithm. In: Proc. 20th International Conference on Pattern Recognition, pp. 177–180 (2010)
11. Fernández, A., García, S., Herrera, F.: Addressing the classification with imbalanced data: Open problems and new challenges on class distribution. In: Proc. 6th International Conference on Hybrid Artificial Intelligent Systems, pp. 1–10 (2011)
12. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling TEchnique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
13. Han, H., Wang, W.Y., Mao, B.H.: Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In: Proc. International Conference on Intelligent Computing, pp. 878–887 (2005)
14. Barandela, R., Sánchez, J., García, V., Rangel, E.: Strategies for learning in class imbalance problems. Pattern Recognition 36, 849–851 (2003)
15. García, S., Herrera, F.: Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. Evolutionary Computation 17, 275–306 (2009)
16. Koknar-Tezel, S., Latecki, L.: Improving SVM classification on imbalanced time series data sets with ghost points. Knowledge and Information Systems 28, 1–23 (2011)
17. Sousa, A., Mendonca, A., Campilho, A.: Minimizing the imbalance problem in chromatographic profile classification with one-class classifiers. In: Proc. 5th International Conference on Image Analysis and Recognition, pp. 413–422 (2008)
18. Sousa, A., Mendonca, A., Campilho, A.: Dissimilarity-based classification of chromatographic profiles. Pattern Analysis & Applications 11, 409–423 (2008)
19. Duin, R.P.W., Pękalska, E.: The Dissimilarity Representation for Structural Pattern Recognition. In: San Martin, C., Kim, S.-W. (eds.) CIARP 2011. LNCS, vol. 7042, pp. 1–24. Springer, Heidelberg (2011)
20. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. on Information Theory 14, 515–516 (1968)

21. Barandela, R., Ferri, F.J., Sánchez, J.S.: Decision boundary preserving prototype selection for nearest neighbor classification. International Journal of Pattern Recognition and Artificial Intelligence 19, 787–806 (2005)
22. Kubat, M., Matwin, S.: Addressing the curse of imbalanced training sets: one-sided selection. In: Proc. 14th International Conference on Machine Learning, Nashville, USA, pp. 179–186 (1997)
23. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
24. Daskalaki, S., Kopanas, I., Avouris, N.: Evaluation of classifiers for an uneven class distribution problem. Applied Artificial Intelligence 20, 381–417 (2006)
25. Fatourechi, M., Ward, R., Mason, S., Huggins, J., Schlogl, A., Birch, G.: Comparison of evaluation metrics in classification applications with imbalanced datasets. In: Proc. 7th International Conference on Machine Learning and Applications, pp. 777–782 (2008)
26. Huang, J., Ling, C.X.: Constructing new and better evaluation measures for machine learning. In: Proc. 20th International Joint Conference on Artificial Intelligence, pp. 859–864 (2007)
27. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proc. 3rd International Conference on Knowledge Discovery and Data Mining, pp. 43–48 (1997)
28. Sokolova, M., Lapalme, G.: A systematic analysis of performance measures for classification tasks. Information Processing & Management 45, 427–437 (2009)
29. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
30. García, S., Fernández, A., Luengo, J., Herrera, F.: Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences 180, 2044–2064 (2010)
31. Iman, R.L., Davenport, J.M.: Approximations of the critical region of the friedman statistic. Communications in Statistics – Theory and Methods 9, 571–595 (1980)

Simplex Decompositions Using Singular Values Decomposition

Madhusudana Shashanka and Michael Giering

United Technologies Research Center, East Hartford, CT 06108, U.S.A.
{ShashaM,GierinMJ}@utrc.utc.com

Abstract. Probabilistic Latent Semantic Analysis (PLSA) is a popular technique to analyze non-negative data where multinomial distributions underlying every data vector are expressed as linear combinations of a set of *basis distributions*. These learned basis distributions that characterize the dataset lie on the standard simplex and themselves represent corners of a simplex within which all data approximations lie. In this paper, we describe a novel method to extend the PLSA decomposition where the *bases* are not constrained to lie on the standard simplex and thus are better able to characterize the data. The locations of PLSA basis distributions on the standard simplex depend on how the dataset is aligned with respect to the standard simplex. If the directions of maximum variance of the dataset are orthogonal to the standard simplex, then the PLSA bases will give a poor representation of the dataset. Our approach overcomes this drawback by utilizing Singular Values Decomposition (SVD) to identify the directions of maximum variance, and transforming the dataset to align these directions parallel to the standard simplex before performing PLSA. The learned PLSA features are then transformed back into the data space. The effectiveness of the proposed approach is demonstrated with experiments on synthetic data.

Keywords: Matrix factorization, Probabilistic Latent Semantic Analysis (PLSA), Nonnegative Matrix Factorization (NMF), Singular Values Decomposition (SVD).

1 Introduction

The need for analyzing non-negative data arises in several applications such as computer vision, semantic analysis and gene expression analysis among others. Nonnegative Matrix Factorization (NMF) [1,2] was specifically proposed to analyze such data where every data vector is expressed as a linear combination of a set of characteristic *basis vectors*. The weights with which these vectors combine differ from data point to data point. All entries of the basis vectors and the weights are constrained to be nonnegative. The nonnegativity constraint produces basis vectors that can only combine additively without any cross-cancellations and thus can be intuitively thought of as *building blocks* of the dataset. Given these desirable properties, the technique has found wide use across different applications. However, one of the main drawbacks of NMF is that the *energies* of data vectors is split between the basis vectors and mixture weights during decomposition. In other words, the basis vectors may lie in an entirely different part of the data space making any geometric interpretation meaningless.

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 161–170.
DOI: 10.1007/978-3-642-36530-0_13

Probabilistic Latent Semantic Analysis (PLSA) [3] is a related method with probabilistic foundations which was proposed around the same time in the context of semantic analysis of document corpora. A corpus of documents is represented as a matrix where each column vector corresponds to a document and each row corresponds to a word in the vocabulary and the entry corresponds to the numer of times the word appeared in the document. PLSA decomposes this matrix as a linear combination of a set of multinomial distributions over the words called *topics* where the weight vectors are multinomial distributions as well. Non-negativity constraint is imposed implicitly because the extracted topics or *basis distributions* and weights represent probabilities. It has been shown that the underlying computations in NMF and PLSA are identical [4,5]. However, unlike NMF where there are no additional constraints beyond nonegativity, PLSA bases and weights being multinomial distriutions also have the contraint that the entries sum to 1. Since the weights sum to 1, the PLSA approximations of the data can be thought of as lying within a simplex defined by the basis distriutions. Shashanka [6] formalizes this geometric intuition as *Simplex Decompositions* where the model extracts basis vectors that combine additively and correspond to the corners of a simplex surrounding the modeled data. PLSA and its extensions such as Latent Dirichlet Allocation [7] and Correlated Topic Models [8] are specific examples of Simplex Decompositions.

Since PLSA (and other PLSA extensions in the family of *topic models*) does not decompose the data-vectors themselves but the underlying multinomial distributions (i.e. the data vectors normalized to sum to unity), the extracted basis vectors don't lie in the data space but lie on the standard simplex. This can be a drawback depending on the dataset under consideration and may pose a particular poroblem if the data is aligned such that most of the variability and structure characterizing the dataset lies in directions orthogonal to the standard simplex. In such cases, the projections of the data vectors onto the simplex (which is what is decomposed by PLSA) carry very little information about the shape of the data distribution and thus the obtained PLSA bases are much less informative.

In this paper, we propose an approach to get around this drawback of PLSA (and other related topic models). We first use Singular Values Decomposition (SVD) to identify the directions of the most variability in the dataset and then transform the dataset so that these vectors are parallel to the standard simplex. We perform PLSA on the transformed data and obtain PLSA basis vectors in the transformed space. Since the transformation is affine and invertible, we apply the inverse transformation on the basis vectors to obtain basis vectors the characterize the data in the original data space. These basis vectors no longer are constrained to live on the standard simplex but lie within the data space and correspond to corners of a simplex that surrounds all the data points.

The paper is organized as follows. In Section 2, we provide the necessary background by describing the PLSA algorithm and geometry. Section 3 describes our proposed approach and constitutes the bulk of the paper. We illustrate the applicability of the method by applying the proposed technique on synthetic data. We also provide a short discussion of the algorithm and its applicability for semi-nonnegative factorizations. We conclude the paper in Section 4 with a brief summary and avenues for future work.

2 Background

Consider an $M \times N$ non-negative data matrix $\mathbf{V}$ where each column $\mathbf{v}_n$ represents the n-th data vector and v_{mn} represents the (mn)-th element. Let $\bar{\mathbf{v}}_n$ represent the normalized vector $\mathbf{v}_n$ and $\bar{\mathbf{V}}$ is the matrix $\mathbf{V}$ with all columns normalized.

PLSA characterizes the bidimensional distribution $P(m,n)$ underlying $\mathbf{V}$ as

$$P(m,n) = P(n)P(m|n) = P(n)\sum_z P(m|z)P(z|n), \tag{1}$$

where z is a latent variable. PLSA represents $\bar{\mathbf{v}}_n$ as data distributions $P(m|n)$ which in turn is expressed as a linear combination of *basis distributions* $P(m|z)$. These basis distributions combine with different proportions given by $P(z|n)$ to form data distributions.

PLSA parameters $P(m|z)$ and $P(z|n)$ can be estimated through iterations of the following equations derived using the EM algorithm,

$$\begin{aligned} P(z|m,n) &= \frac{P(m|z)P(z|n)}{\sum_z P(m|z)P(z|n)}, \\ P(m|z) &= \frac{\sum_n v_{mn}P(z|m,n)}{\sum_m \sum_n v_{mn}P(z|m,n)}, \quad \text{and} \\ P(z|n) &= \frac{\sum_m v_{mn}P(z|m,n)}{\sum_m v_{mn}}. \end{aligned}$$

EM algorithm guarantees that the above updates converge to a local optimum.

PLSA can be written as a matrix factorization

$$\bar{\mathbf{V}}_{M\times N} \approx \mathbf{W}_{M\times Z}\mathbf{H}_{Z\times N} = \mathbf{P}_{M\times N}, \tag{2}$$

where $\mathbf{W}$ is the matrix of basis distributions $P(m|z)$ with column $\mathbf{w}_z$ corresponding to the z-th basis distribution, $\mathbf{H}$ is the mixture weight distriution matrix of entries $P(z|n)$ with column $\mathbf{h}_n$ corresponding to the n-th data vector, and $\mathbf{P}$ is the matrix of model approximations $P(m|n)$ with column $\mathbf{p}_n$ corresponding to the n-th data vector. See Figure 1 for an illustration of PLSA.

3 Algorithm

The previous section described PLSA algorithm and illustrated the geometry of the technique. This section presents our proposed approach. We first briefly present the motivation for our algorithm and then describe the details of the algorithm. We illustrate the algorithm by applying it on a synthetic dataset.

3.1 Motivation

As illustrated in Figure 1, the basis distributions obtained by applying PLSA on a dataset lie on the Standard simplex. The basis distributions form the corners of a *PLSA Simplex* containing not the original datapoints but the normalized datapoints instead.

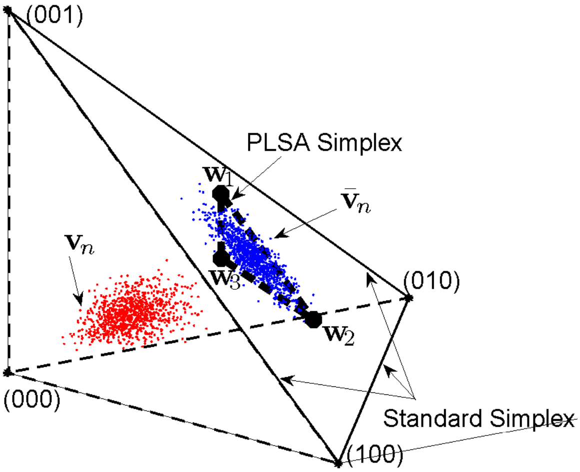


Fig. 1. Illustration of Probabilistic Latent Semantic Analysis. The data matrix $\mathbf{V}$ with 1000 3-dimensional vectors $\mathbf{v}_n$ is shown as red points and the normalized data $\bar{\mathbf{V}}$ is shown as blue points on the Standard simplex. PLSA was performed on $\mathbf{V}$ and the three extracted basis distributions shown by $\mathbf{w}_1$, $\mathbf{w}_2$ and $\mathbf{w}_3$ are points on the Standard simplex that form the corners of the PLSA simplex around normalized data points $\bar{\mathbf{v}}_n$ shown in blue.

Our goal is to extend the technique so that the basis vectors form a simplex around the original datapoints. In other words, we would like to remove the constraint that the basis vectors form multinomial distributions and thus they don't have to lie on the standard simplex. However, since we need the basis vectors to still form a simplex around the data approximations, the mixture weights with which they combine are still constrained to be multinomial distributions.

The necessity of such an approach becomes apparent when one considers the implication of normalization of datapoints that PLSA implicitly does. The normalization skews the relative geometry of datapoints. In certain cases, the normalization can hide the real shape of the distriution of datapoints as illustrated in Figure 2.

3.2 Problem Formulation

Given the data matrix $\mathbf{V}$, we would like to find a matrix decomposition similar to equation 2 of the form

$$\mathbf{V}_{M\times N} \approx \mathcal{W}_{M\times Z}\mathcal{H}_{Z\times N} = \mathcal{P}_{M\times N} \tag{3}$$

where Z is the dimensionality of the desired decomposition, $\mathcal{W}$ is the matrix of basis vectors, $\mathcal{H}$ is the matrix of mixture weights, and $\mathcal{P}$ is the matrix of approximations.

The above equation is similar to equation (2) but with important differences. In equation (2), the matrix undergoing decomposition is $\bar{\mathbf{V}}$ whereas the goal here is to decompose the original data matrix $\mathbf{V}$. The matrix $\mathcal{W}$ is analogous to $\mathbf{W}$ from equation (2) but unlike the columns of $\mathbf{W}$ that are constrained to sum to 1, the columns of $\mathcal{W}$ have no such constraints. Similarly, $\mathcal{P}$ is analogous to $\mathbf{P}$ but the columns of the former are

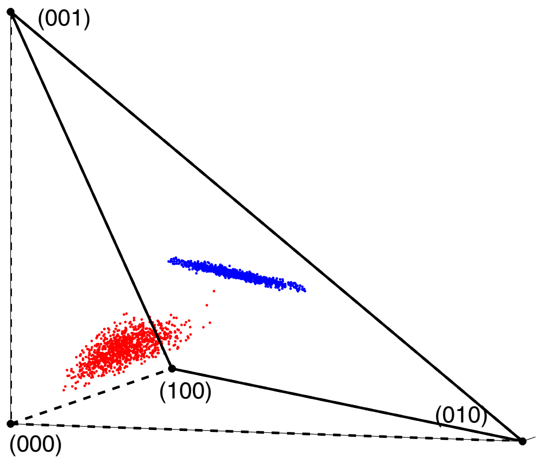


Fig. 2. Illustration of normalization on a dataset. Points in red represents a dataset of 1000 3-dimensional points where the directions of maximum variance are orthogonal to the plane corresponding to the standard simplex. Thus, the projection of points in the dataset onto the standard simplex removes important information about the distribution of datapoints.

not constrained to sum to 1 like the columns of $\mathbf{P}$. However, since both equations (2) and (3) are simplex decompositions, matrices $\boldsymbol{\mathcal{H}}$ and $\mathbf{H}$ are alike with entries in each of their columns constrained to sum to 1.

3.3 Algorithm

Consider a scenario where a dataset $\mathbf{V}$ that we desire to decompose using PLSA already lies on the standard simplex. Then, all the constraints that we need as described in the previous subsection are already satisfied. Since all data points lie on the standard simplex, the dataset $\mathbf{V}$ is identical to its normalized version $\bar{\mathbf{V}}$. Hence, the decomposition desired in equation (3) becomes identical to the decomposition in equation (2). We can apply PLSA directly to the given dataset $\mathbf{V}$ and obtain the desired basis vectors.

This observation points to the approach we present below. If we could transform the dataset so that all points lie on the standard simplex and the transformation is invertible, we can achieve the desired decomposition. However, the standard simplex in M-dimensional space represents part of the $(M-1)$-dimensional hyperplane. Thus, instead of being able to have the points exactly lie on the standard simplex, we are constrained to transforming data such that the projections of the data onto $(M-1)$ dimensions of our choice will lie on the simplex. Choosing the first $(M-1)$ principal components of the dataset as the $(M-1)$ dimensions on which data will be projected will produce the least error of all possible projections.

The problem now reduces to finding the right transformation that takes the projections of the data on the first $(M-1)$ principal components and aligns them parallel to the standard simplex. The last principal component is transformed such that it left orthogonal to the standard simplex. We leverage the work of [6] to define this transformation matrix. See the appendix for more details.

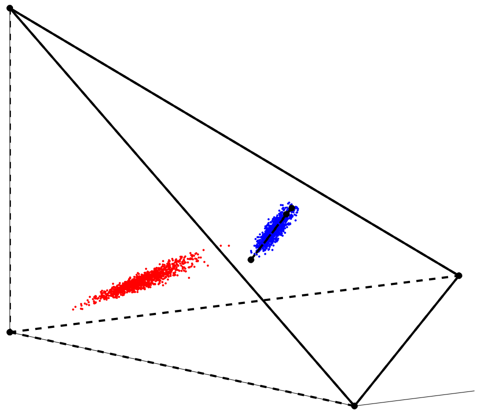


Fig. 3. Results of applying PLSA on the dataset shown in Figure 2. Since the projections of data points on the standard simplex (shown in blue) have narrow variance in one direction, the PLSA simplex obtained is degenerate and almost forms a straight line through the data projections.

Given the data matrix $\mathbf{V}_{M\times N}$, the entire algorithm can be summarized as follows:

1. Center the data by removing the mean vector to obtain $\hat{\mathbf{V}}$, i.e. $\hat{\mathbf{V}} = \mathbf{V} - mean(\mathbf{V})$.
2. Perform SVD of matrix $\hat{\mathbf{V}}^T$ to obtain $\mathbf{U}$, the matrix of data projections on the singular vectors, i.e. $\hat{\mathbf{V}}^T = \mathbf{USX}^T$.
3. Obtain the $M \times M$ transformation matrix $\mathbf{T}$ (see Appendix for details of this computation).
4. Transform the data to lie parallel to the standard simplex, i.e. $\mathbf{B} = (\mathbf{UT}^T)^T$.
5. Center the transformed data such that the centroid of the simplex coincides with the data mean, i.e. $\bar{\mathbf{B}} = \mathbf{B} - mean(\mathbf{B}) + \mathbf{c}$, where $\mathbf{c}$ is a vector corresponding to the centroid of the standard simplex.
6. Ensure all entries of $\bar{\mathbf{B}}$ are nonnegative by subtracting the minimum entry from the matrix, i.e. $\hat{\mathbf{B}} = \bar{\mathbf{B}} - min(\bar{\mathbf{B}})$.
7. Normalize the matrix $\hat{\mathbf{B}}$ such that entries of the center of the dataset sum to 1, i.e. $\mathbf{B}' = \hat{\mathbf{B}}/b$, where $b = 1 - min(\bar{\mathbf{B}})$.
8. The matrix is now ready for PLSA. Apply PLSA on $\mathbf{B}'$ to obtain $\mathbf{W}$ and $\boldsymbol{\mathcal{H}}$, i.e. $\mathbf{B}' \approx \mathbf{W}\boldsymbol{\mathcal{H}}$.
9. Undo steps 7, 6, 5 and 4 respectively for the basis vector matrix $\mathbf{W}$ to obtain $\bar{\mathbf{W}}$, i.e.
   - $\mathbf{W} = \mathbf{W} \times b$
   - $\mathbf{W} = \mathbf{W} + min(\bar{\mathbf{B}})$
   - $\mathbf{W} = \mathbf{W} + mean(\mathbf{B}) - \mathbf{c}$
   - $\bar{\mathbf{W}} = \mathbf{W}^T\mathbf{T}$
10. Undo the SVD projection and data centering for $\bar{\mathbf{W}}$ to obtain $\boldsymbol{\mathcal{W}}$, i.e.
   - $\bar{\mathbf{W}} = (\bar{\mathbf{W}}\mathbf{SX}^T)^T$
   - $\boldsymbol{\mathcal{W}} = \bar{\mathbf{W}} + mean(\mathbf{V})$

The desired decomposition is given by $\mathbf{V} \approx \boldsymbol{\mathcal{W}}\boldsymbol{\mathcal{H}}$.

For experiments, we created a synthetic dataset of 1000 3-dimensional points as illustrated in Figure 2. The dataset was created in such a way that the directions of maximal

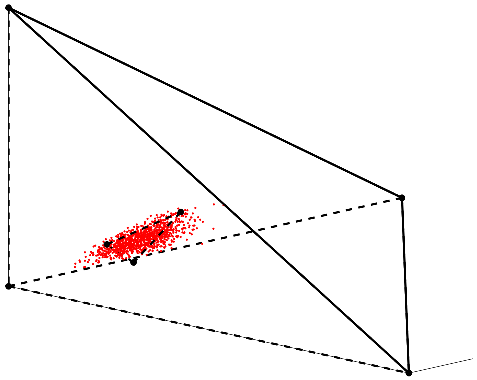


Fig. 4. Result of applying our approach on the dataset illlustrated in Figure 2. As desired, the extracted basis vectors form a simplex (dotted black line around the red points) around the original datapoints instead of around the data projections on the standard simplex.

variance present in the data was orthogonal to the plane of the standard simplex. Results of applying PLSA on the dataset is summarized in Figure 3 and results of applying the proposed approach is illustrated in Figure 4. We also created a second synthetic dataset with the presence of two distinct clusters. As shown in Figure 5, the cluster structure is lost when the data is projected on to the standard simplex. Also shown in figure are the PLSA-simplex that surrounds the projected points and the simplex resulting from Simplex Decomposition that surrounds the data.

3.4 Discussion

We first point out that even though we have used PLSA as the specific example, the proposed approach is applicable to any topic modeling technique such as Latent Dirichlet Allocation or Correlated Topic Models where data distributions are expressed as linear combinations of charateristic basis distributions.

Complexity. At its core, the proposed algorithm utilized Singular Values Decomposition and a chosen topic model such as PLSA. The data is pre-processed by performing SVD and aligning the resulting SVD directions with the Standard Simplex. Features from the topic model are computed on data that has been transformed, and then are processed back to the original data space. SVD, along with the specific implementation of the topic models, act as the primary complexity bottlenecks. Thus, the complexity of the proposed approach depends on the complexity of the algorithm chosen for implementing the topic model.

Relation to Semi-nonnegative Factorization. In the approach described in the previous subsections, no explicit constraints were placed as to the nonnegativity of the entries of basis vectors. So far in this paper, we have focused on data that have nonnegative entries but the proposed approach is also applicable for datasets with real-valued entries. The algorithm described earlier can be applied to any arbitrary datasets with real-values

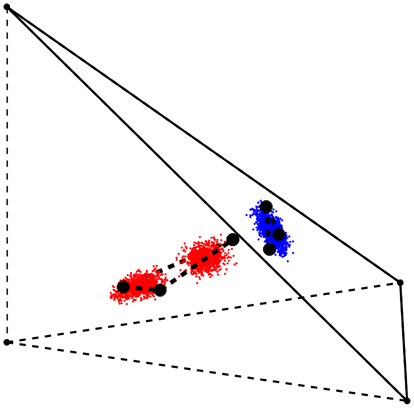


Fig. 5. Results of applying PLSA and Simplex Decomposition on a synthetic dataset with two clusters. The PLSA features form a simplex around the projected points where the cluster information has been lost. Simplex Decomposition results in features that surround both the clusters in the dataset.

entries without any modifications. In other words, the algorithm can be applied to arbitrary real-valued datasets to obtain real-valued features such that the mixture weights represent multinomial distributions (non-negative and sum to 1). This enables one to transform arbitrary datasets and represent them as points on a standard simplex.

This is an alternative approach to the one proposed by Shashanka [6]. In that work, data is transformed into the next higher dimension so that PLSA can be applied while in this work, we use SVD to align the dataset along the dimensions of the standard simplex. It will be instructive to compare the two approaches in this context and we leave that for future work.

The authors are not aware of any other work in the topic modeling community to extend their techniques for handling real-valued datasets. However, Nonnegative Matrix Factorization has been extended to generalized datasets. Specifically, Ding et al. [9] proposed a technique called Semi-Nonnegative Matrix Factorization (Semi-NMF) that decomposes a real-valued matrix into a product of a real-valued matrix and a non-negative matrix. It can be represented as $\mathbf{X}_{\pm} \approx \mathbf{F}_{\pm}\mathbf{G}_{+}^{T}$ where the subscripts indicate the signs of entries allowed in the matrices. This is similar to simplex decomposition expressed as a matrix factorization as shown in equation (3). Since matrices $\mathbf{V}$ and $\boldsymbol{\mathcal{W}}$ can have any real-valued entries whereas matrix $\boldsymbol{\mathcal{H}}$ is constrained to have only non-negative entries, the proposed simplex decomposition also qualifies as Semi-NMF. However, in Semi-NMF as proposed in Ding et al. [9], the non-negative matrix has no additional constraints and thus the method is not a simplex decomposition. The extracted features cannot be interpreted geometrically as corners of a convex-hull surrounding the dataset.

Our approach is more general in nature. As we pointed out earlier, we chose PLSA as an example for exposition in this paper but the any other topic model can be implemented as part of the proposed approach. Specifically, one can impose prior distributions on the mixture weights in our approach. Also, any new adances in topic model implementations can be incorporated into the proposed algorithm.

4 Conclusions

In this paper, we presented a novel approach to perform Simplex Decompositions on datasets. Specifically, the approach learns a set of basis vectors such that each data vector can be expressed as a linear combination of the learned set of bases and where the corresponding mixture weights are nonnegative and sum to 1. PLSA performs a similar decomposition but it characterizes the normalized datapoints instead of the original dataset itself. We demonstrated the spurious effect such a normalization can have with the help of synthetic datasets. We described our approach and demonstrated that it provides a way to overcome this drawback. We showed that the proposed algorithm is applicable for semi-nonnegative matrix factorizations. This work has several other potential applications in tasks such as clustering, feature extraction, and classification. We would like to continue this work by applying the technique on real-world problems and demonstrating its usefulness. We also intend to extend this work to be applicable to other related latent variable methods such as Probabilistic Latent Component Analysis.

References

1. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature 401 (1999)
2. Lee, D., Seung, H.: Algorithms for Non-negative Matrix Factorization. In: NIPS (2001)
3. Hofmann, T.: Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning 42 (2001)
4. Gaussier, E., Goutte, C.: Relation between PLSA and NMF and Implications. In: Proc. ACM SIGIR Conf. on Research and Dev. in Information Retrieval, pp. 601–602 (2005)
5. Smaragdis, P., Raj, B.: Shift-Invariant Probabilistic Latent Component Analysis. Journal of Machine Learning Research (2007) (submitted)
6. Shashanka, M.: Simplex Decompositions for Real-Valued Datasets. In: Proc. Intl. Workshop on Machine Learning and Signal Processing (2009)
7. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet Allocation. Jrnl of Machine Learning Res. 3 (2003)
8. Blei, D., Lafferty, J.: Correlated Topic Models. In: NIPS (2006)
9. Ding, C., Li, T., Jordan, M.: Convex and semi-nonnegative matrix factorizations. Technical Report 60428, Lawrence Berkely National Laboratory (2006)

Appendix

Here, we briefly describe how to choose the transformation matrix $\mathbf{T}$ that transforms M-dimensional data $\mathbf{V}$ such that the first $(M-1)$ principal components lie parallel to the standard $(M-1)$-Simplex. We need to indentify a set of $(M-1)$ M-dimensional orthonormal vectors that span the standard $(M-1)$-simplex.

Shashanka [6] developed a procedure to find exactly such a matrix and the method is based on induction. Let $\mathbf{R}_M$ denote a $M \times (M-1)$ matrix of $(M-1)$ orthogonal vectors. Let $\mathbf{1}_M$ and $\mathbf{0}_M$ denote M-vectors where all the entries are 1's and 0's respectively. Similarly, let $\mathbf{1}_{a\times b}$ and $\mathbf{0}_{a\times b}$ denote $a \times b$ matrices of all 1's and 0's respectively. They showed that the matrix $\mathbf{R}_{(M+1)}$ given by

$$\begin{bmatrix} \mathbf{R}_M & \mathbf{1}_M \\ \mathbf{0}^T_{(M-1)} & -M \end{bmatrix} \quad \text{if } M \text{ is even, and}$$

$$\begin{bmatrix} \mathbf{R}_{(M+1)/2} & \mathbf{0}_{(M+1)/2\times(M-1)/2} & \mathbf{1}_{(M+1)/2} \\ \mathbf{0}_{(M+1)/2\times(M-1)/2} & \mathbf{R}_{(M+1)/2} & -\mathbf{1}_{(M+1)/2} \end{bmatrix},$$

if M is odd, is orthogonal. $\mathbf{R}_{(M+1)}$ is then normalized to obtain an orthonormal matrix.

Given the above relation and the fact that $\mathbf{R}_1$ is an empty matrix, one can compute $\mathbf{R}_M$ inductively for any value of M.

We have an additional constraint that the last principal component be orthogonal to the standard simplex and this can be easily achieved by appending a column vector of 1's to $\mathbf{R}_M$.

Thus, the matrix $\mathbf{T}$ defining our desired transformation is given by $[\mathbf{R}_M \quad \mathbf{1}_M]$.

On the Understanding of the Stream Volume Behavior on Twitter

Juan Zamora[1]*, Miguel Guevara[1,2], Gabriela Dominguez[1], Héctor Allende[1], Alejandro Veloz[3], and Rodrigo Salas[3]

[1] Universidad Técnica Federico Santa María, Departamento de Informática, Valparaíso, Chile
[2] Universidad de Playa Ancha, Departamento de Computación e Informática, Valparaíso, Chile
[3] Universidad de Valparaíso, Departamento de Ingeniería Biomédica, Valparaíso, Chile
jzamora@inf.utfsm.cl, gabriela.dominguez@postgrado.usm.cl,
hallende@inf.utfsm.cl, miguel.guevara@upla.cl,
{alejandro.veloz,rodrigo.salas}@uv.cl

Abstract. Twitter has become the most widely used microblogging service nowadays, where people tells and spread, with short messages, what are they feeling or what it is happening at that moment. For this reason, having an insight of the behavior of the messages stream inside the social network could be of great help to support difficult challenges such as event detection, credibility analysis, and marketing, among others problems in social network analysis. A massive amount of data is generated in this context, and a simple idea that might be useful for every challenging mining task consists of predicting the amount of messages (*stream volume*) that will be emitted in some specific time span.

In this work we model the messages' stream volume as a time series by counting the number of collected messages during a time interval. Moreover, computational intelligence techniques are applied to identify the most influential regressors or lags, and a nonlinear autoregressive model is adjusted to this time series. Simulation experiments were conducted for a sample of over 900K collected tweets in a specific geographic area. With this methodology, an attempt to answer some questions about the behavior of the stream volume will be made.

Keywords: Social network analysis, Stream volume prediction, Time series forecasting, Non-linear autoregressive models, Computational intelligence.

1 Introduction

Twitter has become one of the main communication medias on the Internet. Users employ this media to share ideas, news or simply feelings about anything, producing in this way a valuable footprint about what is happening at every second and what people think or feel about it. Nowadays Twitter has become one of the most popular microblogging services, having hundreds of millions of messages posted everyday by more than 50M of users. In Twitter, users post short messages with a 140 characters length at most - which are called tweets - commenting about their thoughts, feelings, recent actions or

* This work was supported by the research grants FONDEF-CONICYT D09I1185, FONDECYT 1110854, DIUV 37/2008 and DGIP-UTFSM.

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 171–180.
DOI: 10.1007/978-3-642-36530-0_14

even discussions about recent news. Every posted message has associated its creation timestamp, the message content, some user information and also georeferred information if there is any.

The massive information content generated in Twitter has become an interesting source of research and application. Briefly, some areas and works where the reader could find a more detailed insight are: *Event detection* [13,15,18], *Credibility Analysis* [5,16] and *Marketing in Social Networks* [4,7]. The high frequency generation of the data have important challenges in terms of storage and processing, given the inherent online characteristic of this context. It is in this sense that the stream volume would be a useful information for every task of the aforementioned, specially those tasks that involve important computation processes executed in an online fashion.

Time series analysis over streaming data probably dates back to [6], where the authors propose the sliding window model for computation in data streams and also tackle the problem of maintaining aggregates statistics of a data stream using this approach. Several works have follow this path for computing aggregate estimates over streaming data [11,14,17,21], although at our knowledge no one pointed out to twitter or social network data. Spite of the simplicity of the idea, we think that the stream volume prediction using non-linear models without considering expensive computations such as fourier or wavelet synopsis, text or network analysis in data, may fit quite well in the streaming environment.

Being based upon the idea studied in [8], and also improving the prediction performance, this work is devoted to pose some questions about the potential periodicity of stream volume on Twitter, considering for this matter almost 1 million observations and a novel lag identification technique for autoregressive models [20]. For this purpose, linear and nonlinear methods are employed for the prediction task together with the identified autoregressive structure. After the experimentation with stream volume data aggregated by different time intervals (1, 5, 10, 15 and 60 minutes), the achieved results together with the identified lags will provide interesting evidence that will enable the discussion about the feasibility of the prediction and the interpretation of periodic patterns found in the data for each time granularity.

This work is organized as follows. In next section we deliver the fundamental concepts related to lag identification with the SIFAR algorithm, following with non-linear time series forecasting with artificial neural networks. In section 3 we explain the proposed methodology carried out for the lag identification and for the prediction task. In section 4 we show the attained results by both prediction models and then, the discussion of results is presented before ending with some concluding remarks and future work in the last section.

2 Methodology

2.1 Non-linear Time Series Prediction with Artificial Neural Networks

The statistical approach to forecasting involves the construction of stochastic models to predict the value of an observation x_t using previous observations. This is often accomplished by using linear stochastic difference equation models. By far, the most important class of such models is the linear autoregressive integrate moving average (ARIMA) model.

An important class of Non-linear Time Series models is that of non-linear Autoregressive models (NAR) which is a generalization of the linear autoregressive (AR) model to the non-linear case. A NAR model obeys the equation $x_t = h(x_{t-1}, x_{t-2}, ...,$ $x_{t-p}) + \varepsilon_t$, where h is an unknown smooth non-linear function and ε_t is white noise, and it is assumed that $E[\varepsilon_t | x_{t-1}, x_{t-2}, ...] = 0$. In this case the conditional mean predictor based on the infinite past observation is $\hat{x}_t = E[h(x_{t-1}, x_{t-2}, ...)|x_{t-1}, x_{t-2}, ...]$, with the following initial conditions $\hat{x}_0 = \hat{x}_{-1} = ... = 0$.

On the other hand, Artificial Neural Networks (ANN) have received a great deal of attention in many fields of engineering and science. Inspired by the study of brain architecture, ANN represent a class of non-linear models capable of learning from data. The essential features of an ANN are the basic processing elements referred to as neurons or nodes; the network architecture describing the connections between nodes; and the training algorithm used to estimate values of the network parameters.

Artificial Neural Networks (ANN) are seen by researches as either highly parameterized models or semiparametric structures. ANN can be considered as hypotheses of the parametric form $h(\cdot; \boldsymbol{w})$, where the hypothesis h is indexed by the parameter $\boldsymbol{w}$. The learning process consists in estimating the value of the vector of parameters $\boldsymbol{w}$ in order to adapt the learner h to perform a particular task.

The Multilayer Perceptron (MLP) is the most popular and widely known artificial neural network. In this network, the information is propagated in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes. Figure 1 illustrates the architecture of this artificial neural network with one hidden layer. Furthermore, this model has been deeply studied and several of its properties have been analyzed. One of the most important theorem is about its universal approximation capability (see [12] for details), and this theorem states that *every bounded continuous function with bounded support can be approximated arbitrarily closely by a multi-layer perceptron by selecting enough but a finite number of hidden neurons with appropriate transfer function.*

The non-linear function $h(\boldsymbol{x}, \boldsymbol{w})$ represents the output of the multilayer perceptron, where $\boldsymbol{x}$ is the input signal and $\boldsymbol{w}$ being its parameter vector. For a three layer MLP (one hidden layer), the output computation is given by the following equation

$$g(\boldsymbol{x}, \boldsymbol{w}) = f_2 \left(\sum_{j=1}^{\lambda} w_{kj}^{[2]} f_1 \left(\sum_{i=1}^{d} w_{ji}^{[1]} x_i + w_{j0}^{[1]} \right) + w_{k0}^{[2]} \right) \quad (1)$$

where λ is the number of hidden neurons, $\boldsymbol{x} = (x_1, ..., x_d)$ is the input sample point, and $\boldsymbol{w} = (w_{ji}^{[1]}, w_{j0}^{[1]}, w_{kj}^{[2]}, w_{k0}^{[2]})_{i=1..d, j=1..\lambda}$ is the vector of weights. An important factor in the specification of neural models is the choice of the transfer function, these can be any non-linear function as long as they are continuous, bounded and differentiable. The transfer function of the hidden neurons $f_1(\cdot)$ should be nonlinear while for the output neurons the function $f_2(\cdot)$ could be a linear or nonlinear function.

The MLP learns the mapping between the input space χ to the output space Υ by adjusting the connection strengths between the neurons $\boldsymbol{w}$ called weights. Several

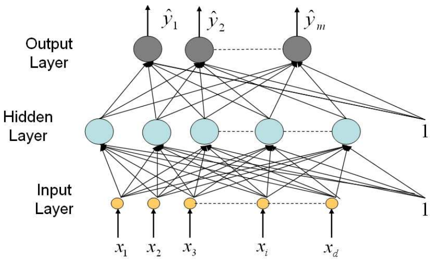


Fig. 1. Network architecture of the Multilayer Perceptron

techniques have been created to estimate the weights, where the most popular is the backpropagation learning algorithm, also known as generalized delta rule, popularized by [19].

Artificial Neural Networks have been successfully applied for time series prediction outperforming classical linear models in nonlinear processes. Refer to the following works for further details [2,3,9]

2.2 Lags Identification of an Autoregressive Time Series

Veloz et al [20] have recently proposed the *SIFAR* technique for the identification of a non-linear process represented by a time series. SIFAR is the acronym of self-identification of lags of an autoregressive TSK-based model, and the aim of the SIFAR method is the identification of the structure of a Takagi- Sugeno-Kang (TSK) fuzzy model to represent the non-linear autoregressive (NAR) relationship $x_t = f(x_{t-1}, ...,$ $x_{t-q})$ by means of a set of n fuzzy if-then rules. However, in this work we will applied the proposed strategy to identified the most relevant lags as explained below.

SIFAR [20] is a clustering-based and model-free method to determine the most suitable lags from a set of candidates of an autoregressive process. Local domains in the regressors space (product space of lags) associated to a region of the target space are generated. The consistence of the mapping is evaluated and the lags that do not contribute to the smoothness of the mapping will be neglected. The identification process consists in the following two stages:

Fuzzy Partition of the Regressors and Target Spaces. In order to establish local linear domains in the regressors and target spaces, clusters of data samples are generated using the fuzzy C-means (FCM) technique [1]. The FCM is first applied to the target space and for each of the obtained clusters, an α-cut that contains the most representative and similar data samples is computed. These α-cuts are defined by

$$O_\alpha^{(h)} = \{y_k \in \mathbf{Y} | \mu_h(y_k) \geq \alpha_o\}$$

where $\alpha_o \in [0, 1]$ is a user-defined threshold. Afterwards, the FCM is applied on the regressors space independently for each set of explanatory vectors whose target values are in $\{O_\alpha^{(1)}, \ldots, O_\alpha^{(n_o)}\}$. In other words, for the h-th α-cut $O_\alpha^{(i)}$, the FCM is applied to the dataset $\{\mathbf{x}_{k^*}\}$, where $k^* = \{k | y_k \in O_\alpha^{(h)}\}$.

Lags Relevance Evaluation. The contribution of each lag variable of the autoregressive process to affect the regularity of the local mapping between regressors and target spaces is quantified according to the following formula:

$$R_h^j = |k_{in}^{(h)} - k_o^{(h)}|,$$

where $k_o^{(h)} = \{k|y_k \in O_\alpha^{(h)}\}$, $k_{in}^{(h)} = \{k|\mathbf{x}_k \in I_\alpha^{(h)}\}$, $I_\alpha^{(h)} = \bigcup_{i=1}^{n_{in}}\{\mathbf{x}_k \in \mathbf{X}|y_k \in O_\alpha^{(h)} \wedge w_i^{(h)}(\mathbf{x}_k) \geq \alpha_{in}\}$, $|\cdot|$ represents the cardinality of the resulting set and $w_i^{(h)}$ is the conjunction of unidimensional fuzzy sets associated to the i-th cluster in the regressors space, i.e., $w_i^{(h)}(\mathbf{x}_k) = \prod_{j=1}^{d} \mu_{ij}^{(h)}(x_{kj})$, where $\mu_{ij}^{(h)}(\cdot)$ is the membership function obtained by the FCM algorithm . Finally, the total relevance of the j-th lag is selected as the maximum of the terms R_h^j, i.e., $R_j = \max_{h \in \{1,\ldots,n_o\}} R_h^j$. Afterwards, the set relevances $\{R_1, \ldots, R_j, \ldots, R_d\}$ for the set of candidate lags are sorted in descending order. The lags that are going to be considered for the incorporation to the model are the q-th elements with highest value, where q is a user-defined parameter.

2.3 Twitter Time Series

In this work, the volume prediction task is attained by using a collection of Tweets pulled from the Twitter stream and filtered by the geographical region of Chile, during the period of time covered between August 13th and September 9th of 2011. The amount of data consists of $947,927$ tweets collected in the aforementioned time span of 25 days (about $36,000$ minutes). In a previous work [8], we have posed the significance of the stream volume prediction task over Twitter, and also a preliminary attempt in this matter was made in order to test the feasibility of the idea. For a detailed information about the extraction and pre-processing procedures, the following references will be quite explanatory [8,10].

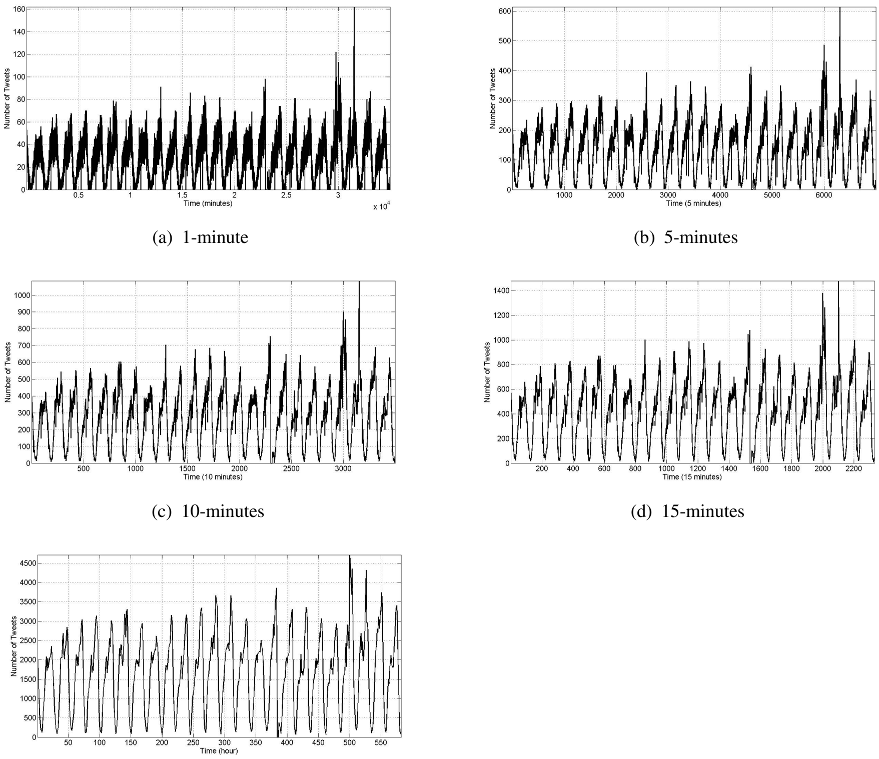
The stream volume is modeled as a time series by counting the number of collected messages during a user-defined time interval, namely 1-5-10-15-60 minutes, and their plots are depicted in figure 2 .

2.4 Time Series Processing

The Twitter Time Series is analyzed with the computational intelligence techniques described above, in order to identify the best non-linear structure able to have a good forecasting performance.

Lag Identification Task

In this stage, the most significant lags are identified by means of the SIFAR strategy described in section 2.2. The SIFAR strategy is problem dependent, for this reason, an exhaustive combination of model parameters are tested in order to find the best set of lags. For example, the maximum order of the possible regressor is a user defined parameter. In this way, several candidate combinations of lags were generated and evaluated, the one with the lowest mean square error is selected, and the lag identification task ends.



(a) 1-minute

(b) 5-minutes

(c) 10-minutes

(d) 15-minutes

(e) 60-minutes

Fig. 2. Aggregated time series of the Chilean tweet stream sampled between August 13th and September 9th of 2011. The selected window sizes or *granularity levels* are (a) 1 minute, (b) 5 minutes and (c) 10 minutes (d) 15 minutes (e) 60 minutes.

Stream Volume Prediction Task

In order to forecast the quantity of tweets collected in a specific time window, the configurations obtained in the previous stage were employed to generate the training data sets with the autoregressive structure. A classical linear autoregressive (AR) model and an three layer MLP were estimated from these data sets. Due to the random initialization of the MLP model, 10 independent runs were executed for each dataset. Finally the average and standard deviation of the performance measures are obtained.

The architecture of the neural network consists in three layers of neurons (one hidden layer), where the number of input neurons depends on the number of lags of each selected configuration, the number of output neurons was set to one (1-step ahead prediction), and we arbitrarily decided to test with ten hidden neurons to maintain a low complexity of the model. We selected the *log sigmoid* transfer function

$$f_1(z) = \frac{1}{1+e^{-z}} \ ,$$

for the hidden neurons and the linear transfer function, $f_2(z) = z$, for the output neurons. The parameters were estimated according to Levenberg-Marquardt optimization algorithm (for this study we have used the Neural Nework toolbox of Matlab).

Performance Measures

In order to compare the performance of the prediction algorithms employed in this work, the Mean Square Error (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

and the Correlation Coefficient (R)

$$R = \frac{\sum_{i=1}^{n}(y_i - \bar{Y})(\hat{y}_i - \bar{P})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{Y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{P})^2}}$$

were computed. In the previous equations, n is the number of samples; y_i, $i = 1..n$, are the targets, while $\hat{y}_i$ are the predictions; $\bar{Y}$ and $\bar{P}$ are the mean values of the targets and predictions respectively.

3 Results

3.1 Lag Identification Task

The resulting configurations of lags are presented in table 1. For the 1-minute aggregation data, the SIFAR model identified five lags from the six past minutes (excluding the minute number 5), denoting a short term dependency. In the 5-minutes data, a long term dependency was identified as the selected lags correspond to the history of almost *16 hours* before. Following with the 10-minutes aggregation data, the five lags identified correspond to the history of between *16 and 17 hours* before. For the 15-minutes data, the identified lags correspond to the history of between *16 and 20 hours* before. Finally, for the 1-hour (60-minutes) data, the dependency found corresponds to what happened between *6 and 8 hours* before.

3.2 Stream Volume Prediction Task

Table 2 shows that the AR and the MLP models attain comparable and acceptable results. Only with the exception of the configuration $05 - C2$ for the MLP, the achieved R values are higher than 90% for both models with a quite low variability, indicating that an important part of the variance in the target variable is well explained by both models. On the other hand, the great variability of the MSE for the MLP model may suggest the existence of some outliers.

Finally, an interesting situation occurs in the results of the $05 - C2$ configuration (5-minute data with configuration C2), where MSE and R are considerably worse than the attained values by the AR model, suggesting a notorious difficulty suffered by the MLP on predicting with this configuration of lags. The SIFAR method is based in the linear relation between the input and output local regions, this strategy may affect the capabilities of the MLP of modeling global non-linear interactions.

Table 1. Configurations identified by the SIFAR algorithm for the autoregressive structure of the time series data, for each dataset

Configuration	Lags for 1 minute
C1	$x_{t-1}, x_{t-2}, x_{t-3}, x_{t-4}, x_{t-6}$
Configuration	Lags for 5 minutes
C2	x_{t-191}, x_{t-192}
Configuration	Lags for 10 minutes
C3	$x_{t-97}, x_{t-98}, x_{t-99}, x_{t-100}, x_{t-101}$
Configuration	Lags for 15 minutes
C4	$x_{t-65}, x_{t-66}, x_{t-67}, x_{t-68}, , x_{t-69}$
Configuration	Lags for 1 hour
C5	$x_{t-6}, x_{t-7}, x_{t-8}$

Table 2. Results attained by both prediction algorithms and for each of the autoregressive configurations identified. For the ANN model, as MSE and R are average values over several runs, they are presented together with its associated standard deviation. Each minute-granularity and configuracion-number pair is denoted in the first column.

	AR		MLP	
#mins-conf.	MSE	R	MSE	R
01-C1	0.001975	0.939540	0.001969 (0.000330)	0.922890 (0.013704)
05-C2	0.001607	0.972377	0.015622 (0.022797)	0.769144 (0.274799)
10-C3	0.002388	0.967639	0.004061 (0.003163)	0.927717 (0.056275)
15-C4	0.003357	0.962108	0.003590 (0.001080)	0.948385 (0.014190)
60-C5	0.012081	0.907956	0.009609 (0.002403)	0.908213 (0.020350)

3.3 Discussion

Considering the results presented in the previous section, the feasibility of stream-volume-predicton idea is reinforced as the results seem promising in terms of performance. Moreover a new question must be posed, and it relates with the periodicity found in the stream volume, and particularly the long-term periodicity, as it is shown that the most relevant lags for 1-step ahead predictions are between 15 and 20 hours in three cases and between 6 and 8 hours in another one. Additionally, for the 1-minute time-window size, a short-term periodic behavior was observed by the exhaustive search of the SIFAR strategy.

Moreover, note the "batch behavior" of all the identified lags, which it means that the recognized configuration of lags come in a sequence fashion. This observation results quite interesting as it may suggest the existence of a kind of packed temporal structure of this data. Anyway, it must be considered the possibility that this "packed structure" may have been generated due to the internal clustering mechanism of SIFAR. Hence, in order to enhance this observation and the potential conclusions attached to it, another algorithms for structure identification of autoregressive processes must be employed. In figure 2, the plots display an interesting behavior known as self-similarity, where at different windows size, the time series pattern are quite similar.

Nevertheless, the abstraction level imposed by the aggregated data and the atomic data itself (timestamp of each tweet), presents - on one hand - the simplicity of the collection and processing as two main advantages, but - on the other hand - it narrows the range of the observations. That is, even when a periodicity in the stream volume appears - which in turn seems very interesting for the study of the phenomenon - the potential cause of this pattern remains hidden as a more thorough analysis of the text and maybe some other sources of evidence are needed. Finally, *is there a periodic behavior in the Stream Volume on Twitter?*. The evidence supports it, but experimentation with more data sets would allow a much stronger conclusion. However, the regular behavior of the the data stream could be of great help in finding odd samples that could be of paramount importance for event detection or other social network applications.

4 Conclusions

In this paper we present an extension of the work [8] that face the problem of stream volume prediction on Twitter. As an enhancement of our previous work, we have used an automatic algorithm - SIFAR - to identify the most relevant lags in an autoregressive process. We also have extended the size of the dataset from $171,991$ to $947,927$ tweets. After this, and using the identified lags, we have compared a linear and a non-linear autoregressive prediction techniques, namely a classic AR model and Multilayer Perceptron model. By means of these improvements, in the present work we are able to observe and identify a periodic behavior in the stream volume and also attain a reasonable performance in the predicion task.

Anyway, a little step on unraveling the behavior of the stream volume in Twitter was made. As interesting observations arise, several questions also appear, which off course will make up the future work in this subject. Is this periodic behavior a constant pattern?; Is it related with some special incident or event ocurring during the analyzed period?; Is there a packed structure in the identified lags group?

In addition, a next step in this problem would be to consider the real-time prediction together with the potential evolution of the data.

References

1. Abonyi, J., Feil, B.: Cluster analysis for data mining and system identification. Birkhäuser Verlag AG (2007)
2. Allende, H., Moraga, C., Salas, R.: Artificial Neural Networks in Time Series Forescasting: A Comparative Analysis. Kybernetika 38(6), 685–707 (2002)
3. Balestrassi, P., Popova, E., Paiva, A., Marangon-Lima, J.: Design of experiments on neural network's training for nonlinear time series forecasting. Neurocomputing 72, 1160–1178 (2009)
4. Banerjee, S., Al-Qaheri, H., Hassanien, A.E.: Mining Social Networks for Viral Marketing using Fuzzy Logic. In: 4th Asia International Conference on Mathematical/Analytical Modelling, Computer Simulation (AMS), pp. 24–28 (2010)
5. Castillo, C., Mendoza, M., Poblete, B.: Information Credibility on Twitter. In: Proceedings of the 20th international conference on World Wide Web (WWW 2011), pp. 675–684. ACM, New York (2011)

6. Datar, M., Gionis, A., Indyk, P., Motwani, R.: Maintaining Stream Statistics over Sliding Windows (extended abstract). In: Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2002), pp. 635–644. SIAM, Philadelphia (2002)
7. Domingos, P.: Mining Social Networks for Viral Marketing. IEEE Intelligent Systems 20(1), 80–82 (2005)
8. Dominguez, G., Zamora, J., Guevara, M., Allende, H., Salas, R.: Stream Volume Prediction in Twitter with Artificial Neural Networks. In: Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods (ICPRAM 2012), vol. 2, pp. 488–493. SciTePress, Algarve (2012)
9. Faraway, J., Chatfield, C.: Time series forecasting with neural networks: a comparative study using the airline data. Applied Statistics 47(2), 231–250 (1998)
10. Guevara, M. , Zamora, J. , Salas, R.: Collecting and Processing Tweets. UTFSM Research Report (available upon request) (2011)
11. Guha, S., Koudas, N., Shim, K.: Approximation, Streaming Algorithms for Histogram Construction Problems. ACM Transaction on Database Systems 31, 396–438 (2006)
12. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feedforward Networks are Universal Approximators. Journal of Neural Networks 2, 359–366 (1989)
13. Lee, C.-H., Wu, C.-H., Chien, T.-F.: Burs**T**: A Dynamic Term Weighting Scheme for Mining Microblogging Messages. In: Liu, D. (ed.) ISNN 2011, Part III. LNCS, vol. 6677, pp. 548–557. Springer, Heidelberg (2011)
14. Lee, L.K., Ting, H.F.: Maintaining Significant Stream Statistics over Sliding Windows. In: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA 2006), pp. 724–732. ACM, New York (2006)
15. Mathioudakis, M., Koudas, N.: Twittermonitor: Trend Detection over the Twitter Stream. In: Proceedings of the International Conference on Management of Data (SIGMOD 2010), pp. 1155–1158. ACM, New York (2010)
16. Mendoza, M., Poblete, B., Castillo, C.: Twitter under crisis: can we trust what we rt? In: Proceedings of the 1st Workshop on Social Media Analytics (SOMA 2010), pp. 71–79. ACM, New York (2010)
17. Pan, B., Demiryurek, U., Banaei-Kashani, F., Shahabi, C.: Spatiotemporal Summarization of Traffic Data Streams. In: Proceedings of the ACM SIGSPATIAL International Workshop on GeoStreaming (IWGS 2010), pp. 4–10. ACM, New York (2010)
18. Petrovic, S., Osborne, M., Lavrenko, V.: Streaming First Story Detection with Application to Twitter. In: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 181–189. The Association for Computational Linguistics, California (2010)
19. Rumelhart, D., Hinton, G., William, R.: Learning Internal Representation by Backpropagation Errors. Nature Journal 323, 533–536 (1986)
20. Veloz, A., Salas, R., Allende-Cid, H., Allende, H.: SIFAR: Self-Identification of Lags of an Autoregressive TSK-based Model. In: Proceedings of the 42nd International Symposium on Multiple-Valued Logic (ISMVL 2012). IEEE, Victoria (2012)
21. Zhu, Y., Shasha, D.: Statstream: Statistical Monitoring of Thousands of Data Streams in Real Time. In: Proceedings of the 28th International Conference on Very Large Data Bases (VLDB 2002), pp. 358–369. Morgan Kaufmann (2002)

Training Selection with Label Propagation for Semi-supervised Land Classification and Segmentation of Satellite Images

Olga Rajadell and Pedro García-Sevilla*

Institute of New Imaging Technologies, University Jaume I, Spain
{orajadel,pgarcia}@lsi.uji.es

Abstract. Different scenarios can be found in land classification and segmentation of satellite images. First, when prior knowledge is available, the training data is generally selected by randomly picking samples within classes. When no prior knowledge is available the system can pick samples at random among all unlabeled data, which is highly unreliable or it can rely on the expert collaboration to improve progressively the training data applying an active learning function. We suggest a scheme to tackle the lack of prior knowledge without actively involving the expert, whose collaboration may be expensive. The proposed scheme uses a clustering technique to analyze the feature space and find the most representative samples for being labeled. In this case the expert is just involved in labeling once a reliable training data set for being representative of the feature space. Once the training set is labeled by the expert, different classifiers may be built to process the rest of samples. Three different approaches are presented in this paper: the result of the clustering process, a distance based classifier, and support vector machines (SVM).

Keywords: Semi-supervised classification, Image segmentation, Hyper-spectral imaging, mode seek clustering.

1 Introduction

The classification and segmentation of land usage in satellite images generally requires an expert who provides the corresponding labels for the different areas in the images. Some authors work with prior knowledge in a supervised scenario and training data is selected within each class [1][2]. Lately the research interest in active learning techniques, which move to a semi-supervised scenario, is raising. In new real databases, the expert labeling involves whether prior knowledge or checking at the land place itself, which could be highly expensive. The expert collaboration may be needed an unknown number of steps to improve the classification by helping in the training selection until the convergence condition is achieved [3][4]. Hence, the expert collaboration can be highly expensive and picking at random among the unlabeled pool is not convenient

* This work has been partly supported by grant FPI PREDOC/2007/20 from Fundació Caixa Castelló-Bancaixa and projects CSD2007-00018 (Consolider Ingenio 2010) and AYA2008-05965-C04-04 from the Spanish Ministry of Science and Innovation.

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 181–192.
DOI: 10.1007/978-3-642-36530-0_15

because classes are often very unbalanced and the probabilities of getting an efficient representative training data is inverse to the amount of labeled samples. Consequently, decreasing the size of labeled data is a problem. Whereas for classifier based on distances, larger training sets overfit our classifier and it is preferable to provide the classifier with a few interesting highly descriptive samples [5]; for other types of classifiers providing a considerable amount of training samples is a concern.

In unsupervised scenarios, data analysis techniques have proved being good at providing relevant data when no prior knowledge is available. Among them, clustering techniques allow us to divide data in groups of similar samples. Specially when samples represent pixels from an image, clustering algorithms have successfully been applied to image segmentation in various fields and applications [6]. We aim to segment and classify hyper-spectral satellite images. Fully unsupervised procedures often have insufficient accurate classification results. For such a reason, a hybrid scenario between supervised and unsupervised techniques is our target where the methods applied could take into account some labels to build a classifier. We suggest a cluster-based training selection. This approach selects the training samples according to an unsupervised analysis of the data (mode seek clustering). The selected data (centers of the clusters) are likely to well represent those samples that were clustered together. This scheme was presented in [7] where a $KNN1$ classifier was used.

Here we also introduce label propagation to adapt the method to other classifiers. For the sake of using a SVM classifier, the unlabeled data contained in each cluster is modeled regarding the distribution of their distances to their corresponding centers. The label of the center is propagated to those samples that fit this model. Besides we also test the result of assigning labels to unlabeled samples according to the result given by the cluster itself and the labels provided by the expert for the modes of clusters. For all cases, the suggested scheme is compared with the supervised state of the art classification, resulting in outperforming previous works.

A review of the sample selection scheme with its spatial improvement is presented in Section 2. Several classification alternatives are presented in Section 3. Results will be shown and analyzed in Section 5. Finally, Section 6 presents some conclusions.

2 Preliminaries

Nowadays, due to the improvement in the sensors, databases used for segmentation and classification of hyper-spectral satellite images are highly reliable in terms of spectral and spatial resolution. Therefore, we can consider that our feature space representation of the data is also highly reliable. On the other hand, in segmentation and classification of this kind of images the training data used has not been a concerned so far, without worrying about providing the most reliable information [5]. The scheme suggested in [7] was a first attempt in this sense. It was proposed an unsupervised selection of the training samples based on the analysis of the feature space to provide a representative set of labeled data. It proceeds as follows:

1. In order to reduce the dimensionality of the problem, a set of spectral bands, given a desired number, is selected by using a band selection method. The WaLuMi band

selection method [8] was used in this case, although any other similar method could be used.

2. A clustering process is used to select the most representative samples in the image. In this case, we have used the Mode Seek clustering procedure which is applied over the reduced feature space. An improvement in the clustering process is included by adding the spatial coordinates of each pixel in the image as additional features. Since the clustering is based on distances, spatial coordinates should also be taken into account assuming the class connection principle.
3. The modes (centers of the clusters) resulting of the previous step define the training set for the next step. The expert is involved at this point, only once, by providing the corresponding labels of the selected samples.
4. The classification of the rest of non-selected samples is performed, using the training set defined above to build the classifier. Three different classification experiments have been performed here: a KNN classifier with $k = 1$, a direct classification with the results of the clustering process, and an extension will be presented for the use of SVM.

2.1 Mode Seek Clustering

Given a hyper-spectral image, all pixels can be considered as samples which are characterized by their corresponding feature vectors (spectral curve). The set of features defined is called the feature space and samples (pixels) are represented as points in that multi-dimensional space. A clustering method groups similar objects (samples) in sets that are called clusters. The similarity measure between samples is defined by the cluster algorithm used. A crucial problem lies in finding a good distance measure between the objects represented by these feature vectors. Many clustering algorithms are well known. A KNN mode seeking method will be used in this paper [9]. It selects a number of modes which is controlled by the neighborhood parameter (s). For each class object x_j , the method seeks the dissimilarity to its s^{th} neighbors. Then, for the s neighbors of x_j , the dissimilarities to their s^{th} neighbors are also computed. If the dissimilarity of x_j to its s^{th} neighbor is minimum compared to those of its s neighbors, it is selected as prototype. Note that the parameter s only influences the scheme in a way that the bigger it is the less clusters the method will get since more samples will be grouped in the same cluster, that is, less modes will be selected as a result. For further information about the mode seek clustering method see [9] and [5]

2.2 Spatial Improvement

The clustering algorithm searches for local density maxima where the density function has been calculated using the distances for each sample in its s neighborhood using a dissimilarity measure as the distance between pairs of samples. In that difference, all features (dimensions) are considered. When features do not include any spatial information the class connection principle is missed (pixels that lie near in the image are likely to belong to the same class). Therefore, we suggest to include the spatial coordinates among the feature of the samples. See Fig 1.(a) where all samples have been represented in the three first features space and in different color per class. Notice that, when

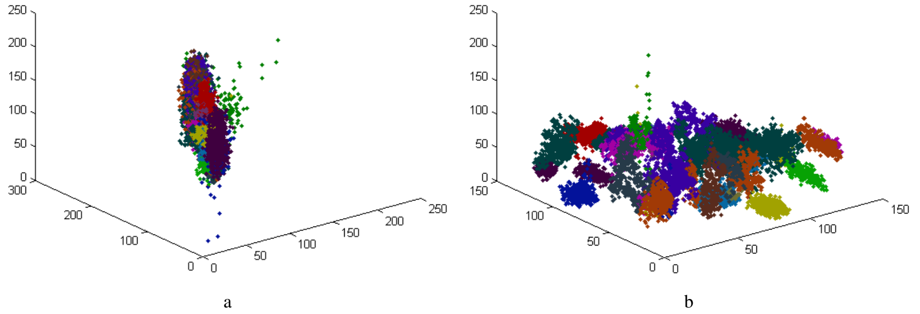


Fig. 1. Effects of including spatial information in the feature space. Plots show the samples of the database in the feature space, colored per class according to the ground-truth. (a) no spatial information is available. (b) spatial coordinates are included.

no spatial data is considered and all classes are located in the same space and when no prior knowledge is available for the clustering process, finding representatives for each class would be difficult since the classes themselves may lie together. Moreover, different areas of the same class may be within the same cloud. However, when spatial data is included, Fig 1.(b), the single cloud of samples is broken according to spatial distances and classes (fields) are more separable. In this sense also samples belonging to the same class but lying in different places of the image are separable.

In [7] it was suggested to weigh the spatial coordinates by an arbitrary number to reinforce two samples that are close spatially to have a closer distance and the way round. Such a weight should be decided in terms of the range of the features provided by the spectrometer so the coordinates are overweighed but they do not cause the rest of features be dismissed in the global measure.

3 Classification Alternatives

The whole dataset was first reduced to 10 bands using the band selection method named in Section 2. This method is used for minimizing the correlation between features but maximizing the amount of information provided, all that without changing the feature space. Clustering was carried out tuning the parameter s to get a prefixed number of selected samples. Three different classification alternatives have been used.

3.1 Straightforward Schemes

1. First a KNN with k=1 classification has been performed with the labeled samples as training set. This is not an arbitrary choice, because the clustering procedure used is based on densities calculated on a dissimilarity space, and therefore, the local maxima correspond to samples which minimize its dissimilarity with a high amount of samples around it. Thus, the selected samples are highly representative in distance-based classifiers.
2. Second, another classification process has been performed using the straightforward result of the clustering procedure. The expert labels the selected samples.

Then, all samples belonging to the cluster that each labeled sample is representing are automatically labeled in the same class. This provides a very fast pixel classification scheme as the clustering result is already available.

3.2 Extension to SVM

The scheme, as it has been presented, is not useful for classifiers that are not based on distances. However, we would like to check if providing relevant training data may be also useful for other classifiers. In this case, we extend the proposed method for SVM. For such a classifier, it would be useful to model the data shape and not their centers. Nevertheless, we do not want to increase the amount of labeled data. According to these criteria we suggest using the label of the centers as in the previous cases and using a label propagation technique to those samples fitting certain model with the aim of modeling the shape of the data and provide the SVM with a useful training set. The main idea behind label propagation is the cluster assumption. Two samples x_i and x_j have a high probability of sharing the same label y if there is a path between them in X which moves through regions of significant density [10]. Many graph-based techniques can be found in literature [11]. To propagate labels using the cluster analysis already performed and according to the main idea of label propagation, we suggest propagating the label of all cluster centers as follows:

Given the set of clusters $W = \{w_1, ..., w_t\}$
and distances $D_i = \{d_1, ..., d_s\}$
where $d_j = distance(center_{w_i}, x_j)$ and $x_j \in w_i$
we can assign the label y_{w_i} according:
(x_j, y_{w_i}) if $0.8 * max(D_i) \leq d_j \leq 0.85 * max(D_i)$

We considered the possibility of propagating the label to the whole cluster or all the data included in the sphere created taking as a limit $0.8 * max(D_i)$. There are two reasons for discarding these options. In the case first, propagating the label of the center to all data points in the cluster increases the errors introduced by label propagation since the further a data point is from its center the more possibilities that they do not share the same label, according the cluster assumption. As for both cases, we aimed to use a SVM as classifier and training is the most expensive step. Increasing considerably the training data has an undesired effect on the computation time. This is rather an arbitrary choice and we are currently working on the direction of how to better determine this parameter.

4 Data Sets

A well-known data set has been used in the experiments (see Fig 4). Hyper-spectral image 92AV3C was provided by the Airborne Visible Infrared Imaging Spectrometer (AVIRIS) and acquired over the Indian Pine Test Site in Northwestern Indiana in 1992. From the 220 bands that composed the image, 20 are usually ignored (the ones that cover the region of water absorption or with low SNR) [12]. The image has a spatial dimension of 145×145 pixels. Spatial resolution is 20m per pixel. Classes range from

20 to 2468 pixels in size. In it, three different growing states of soya can be found, together with other three different growing states of corn. Woods, pasture and trees are the bigger classes in terms of number of samples (pixels). Smaller classes are steel towers, hay-windrowed, alfalfa, drives, oats, grass and wheat. In total, the dataset has 16 labeled classes and unlabeled part which is known as the background. This so called background will be here considered as the 17 class for the segmentation experiments.

We will analyze the details and performance for AVIRIS data set since it is a widely used data set. However we will show results with two other data sets, HYMAP and also CHRISPROBA (see Fig 4).

The DAISEX99 project provides useful aerial images about the study of the variability in the reflectance of different natural surfaces. This source of data, which is referred to as HyMap, corresponds to a spectral image (700 × 670 pixels and seven classes that are composed of crops and an unknown class) acquired with the 128-band HyMap spectrometer during the DAISEX9́9 campaign (http:/io.uv.es/projects/daisex/). The last data set was acquired by the satellite PROBA which has a positional spectroradiometric system (CHRIS) that measures the spectral radiance, i.e., the amount of light that passes through or is emitted from a particular area. System CHRISPROBA is able to operate in several acquisition modes. The image used in this paper come from the mode that operates on an area of 15 × 15 km, with a spatial resolution of 34 m, obtaining a set of 62 spectral bands that range from 400 to 1050 nm (641 × 617 pixels and nine classes that are composed of crops and an unknown class). The camera has a spectral resolution of 10 nm. Concretely, this image covering the area that is known as Barrax (Albacete, Spain) has 52 bands.

5 Experimental Results

In this section we will analyze the details of the method for AVIRIS data set. Later results will be shown for the other two data sets. In Fig 3 the results obtained using several classification strategies are compared: KNN using only the center of the clusters for the training set, SVM after label propagation, KNN using the same training set used for the SVM, and the classification using the plain output of the mode seek clustering. It was already shown in [7] that the scheme used with KNN clearly outperformed the random selection. Now, the classification result for the KNN classifier adding more samples in the clusters assuming the same label is very similar to the ones obtained with the KNN classifier using only the cluster centers. The SVM classifier provided the worst results in all experiments. This may be due to the fact that the double threshold scheme proposed assumes a spherical distribution of the samples around the cluster centers. However, this is not the case in general, and that is the reason why SVM cannot properly model the borders of the classes using these training samples. On the other hand, the mode seek clustering classification outperformed all other methods. The reason is that this sort of clustering is not based on the distance to a central sample in the cluster but to the distance to other samples in the clusters. When the distance to a central point is considered, a spheric distribution of the pixels around this point is assumed. However, the mode seek clustering provides clusters that may adapt to different shapes, depending on the distribution of the samples in the feature space, and these clusters can be modeled using just one sample.

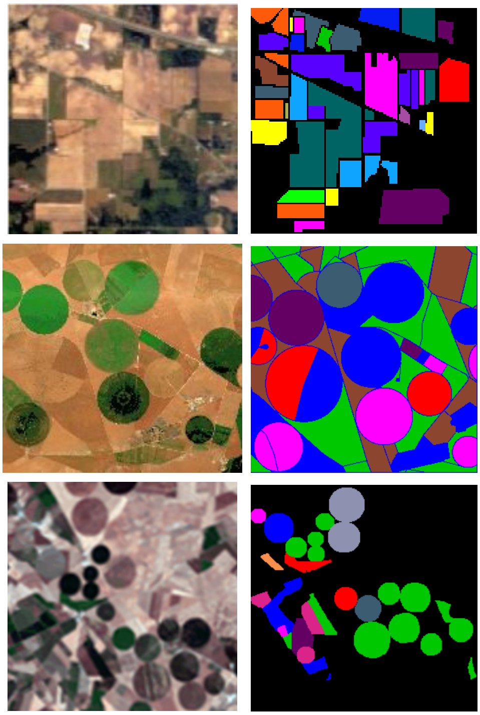


Fig. 2. AVIRIS, HYMAP and CHRIS-PROBA data sets (respectively per row). Color composition and ground-truth.

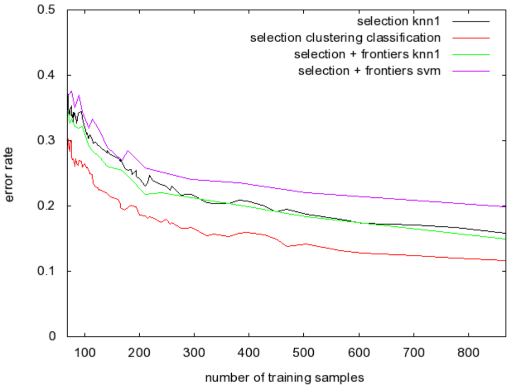
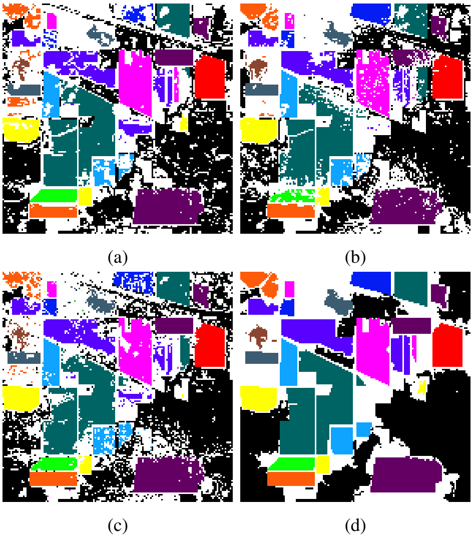


Fig. 3. Learning curve in terms of error rate when increasing the size of training data in number of samples selected by the scheme suggested. Different classification methods tested using the 92AV3C database.

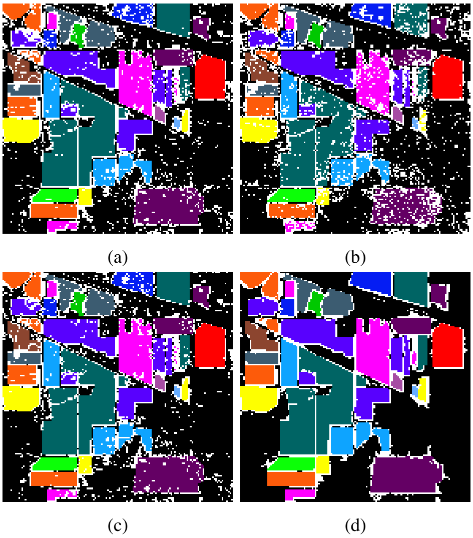
The database has 21025 samples. Fig. 4 show the classification results of several classifiers when 0.33% of the pixels in the image (69 pixels) was labeled by the expert. The classification errors are shown as white pixels. It can be noted that the clustering



(a) (b)

(c) (d)

Fig. 4. Segmentation-classification results using 0.33% of data for the selected training set using several classifiers. (a) KNN using the cluster centers. (b) SVM (c) KNN using the same training set as for the SVM (d) mode seek clustering.

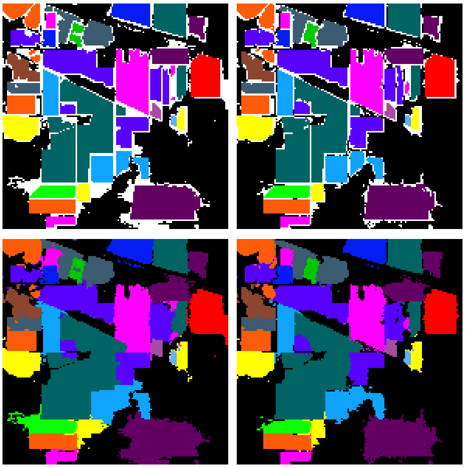


(a) (b)

(c) (d)

Fig. 5. Segmentation-classification results using 4% of data for the selected training set using several classifiers. (a) KNN using the cluster centers. (b) SVM (c) KNN using the same training set as for the SVM (d) mode seek clustering.

classifier outperformed the other classifiers not only in the percentage of classification rate but also providing smooth compact regions in the image. Similar results can be seen in Fig. 5 where 4% of the pixels in the image was labeled, where the classification



$error = 0.157$ $error = 0.116$

(a) (b)

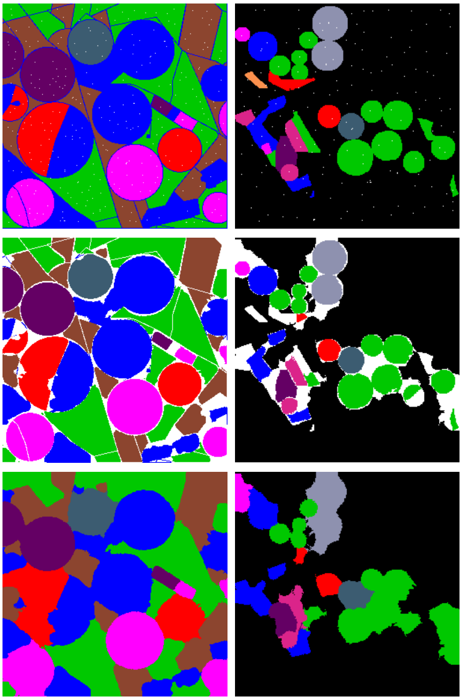
Fig. 6. Segmentation-classification results using different amounts of data for the selected training set using the proposed scheme and the clustering based classification. (a) Using 2% of the data. (b) Using 4% of the data.

errors tend to concentrate in the borders of the different regions in the image. Note that the segmentation results are quite smooth even for the background class.

Let's consider the 2% of the samples and the cluster-based classification. See results in Fig 6.(a). Observe the top left part of the image where the selection manages to detect all of them although the classes are lying one next to each other and their size is not big. The best result is presented in Fig 6.(b), it is the classification-segmentation result for the 17-classes problem using 4% of the data. The overall error rate is 0.116 and the most relevant error is the lost of very small classes that cannot be found by the clustering. In Table 1 the results per class are presented for different sizes of the training set using cluster classification. Observe that the accuracy per class of a reduced training set is good when the class has been detected by the cluster. As long as one class is missed in the selection of the training data, this class will be entirely misclassified.

A brief overview of the results for the other two data sets can be found in Fig 7. This data sets have higher spatial resolution and better results were expected for them. Indeed, error rates of 0.1 are reached for both when less than 0.5% of the data is used for training. In this cases, all classes are big enough in number of samples and there are no classes missed in the selection process. Again, errors are placed at the borders of the areas. Note that in HYMAP data set there is an area defined in the groundtruth that draws a line around all visible shapes and it is labeled. This area is too narrow and always confused with the adjacent classes, for such an example of class distribution this method will have difficulties since their samples are spatially very close to other areas and they never form a structure big enough to be detected by itself.

In Table 1 where the error rate per class is shown, we can see that the results obtained using 2% of the samples are already comparable in terms of per class accuracy with



$error = 0.129$ $error = 0.109$

(a) (b)

Fig. 7. Segmentation-classification results for other data sets selecting the training set using the proposed scheme and the clustering based classification. The training set selected is shown at the first row, at the second row the error resulting is presented in white and last row shows the classification result for (a) Using 0.312% of the data set HYMAP. (b) Using 0.244% of the data set CHRIS-PROBA.

results obtained in supervised scenarios using 5% of the data [1]. Notice that classes with only one spatial area are well classified with few samples needed, such as Alfalfa, Wheat, Hay-windrowed, Grass/pasture-mowed and Corn. Some of them (as Wheat and Hay-windrowed) were already well classified when only 0.33% training data was used. The rest of the classes are divided in different spatial areas and their detection is highly dependant on the size of the area and the amount of different classes that surrounds them. Soybeans-min-till class is from the beginning well classified with only 10 samples, this is a large class whose different areas in the image are also large and well defined. The same can be concluded for other classes like Bldg-Grass-Tree-Drives or Woods. However, class Soybeans-clean till is confused with the classes around since the areas where it lies in are small despite of being a big class. The background is a special case, although it is treated here as a single class for segmentation purposes, it consists of different areas with probably considerably different spectral signatures and, if a part of it would be missing in the training data, that part will be misclassified.

Table 1. Accuracy per class for the 17 classes classification of the AVIRIS dataset using 12 features (ten spectral features and two spatial coordinates). For a training sets of 0.33%, 2% and 4% of the data using the clustering-based classifier.

	0.33% of training data		2% of training data		4% of training data	
classes	training/total	error	training/total	error	training/total	error
Heterogenous background	22/10659	0.432	171/10659	0.262	367/10659	0.193
Stone-steel towers	0/95	1	2/95	0.139	5/95	0.033
Hay-windrowed	2/489	0.004	10/489	0.004	25/489	0.004
Corn-min till	5/834	0.214	18/834	0.076	40/834	0.045
Soybeans-no till	5/968	0.185	25/968	0.060	40/968	0.072
Alfalfa	0/54	1	1/54	0.038	3/54	0.039
Soybeans-clean till	2/614	0.488	15/614	0.066	28/614	0.056
Grass/pasture	3/497	0.105	12/497	0.064	28/497	0.042
Woods	6/1294	0.023	29/1294	0.034	58/1294	0.026
Bldg-Grass-Tree-Drives	3/380	0.021	9/380	0.011	12/380	0.011
Grass/pasture-mowed	0/26	1	1/26	0.040	1/26	0.040
Corn	1/234	0.601	6/234	0.070	10/234	0.049
Oats	0/20	1	0/20	1	0/20	1
Corn-no till	6/1434	0.278	35/1434	0.067	63/1434	0.035
Soybeans-min till	10/2468	0.069	70/2468	0.023	143/2468	0.018
Grass/trees	4/747	0.067	18/747	0.033	34/747	0.042
Wheat	1/212	0.009	7/212	0.005	11/212	0.005
Overall error		0.299		0.156		0.116

6 Conclusions

A training data selection method has been proposed in a segmentation classification scheme for scenarios in which no prior knowledge is available. This aims at improving classification and reducing the interaction with the expert who would label a very small set of points only once. This is highly interesting when expert collaboration is expensive. To get representative training data, mode seek clustering is preformed. This type of clustering provides modes (representative samples) for each cluster found in the feature space and those modes are the selected samples for labeling. Thanks to a spatial improvement in the clustering, the modes provided do not contain redundant training information and can represent different spatial areas in the image that belong to the same class. The training selection has been used over several classifiers. We have experimentally proved that distance based classifiers are more adequate than SVM for such an approach. Furthermore, we have also shown that the classification obtained from the mode seek clustering outperformed the simple distance based classifiers because it better adapts to the shapes of the clusters in the feature space.

All classification strategies benefit from the selection of the labeled data to improve their performances. They provide very good results even with less labeled data than provided in other scenarios where training data was randomly selected.

References

1. Tarabalka, Y., Chanussot, J., Benediktsson, J.A.: Segmentation and classification of hyperspectral images using watershed transformation. Patt. Recogn. 43, 2367–2379 (2010)
2. Plaza, A., et al.: Recent advances in techniques for hyperspectral image processing. Remote Sensing of Environment 113, 110–122 (2009)
3. Tuia, D., Ratle, F., Pacifici, F., Kanevski, M., Emery, W.: Active learning methods for remote sensing image classification. IEEE Transactions on Geoscience and Remote Sensing 47, 2218–2232 (2009)
4. Li, J., Bioucas-Dias, J., Plaza, A.: Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. IEEE TGRS 48, 4085–4098 (2010)
5. Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 24, 603–619 (2002)
6. Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 898–916 (2011)
7. Rajadell, O., Dinh, V.C., Duin, R.P., García-Sevilla, P.: Semi-supervised hyperspectral pixel classification using interactive labeling. In: Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, WHISPERS 2011 (2011)
8. Martínez-Usó, A., Pla, F., Sotoca, J., García-Sevilla, P.: Clustering-based hyperspectral band selection using information measures. IEEE Trans. on Geoscience & Remote Sensing 45, 4158–4171 (2007)
9. Cheng, Y.: Mean shift, mode seek, and clustering. IEEE Transaction on Pattern Analysis and Machine 17, 790–799 (1995)
10. Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2006)
11. Chi, M., Yu, X.H.S.: Mixture model label propagation. In: 19th ACM International Conference on Information and Knowledge Management, pp. 1889–1892 (2010)
12. Landgrebe, D.A.: Signal Theory Methods in Multispectral Remote Sensing, 1st edn. Wiley, Hoboken (2003)

Hyperspectral Imagery Framework for Unmixing and Dimensionality Estimation*

José M.P. Nascimento[1] and José M. Bioucas-Dias[2]

[1] Instituto de Telecomunicações and Instituto Superior de Engenharia de Lisboa
R. Conselheiro Emídio Navarro, N. 1, 1959-007 Lisbon, Portugal

[2] Instituto de Telecomunicações and Instituto Superior Técnico, Technical University of Lisbon,
Av. Rovisco Pais, Torre Norte, Piso 10, 1049-001 Lisbon, Portugal

zen@isel.pt

http://www.deetc.isel.pt/jnascimento/

Abstract. In hyperspectral imagery a pixel typically consists mixture of spectral signatures of reference substances, also called endmembers. Linear spectral mixture analysis, or linear unmixing, aims at estimating the number of endmembers, their spectral signatures, and their abundance fractions.

This paper proposes a framework for hyperpsectral unmixing. A blind method (SISAL) is used for the estimation of the unknown endmember signature and their abundance fractions. This method solve a non-convex problem by a sequence of augmented Lagrangian optimizations, where the positivity constraints, forcing the spectral vectors to belong to the convex hull of the endmember signatures, are replaced by soft constraints. The proposed framework simultaneously estimates the number of endmembers present in the hyperspectral image by an algorithm based on the minimum description length (MDL) principle. Experimental results on both synthetic and real hyperspectral data demonstrate the effectiveness of the proposed algorithm.

Keywords: Blind hyperspectral unmixing, Minimum volume simplex, Minimum Description Length (MDL), Variable splitting augmented lagrangian, Dimensionality reduction.

1 Introduction

Although, there have been significant improvements in the hyperspectral sensors, there are in an image pixels that contain more than one substance, i.e., the acquired spectral vectors are mixtures of the substances spectral signatures present in the scene [6,19].

The linear mixing assumption has been widely used to describe the observed hyperspectral vectors. According to this assumption, a mixed pixel is a linear combination of endmembers signatures weighted by the corresponding abundance fractions. Due to physical considerations, the abundance fractions are subject to the so-called nonnegativity and a full-additivity constraints [6].

* This work was supported by the Instituto de Telecomunicações and by the Fundação para a Ciência e Tecnologia under project HoHus (PEst-OE/EEI/LA0008/2011).

P.L. Carmona et al. (Eds.): *Pattern Recognition – Applications and Methods*, AISC 204, pp. 193–204.
DOI: 10.1007/978-3-642-36530-0_16

Hyperspectral unmixing, aims at estimating the number of reference materials, also called endmembers, their spectral signatures, and their abundance fractions [17]. Hyperspectral linear unmixing approaches can be classified as either statistical or geometrical. Statistical methods very often formulate the problem under the Bayesian framework [14] [1] [18] [21].

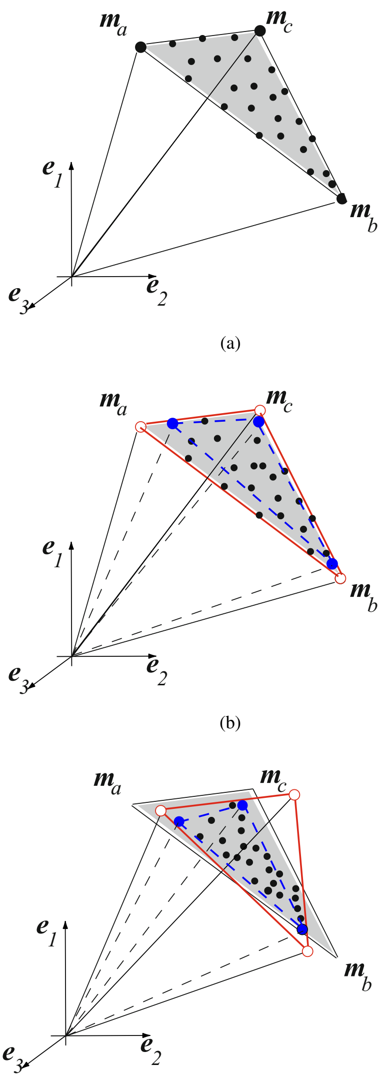
The geometric perspective just referred to has been exploited by many algorithms. These algorithms are based on the fact that, under the linear mixing model, hyperspectral vectors belong to a simplex set whose vertices correspond to the endmembers signatures. Thus, finding the endmembers is equivalent to identifying the vertices of the referred to simplex [20].

Some algorithms assume the presence of, at least, one pure pixel per endmember (*i. e.*, containing just one material). Some popular algorithms taking this assumption are the *pixel purity index* (PPI), [7], *vertex component analysis* (VCA), [20], the *automated morphological endmember extraction* (AMEE) [22], and the N-FINDR [26] (see [9] for recently introduced reinterpretations and improvements of N-FINDR). These methods are followed by a fully constrained least square estimation [16] or by a maximum likelihood estimation [24] of the abundance fractions to complete the unmixing procedure.

If the pure pixel assumption is not fulfilled, which is a more realistic scenario, the unmixing process is a rather challenging task, since some endmembers are not in the dataset. Some recent methods, in the vein of Craig's work *minimum Volume Transform* (MVT) [12] which finds the smallest simplex that contain the dataset, are the *simplex identification via split augmented Lagrangian* (SISAL) [4], *iterated constrained endmember* (ICE), [3], the *minimum-volume enclosing simplex algorithm* (MVES) [10], *successive volume maximization* (SVMAX) [9], and the *alternating projected subgradients* (APS) [28].

Fig. 1 illustrates three datasets raising different degrees of difficulties in what unmixing is concerned: the dataset shown in Fig.1(a) contains pure pixels, *i.e.*, the spectra corresponding to the simplex vertices are in the dataset. This is the easiest scenario with which all the unmixing algorithms cope without problems; the dataset shown in Fig.1(b) does not contain pure pixels, at least for some endmembers. This is a much more challenging, usually attacked with the minimum volume based methods, note that pure-pixels based methods are outperformed under these circumstances; Fig.1(c), contains a highly mixed dataset where only statistical methods can give accurate unmixing results.

Most of these methods assume that the number of endmembers are known a-priori or estimated for some method, such as, NWHFC [11], HySime [5], and *Second moment linear dimensionality* (SML) [2]. The *robust signal subspace estimation* (RSSE) [13] have been proposed in order to estimate the signal subspace in the presence of rare signal pixels, thus it can be used as a preprocessing step for small target detection applications. *Sparsity promoting ICE* (SPICE) [27] is an extension of ICE algorithm that incorporates sparsity-promoting priors to find the correct number of endmembers. The framework presented in [8] also estimates the number of endmembers when it unmix the data. This framework has the disadvantage of using the Unsupervised Fully Constrained Least Squares (UFCLS) algorithm proposed in [16] which assumes the presence of at least one pure pixel per endmember.



(a)

(b)

(c)

Fig. 1. Illustration of tree scenarios: (a) with pure pixels (solid line - estimated simplex by all methods); (b) without pure pixels and with pixels in the facets (solid red line - estimated simplex based on minimum volume; dashed blue line - estimated simplex by pure-pixel based methods); (c) highly mixed pixels (solid red line - estimated simplex based on minimum volume; dashed blue line - estimated simplex by pure-pixel based methods)

This paper proposes a framework for linear hyperpsectral unmixing. SISAL [4] is used for the estimation of the endmember signature and their abundance fractions, while, based on the minimum description length (MDL) principle the number of endmembers is inferred. SISAL belongs to the minimum volume class methods.

This paper is organized as follows. Section 2 formulates the problem and describes the fundamentals of the proposed method. Section 3 presents the method to infer the number of endmembers. Section 4 illustrates aspects of the performance of the proposed approach with experimental data based on U.S.G.S. laboratory spectra and with real hyperspectral data collected by the AVIRIS sensor, respectively. Section 5 concludes with some remarks.

2 Problem Formulation

Assuming the linear observation model, each pixel $\mathbf{y}$ of an hyperspectral image can be represented as a spectral vector in $\mathbb{R}^l$ (l is the number of bands) and is given by $\mathbf{y} = \mathbf{M}\mathbf{s} + \mathbf{n}$, where $\mathbf{M} \equiv [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p]$ is an $l \times p$ mixing matrix ($\mathbf{m}_j$ denotes the jth endmember spectral signature), p is the number of endmembers present in the covered area, $\mathbf{s} = [s_1, s_2, \ldots, s_p]^T$ is the abundance vector containing the fractions of each endmember (notation $(\cdot)^T$ stands for vector transposed), and vector $\mathbf{n}$ holds the sensor noise and modeling errors.

To fix notation, let $\mathbf{Y} \equiv [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \mathbb{R}^{l\times n}$ denote a matrix holding the n observed spectral vectors, $\mathbf{S} \equiv [\mathbf{s}_1, \ldots, \mathbf{s}_n] \in \mathbb{R}^{p\times n}$ a matrix holding the respective abundance fractions, and $\mathbf{N} \equiv [\mathbf{n}_1, \ldots, \mathbf{n}_n] \in \mathbb{R}^{l\times n}$ accounts for additive noise. To be physically meaningful, abundance fractions are subject to non-negativity and constant sum constraints, *i.e.*, $\{\mathbf{s} \in \mathbb{R}^p : \mathbf{s} \succeq \mathbf{0}, \mathbf{1}_p^T\mathbf{s} = \mathbf{1}_n^T\}$[1]. Therefore

$$\begin{aligned} \mathbf{Y} &= \mathbf{M}\mathbf{S} + \mathbf{N} \\ \text{s.t. :} \quad & \mathbf{S} \succeq \mathbf{0}\,,\ \mathbf{1}_p^T\mathbf{S} = \mathbf{1}_n^T. \end{aligned} \tag{1}$$

Usually the number of endmembers is much lower than the number of bands ($p \ll L$). Thus, the observed spectral vectors can be projected onto the signal subspace. The identification of the signal subspace improves the SNR, allows a correct dimension reduction, and thus yields gains in computational time and complexity [5].

Let $\mathbf{E}_p$ be a matrix, with orthonormal columns, spanning the signal subspace. Thus

$$\begin{aligned} \mathbf{X} &\equiv \mathbf{E}_p^T\mathbf{Y} + \mathbf{E}_p^T\mathbf{N} \\ &= \mathbf{A}\mathbf{S} + \mathbf{N}^*, \end{aligned} \tag{2}$$

where $\mathbf{X} \equiv [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p\times n}$ denote a matrix holding the projected spectral vectors, $\mathbf{A} = \mathbf{E}_p^T\mathbf{M}$ is a $p \times p$ square mixing matrix, and $\mathbf{N}^*$ accounts for the projected noise.

Linear unmixing amounts to infer matrices $\mathbf{A}$ and $\mathbf{S}$. This can be achieved by fitting a minimum volume simplex to the dataset [12]. Finding a minimum volume matrix $\mathbf{A}$ subject to constraints in (1), leads to the non-convex optimization problem

$$\begin{aligned} \widehat{\mathbf{Q}} &= \arg\min_{Q} \{-\log|\det \mathbf{Q}|\} \\ \text{s.t. :} \quad & \mathbf{Q}\mathbf{X} \succeq \mathbf{0}\,,\ \mathbf{1}_p^T\mathbf{Q}\mathbf{X} = \mathbf{1}_n^T, \end{aligned} \tag{3}$$

[1] $\mathbf{s} \succeq \mathbf{0}$ means $s_j \geq 0$, for $j = 1, \ldots, p$ and $\mathbf{1}_p^T \equiv [1, \ldots, 1]$.

where $\mathbf{Q} \equiv \mathbf{A}^{-1}$. The constraint $\mathbf{1}_p^T \mathbf{Q}\mathbf{X} = \mathbf{1}_n^T$ can be simplified, by multiplying the equality on the right hand side by $\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$, resulting $\mathbf{1}_p^T \mathbf{Q}\mathbf{X} = \mathbf{1}_n^T \Leftrightarrow \mathbf{1}_p^T \mathbf{Q} = \mathbf{a}^T$, where $\mathbf{a}^T \equiv \mathbf{1}_n^T \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1}$.

SISAL aims to give a sub-optimal solution of (3) solving the following problem by a sequence of augmented Lagrangian optimizations:

$$\begin{aligned} \widehat{\mathbf{Q}}^* &= \arg\min_{Q} \{-\log|\det \mathbf{Q}| + \lambda \|\mathbf{Q}\mathbf{X}\|_h\} \\ \text{s.t. } : \quad & \mathbf{1}_p^T \mathbf{Q} = \mathbf{a}^T, \end{aligned} \tag{4}$$

where $\|\mathbf{Q}\mathbf{X}\|_h \equiv \sum_{ij} h(\mathbf{Q}\mathbf{X})$, $h(x) \equiv \max(-x, 0)$ is the so-called hinge function and λ is the regularization parameter. Notice that $\|\mathbf{Q}\mathbf{X}\|_h$ penalizes negative components of $\mathbf{Q}\mathbf{X}$, thus playing the rule of a soft constraint, yielding solutions that are robust to outliers, noise, and poor initialization.(see [4] for details).

3 Number of Endmembers Estimation

The MDL principle proposed by Rissanen [23] aims to select the model that offers the shortest description length of the data. This approach can be used to estimate the number of endmembers [8]. The well-known MDL criterion for n i.i.d. observations, in general, is given by

$$\widehat{k}_{MDL} = \arg\min_{k} \left\{ \mathcal{L}(\mathbf{X}|\widehat{\boldsymbol{\theta}}_k) + \frac{1}{2}\, k \log n \right\}, \tag{5}$$

where $\mathcal{L}(\mathbf{X}|\widehat{\boldsymbol{\theta}}_k)$ is a likelihood function based on the projected data $\mathbf{X}$ with parameters $\boldsymbol{\theta}$, and $\frac{1}{2}\, k \log n$ is an increasing function penalizing higher values of k [15].

Assuming that the additive noise is Gaussian distributed, *i.e.* $\mathbf{n} \sim \mathcal{N}(0, \boldsymbol{\Lambda})$ and given a set of n i.i.d. observed samples, the likelihood equation is given by:

$$\begin{aligned} \mathcal{L}(\mathbf{X}|\widehat{\boldsymbol{\theta}}_k) &\equiv \sum_{i=1}^{n} \Big[-\log\, p(\mathbf{x}_i|\widehat{\boldsymbol{\theta}}_k) \Big] \\ &= \frac{n}{2}\Big(p \log(2\pi) + \log|\det \boldsymbol{\Lambda}| \Big) + \frac{1}{2} \mathrm{tr} \Big[(\mathbf{X} - \mathbf{A}\mathbf{S})^T\, \boldsymbol{\Lambda}^{-1}\, (\mathbf{X} - \mathbf{A}\mathbf{S}) \Big], \end{aligned} \tag{6}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix, matrices $\mathbf{A}$ and $\mathbf{S}$ are replaced by their estimates using SISAL algorithm, the noise covariance matrix, $\boldsymbol{\Lambda}$, is estimated using the algorithm based on the multiple regression theory proposed in [5] and the number of free parameters is $k = p^2$. The resulting optimization algorithm is an iterative scheme that requires to compute the objective function and to estimate the matrices $\mathbf{A}$, $\mathbf{S}$, and $\boldsymbol{\Lambda}$ for each value of p.

4 Experiments

This section provides simulated and real data experiments to illustrate the algorithm's performance. The proposed method is tested and compared with SPICE [27] on different simulated scenarios concerning with different signal-to-noise ratio (SNR), absence of pure pixels, and number of endmembers present in the scene. The proposed method is also applied to real hyperspectral data collected by the AVIRIS sensor over Cuprite, Nevada.

4.1 Evaluation with Simulated Data

In this section the proposed method is tested on simulated scenes. To evaluate the performance of the algorithm the well-known spectral angle distance (SAD) metric is used [17]. SAD measures the shape similarity between the ith endmember signature $\mathbf{m_i}$ and its estimate $\widehat{\mathbf{m}}_i$. Based on this metric we define a spectral root mean square angle error, given by:

$$\epsilon_m \equiv \frac{1}{p}\left[\sum_{i=1}^{p}\left(\arccos\frac{\mathbf{m}_i^T\widehat{\mathbf{m}}_i}{\|\mathbf{m}_i\|\|\widehat{\mathbf{m}}_i\|}\right)^2\right]^{1/2}. \tag{7}$$

To measure the similarity between the observed data and the unmix result it is also computed the residual error between the observed pixels and their estimates:

$$r_{ls} \equiv \|\mathbf{Y} - \widehat{\mathbf{M}}\widehat{\mathbf{S}}\|_F^2, \tag{8}$$

where $\widehat{\mathbf{M}} = \mathbf{E}_p\widehat{\mathbf{A}}$ and $\widehat{\mathbf{S}}$ are estimated by SISAL.

Concerning the simulated data creation an hyperspectral image composed of 10^4 pixels is generated according to expression (1), where spectral signatures where selected from the USGS digital spectral library. The selection of endmember signatures is arbitrary as long as they are linearly independent. The reflectances contain 224 spectral bands covering wavelengths from 0.38 to $2.5\,\mu m$ with a spectral resolution of $10\,nm$. The abundance fractions are generated according to a Dirichlet distribution given by

$$D(s_1,\ldots,s_p|\mu_1,\ldots,\mu_p) = \frac{\Gamma(\sum_{j=1}^{p}\mu_j)}{\prod_{j=1}^{p}\Gamma(\mu_j)}\prod_{j=1}^{p}s_j^{\mu_j-1}. \tag{9}$$

This density, besides enforcing positivity and full additivity constraints, displays a wide range of shapes, depending on the parameters of the distribution $\boldsymbol{\mu} = [\mu_1,\ldots,\mu_p]$.

In this experiment the Dirichlet parameters are set to $\boldsymbol{\mu} = [3,\ldots,3]$, concerning the additive noise, the SNR, which is defined as

$$\text{SNR} \equiv 10\log_{10}\left(\mathbb{E}\{\mathbf{y}^T\mathbf{y}\}/\mathbb{E}\{\mathbf{n}^T\mathbf{n}\}\right), \tag{10}$$

is set to $30\,dB$.

Fig. 2 presents a scatterplot of the simulated scene for the $p = 3$ case, where dots represent the pixels and circles represent the true endmembers. This figure also shows the endmembers estimates (squares) which are very close to the true ones. Fig.3 shows the endmembers signatures (solid line) and their estimates (dashed line). Note that, in this experiment there is no pure pixels in the dataset, however, the endmembers estimate is very accurate.

Fig 4, presents the evolution of the cost function [see expression (5)] as a function of the number of endmembers. The minimum of the function occurs at $\widehat{k} = 3$ which is the true number of endmembers in the scene.

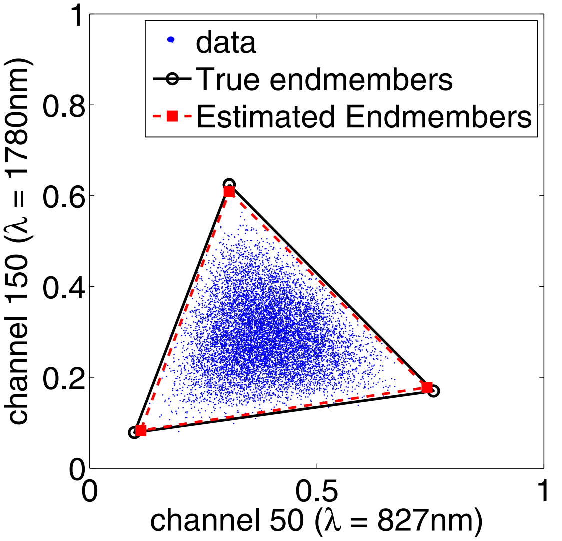


Fig. 2. Scatterplot of the three endmembers mixture: Dataset (blue dots); true endmembers (black circles); Proposed method estimates (red squares)

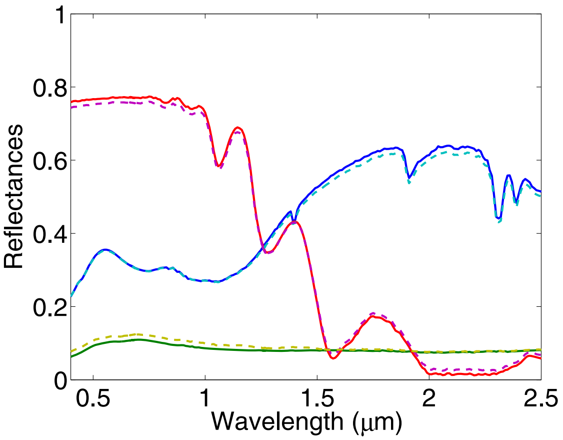


Fig. 3. Endmembers signatures (solid line) and their estimates (dashed line)

Table 1 presents the root mean square error distance ϵ_m, the residual least squares error r_{ls}, and the estimated number of endmembers for different experiments: p is set to $\{3, 5, 10\}$ and the SNR is set to $\{30, 50\}$ dB. Note that the estimated values are exactly the number of endmembers in the scene and the unmix error increases with increasing values of p and with noise level. The results achieved by SPICE in terms of residual error are similar to the proposed method results, although the errors between endmembers signatures and their estimates are worst.

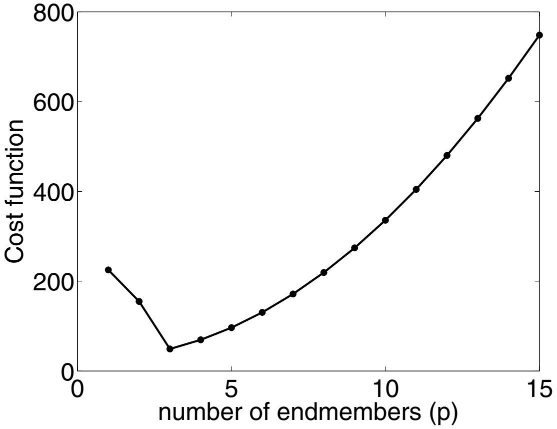


Fig. 4. Cost function evolution as a function of the number of endmembers

Table 1. Results for different scenarios as a function of the SNR and of the number of endmembers (p)

		Proposed Method			SPICE		
SNR	p	$\widehat{k}$	ϵ_m	r_{ls}	$\widehat{k}$	ϵ_m	r_{ls}
30 dB	3	3	0.048	4.76	3	0.293	4.82
	5	5	0.053	6.41	5	0.198	6.47
	10	10	0.929	6.99	6	0.258	7.18
50 dB	3	3	0.042	0.47	3	0.141	1.06
	5	5	0.059	0.64	5	0.432	1.30
	10	10	0.196	0.70	6	0.268	1.70

4.2 Experiments with Real Hyperspectral Data

In this section, the proposed method is applied to a subset (50×90 pixels and 224 bands) of the Cuprite dataset acquired by the AVIRIS sensor on June 19, 1997, Fig. 5 shows band 30 (wavelength $\lambda = 667.3nm$) of the subimage of AVIRIS cuprite Nevada dataset. The AVIRIS instrument covers the spectral region from $0.41\,\mu m$ to $2.45\,\mu m$ in 224 bands with a $10\,nm$ band width. Flying at an altitude of $20\,km$, it has an IFOV of $20\,m$ and views a swath over $10\,km$ wide. This site has been extensively used for remote sensing experiments over the past years and its geology was previously mapped in detail [25].

Table 2 present the residual error and the estimated number of endmembers for SPICE and for the proposed method. The results of both methods are comparable.

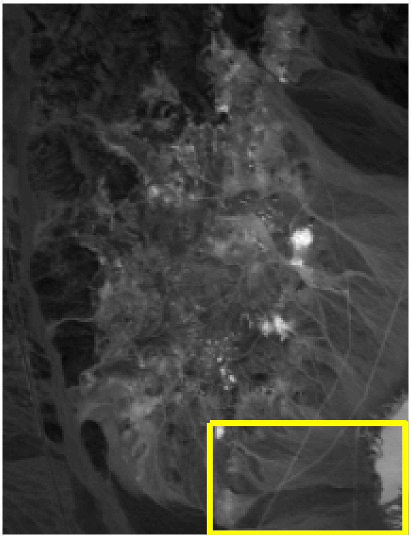


Fig. 5. Band 30 (wavelength $\lambda = 655.8nm$) of the subimage of AVIRIS Cuprite Nevada dataset (rectangle denotes the image fraction used in the experiment)

Table 2. Results for Cuprite dataset

	Proposed method	SPICE
$\widehat{k}$	6	7
r_{ls}	3.13	3.27

Fig.6 (left) shows the estimated signatures, which are compared with the nearest laboratory spectra, to visually distinguish the different endmembers an offset has been added to each signature. Note that, this endmembers are known to dominate the considered subimage [25].

Fig.6 (right) presents the estimated abundance maps for the extracted endmembers. A visual comparison show that these maps are in accordance with the known ground truth. Note that for this region Desert vanish (Fig.6b)) and Sphene (Fig. 6d)) abundance maps are very similar. These results show the potential of the proposed method to simultaneously select the number of endmembers, estimate the spectral signatures, and their abundance fractions.

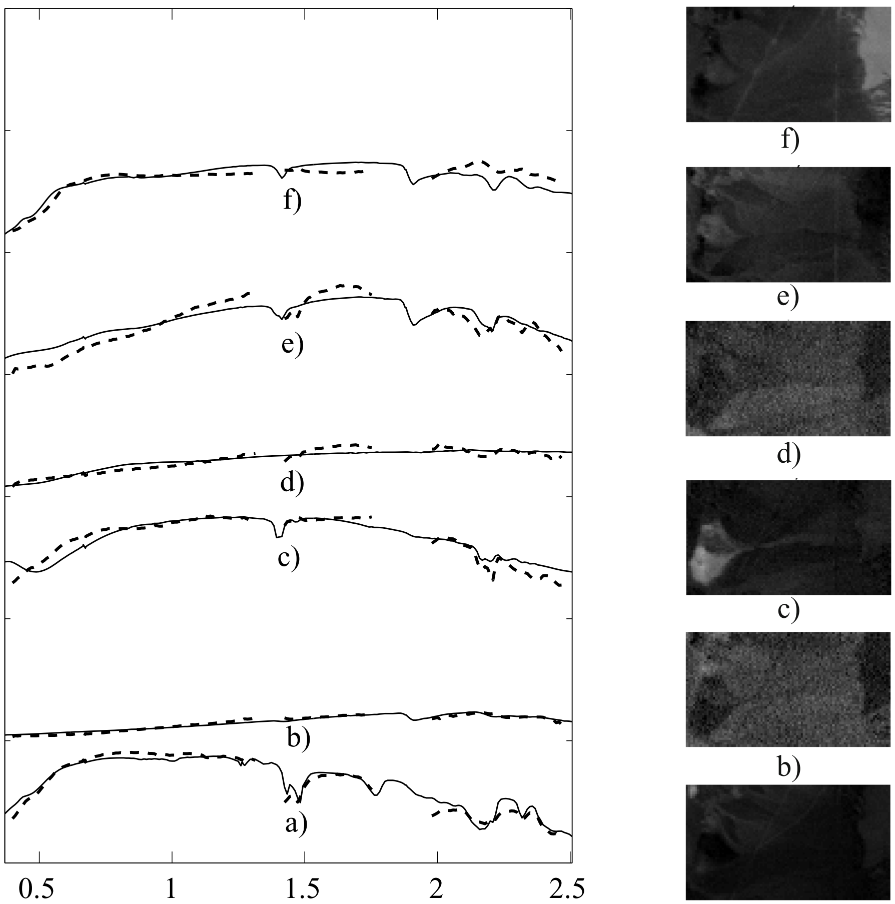


Fig. 6. Experimental results on Cuprite dataset. Left: Comparison of the estimated signatures (dashed line) with the nearest USGS spectra (solid line). Right: Abundance maps estimates. a) Alunite; b) Desert vanish; c) Dumortierite; d) Sphene; e) Kaolinite; f) Montmorillonite.

5 Conclusions

In this paper, a new framework is proposed to blindly unmix hyperspectral data and simultaneously infer the number of endmembers based on the minimum description length (MDL) principle. The estimation of the endmembers spectra and their abundance fractions is based on SISAL, which is a minimum-volume type method, that solves a non-convex problem by a sequence of augmented Lagrangian optimizations, where the positivity constraints, forcing the spectral vectors to belong to the convex hull of the endmember signatures, are replaced by soft constraints. The experimental results achieved on simulated and on real datasets show the potential of the proposed method.

References

1. Arngren, M., Schmidt, M.N., larsen, J.: Bayesian Nonnegative Matrix Factorization with Volume Prior for Unmixing of Hyperspectral Images. In: IEEE Workshop on Machine Learning for Signal Processing (MLSP) (September 2009)
2. Bajorski, P.: Second Moment Linear Dimensionality as an Alternative to Virtual Dimensionality. IEEE Trans. Geosci. Remote Sensing 49(2), 672–678 (2011)
3. Berman, M., Kiiveri, H., Lagerstrom, R., Ernst, A., Dunne, R., Huntington, J.F.: ICE: A Statistical Approach to Identifying Endmembers in Hyperspectral Images. IEEE Trans. Geosci. Remote Sensing 42(10), 2085–2095 (2004)
4. Bioucas-Dias, J.M.: A Variable Splitting Augmented Lagrangian Approach to Linear Spectral Unmixing. In: First IEEE GRSS Workshop on Hyperspectral Image and Signal Processing-WHISPERS 2009 (2009)
5. Bioucas-Dias, J.M., Nascimento, J.M.P.: Hyperspectral Subspace Identification. IEEE Trans. Geosci. Remote Sensing 46(8), 2435–2445 (2008)
6. Bioucas-Dias, J.M., Plaza, A.: Hyperspectral unmixing: geometrical, statistical, and sparse regression-based approaches, vol. 7830. SPIE (2010)
7. Boardman, J.: Automating Spectral Unmixing of AVIRIS Data using Convex Geometry Concepts. In: Summaries of the Fourth Annual JPL Airborne Geoscience Workshop, JPL Pub. 93-26, AVIRIS Workshop, vol. 1, pp. 11–14 (1993)
8. Broadwater, J., Meth, R., Chellappa, R.: Dimensionality Estimation in Hyper-spectral Imagery Using Minimum Description Length. In: Proceedings of the Army Science Conference, Orlando, FL (November 2004)
9. Chan, T.H., Ma, W.K., Ambikapathi, A., Chi, C.Y.: A simplex volume maximization framework for hyperspectral endmember extraction. IEEE Trans. Geosci. Remote Sensing 49(1), 1–17 (2011)
10. Chan, T.H., Chi, C.Y., Huang, Y.M., Ma, W.K.: A Convex Analysis-Based Minimum-Volume Enclosing Simplex Algorithm for Hyperspectral Unmixing. IEEE Trans. Signal Processing 57(11), 4418–4432 (2009)
11. Chang, C.I., Du, Q.: Estimation of Number of Spectrally Distinct Signal Sources in Hyperspectral Imagery. IEEE Trans. Geosci. Remote Sensing 42(3), 608–619 (2004)
12. Craig, M.D.: Minimum-volume Transforms for Remotely Sensed Data. IEEE Trans. Geosci. Remote Sensing 32, 99–109 (1994)
13. Diani, N.A.M., Corsini, G.: Hyperspectral Signal Subspace Identification in the Presence of Rare Signal Components. IEEE Trans. Geosci. Remote Sensing 48(4), 1940–1954 (2010)
14. Dobigeon, N., Moussaoui, S., Coulon, M., Tourneret, J.Y., Hero, A.O.: Joint Bayesian Endmember Extraction and Linear Unmixing for Hyperspectral Imagery. IEEE Trans. Signal Processing 57(11), 4355–4368 (2009)
15. Figueiredo, M.A.T., Jain, A.K.: Unsupervised Learning of Finite Mixture Models. IEEE Trans. Pattern Anal. Machine Intell. 44(3), 381–396 (2002)
16. Heinz, D., Chang, C.-I.: Fully Constrained Least Squares Linear Spectral Mixture Analysis Method for Material Quantification in Hyperspectral Imagery. IEEE Transactions on Geoscience and Remote Sensing 39(3), 529–545 (2001)
17. Keshava, N., Mustard, J.: Spectral Unmixing. IEEE Signal Processing Mag. 19(1), 44–57 (2002)
18. Moussaoui, S., Hauksdóttir, H., Schmidt, F., Jutten, C., Chanussot, J., Brie, D., Douté, S., Benediktsson, J.A.: On the Decomposition of Mars Hyperspectral Data by ICA and Bayesian Positive Source Separation. Neurocomputing 71(10-12), 2194–2208 (2008)
19. Nascimento, J.M.P., Bioucas-Dias, J.M.: Does Independent Component Analysis Play a Role in Unmixing Hyperspectral Data? IEEE Trans. Geosci. Remote Sensing 43(1), 175–187 (2005)

20. Nascimento, J.M.P., Bioucas-Dias, J.M.: Vertex Component Analysis: A Fast Algorithm to Unmix Hyperspectral Data. IEEE Trans. Geosci. Remote Sensing 43(4), 898–910 (2005)
21. Nascimento, J.M.P., Bioucas-Dias, J.M.: Hyperspectral unmixing based on mixtures of dirichlet components. IEEE Transactions on Geoscience and Remote Sensing 50(3), 863–878 (2012)
22. Plaza, A., Martinez, P., Perez, R., Plaza, J.: Spatial/Spectral Endmember Extraction by Multidimensional Morphological Operations. IEEE Trans. Geosci. Remote Sensing 40(9), 2025–2041 (2002)
23. Rissanen, J.: Modeling by Shortest Data Description. Automatica 14, 465–471 (1978)
24. Settle, J.J.: On the Relationship Between Spectral Unmixing and Subspace Projection. IEEE Trans. Geosci. Remote Sensing 34, 1045–1046 (1996)
25. Swayze, G., Clark, R., Sutley, S., Gallagher, A.: Ground-Truthing AVIRIS Mineral Mapping at Cuprite, Nevada. Summaries of the Third Annual JPL Airborne Geosciences Workshop, pp. 47–49 (1992)
26. Winter, M.E.: N-FINDR: An Algorithm for Fast Autonomous Spectral End-member Determination in Hyperspectral Data. In: Proc. of the SPIE Conference on Imaging Spectrometry, vol. 3753, pp. 266–275 (1999)
27. Zare, A., Gader, P.: Sparsity Promoting Iterated Constrained Endmember Detection in Hyperspectral Imagery. IEEE Geosci. Remote Sensing Let. 4(3), 446–450 (2007)
28. Zymnis, A., Kim, S.J., Skaf, J., Parente, M., Boyd, S.: Hyperspectral Image Unmixing via Alternating Projected Subgradients. In: 41st Asilomar Conferece on Signals, Systems, and Computer, pp. 4–7 (2007)

Author Index