Studies in Computational Intelligence 292

Fabrice Guillet
Gilbert Ritschard
Djamel Abdelkader Zighed
Henri Briand (Eds.)

# Advances in Knowledge Discovery and Management

Springer

Fabrice Guillet, Gilbert Ritschard, Djamel Abdelkader Zighed,
and Henri Briand (Eds.)

Advances in Knowledge Discovery and Management

Studies in Computational Intelligence, Volume 292

**Editor-in-Chief**
Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
*E-mail:* kacprzyk@ibspan.waw.pl

---

Further volumes of this series can be found on our homepage: springer.com

Vol. 271. Janusz Kacprzyk, Frederick E. Petry, and Adnan Yazici (Eds.)
*Uncertainty Approaches for Spatial Data Modeling and Processing,* 2009
ISBN 978-3-642-10662-0

Vol. 272. Carlos A. Coello Coello, Clarisse Dhaenens, and Laetitia Jourdan (Eds.)
*Advances in Multi-Objective Nature Inspired Computing,* 2009
ISBN 978-3-642-11217-1

Vol. 273. Fatos Xhafa, Santi Caballé, Ajith Abraham, Thanasis Daradoumis, and Angel Alejandro Juan Perez (Eds.)
*Computational Intelligence for Technology Enhanced Learning,* 2010
ISBN 978-3-642-11223-2

Vol. 274. Zbigniew W. Raś and Alicja Wieczorkowska (Eds.)
*Advances in Music Information Retrieval,* 2010
ISBN 978-3-642-11673-5

Vol. 275. Dilip Kumar Pratihar and Lakhmi C. Jain (Eds.)
*Intelligent Autonomous Systems,* 2010
ISBN 978-3-642-11675-9

Vol. 276. Jacek Mańdziuk
*Knowledge-Free and Learning-Based Methods in Intelligent Game Playing,* 2010
ISBN 978-3-642-11677-3

Vol. 277. Filippo Spagnolo and Benedetto Di Paola (Eds.)
*European and Chinese Cognitive Styles and their Impact on Teaching Mathematics,* 2010
ISBN 978-3-642-11679-7

Vol. 278. Radomir S. Stankovic and Jaakko Astola
*From Boolean Logic to Switching Circuits and Automata,* 2010
ISBN 978-3-642-11681-0

Vol. 279. Manolis Wallace, Ioannis E. Anagnostopoulos, Phivos Mylonas, and Maria Bielikova (Eds.)
*Semantics in Adaptive and Personalized Services,* 2010
ISBN 978-3-642-11683-4

Vol. 280. Chang Wen Chen, Zhu Li, and Shiguo Lian (Eds.)
*Intelligent Multimedia Communication: Techniques and Applications,* 2010
ISBN 978-3-642-11685-8

Vol. 281. Robert Babuska and Frans C.A. Groen (Eds.)
*Interactive Collaborative Information Systems,* 2010
ISBN 978-3-642-11687-2

Vol. 282. Husrev Taha Sencar, Sergio Velastin, Nikolaos Nikolaidis, and Shiguo Lian (Eds.)
Intelligent Multimedia Analysis for Security Applications, 2010
ISBN 978-3-642-11754-1

Vol. 283. Ngoc Thanh Nguyen, Radoslaw Katarzyniak, and Shyi-Ming Chen (Eds.)
*Advances in Intelligent Information and Database Systems,* 2010
ISBN 978-3-642-12089-3

Vol. 284. Juan R. González, David Alejandro Pelta, Carlos Cruz, Germán Terrazas, and Natalio Krasnogor (Eds.)
*Nature Inspired Cooperative Strategies for Optimization (NICSO 2010),* 2010
ISBN 978-3-642-12537-9

Vol. 285. Roberto Cipolla, Sebastiano Battiato, and Giovanni Maria Farinella (Eds.)
*Computer Vision,* 2010
ISBN 978-3-642-12847-9

Vol. 286. Zeev Volkovich, Alexander Bolshoy, Valery Kirzhner, and Zeev Barzily
*Genome Clustering,* 2010
ISBN 978-3-642-12951-3

Vol. 287. Dan Schonfeld, Caifeng Shan, Dacheng Tao, and Liang Wang (Eds.)
*Video Search and Mining,* 2010
ISBN 978-3-642-12899-8

Vol. 288. I-Hsien Ting, Hui-Ju Wu, Tien-Hwa Ho (Eds.)
*Mining and Analyzing Social Networks,* 2010
ISBN 978-3-642-13421-0

Vol. 289. Anne Håkansson, Ronald Hartung, and Ngoc Thanh Nguyen (Eds.)
*Agent and Multi-agent Technology for Internet and Enterprise Systems,* 2010
ISBN 978-3-642-13525-5

Vol. 290. Weiliang Xu and John E. Bronlund
*Mastication Robots,* 2010
ISBN 978-3-540-93902-3

Vol. 291. Shimon Whiteson
*Adaptive Representations for Reinforcement Learning,* 2010
ISBN 978-3-642-13931-4

Vol. 292. Fabrice Guillet, Gilbert Ritschard, Djamel Abdelkader Zighed, and Henri Briand (Eds.)
*Advances in Knowledge Discovery and Management,* 2010
ISBN 978-3-642-00579-4

Fabrice Guillet, Gilbert Ritschard,
Djamel Abdelkader Zighed, and Henri Briand (Eds.)

# Advances in Knowledge Discovery and Management

Springer

Fabrice Guillet
LINA (CNRS UMR 6241)
Polytechnic School of Nantes University
rue C. Pauc, BP 50609
F-44306 Nantes Cedex 3
France
E-mail: Fabrice.Guillet@univ-nantes.fr

Djamel Abdelkader Zighed
Laboratoire ERIC
Université Lumière Lyon 2
5, avenue Pierre Mendès-France, Bât L.
69600 Bron
France
E-mail: Abdelkader.Zighed@univ-lyon2.fr

Gilbert Ritschard
Université de Genève
Department of Econometrics
Uni-Mail, 40, bd du Pont-d'Arve, room 5232
CH-1211 Geneva 4
Switzerland
E-mail: Gilbert.Ritschard@unige.ch

Henri Briand
LINA (CNRS UMR 6241)
Polytechnic School of Nantes University
rue C. Pauc, BP 50609
F-44306 Nantes Cedex 3
France
E-mail: Henri.Braind@univ-nantes.fr
http://www.polytech.univ-nantes.fr/COD/?Pages_personnelles:Henri_Briand

ISBN 978-3-642-00579-4 e-ISBN 978-3-642-00580-0

DOI 10.1007/978-3-642-00580-0

Studies in Computational Intelligence ISSN 1860-949X

Library of Congress Control Number: 2010928588



*Typeset* & *Cover Design:* Scientific Publishing Services Pvt. Ltd., Chennai, India.

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

springer.com

# Preface

During the last decade, the French-speaking scientific community developed a very strong research activity in the field of Knowledge Discovery and Management (KDM or EGC for "Extraction et Gestion des Connaissances" in French), which is concerned with, among others, Data Mining, Knowledge Discovery, Business Intelligence, Knowledge Engineering and Semantic Web. This emerging research area has also been strongly stimulated by the rapid growth of information systems and the web semantic issues.

The success of the first two French-speaking EGC Conferences in 2001 and 2002 resulted naturally in 2002 in the foundation of the International French-speaking EGC Association[1]. The Association organizes since then regular conferences and workshops with the aim of promoting exchanges between researchers and companies concerned with KDM and its application in business, administration, industry or public organizations.

The recent and novel research contributions collected in this book are extended and reworked versions of a selection of the best papers that were originally presented in French at the EGC 2009 Conference held in Strasbourg, France on January 2009.

## Structure of the Book

The volume is organized in four parts.

**Part I** includes five papers concerned by various aspects of *supervised learning or information retrieval*.

The first paper by Matthias Studer and his colleagues considers complex objects such as state sequences for which we can compute pairwise dissimilarities and proposes an original ANOVA-like approach for measuring the information that a predictor provides about their discrepancy. The technique can be extended in the form of a regression tree for complex objects, which

[1] Association "Extraction et Gestion des Connaissances" (EGC), *www.egc.asso.fr*

is convincingly demonstrated through an application on sequential data describing Swiss occupational trajectories.

Extension to multiple factors and as a tree structured analysis to find out how covariates can explain the object discrepancy. Application to the study of Swiss occupational trajectories.

In the second article, Nicolas Voisine, Marc Boullé and Carine Hue propose a parameter free Bayesian approach to evaluate the overall quality of a decision tree grown from a large data base. This permits to transform the learning problem into an optimization one consisting in searching the tree that optimizes the overall criterion. Extensive experimentation results demonstrate that such optimal trees reach similar predictive performance as state-of-the-art trees, while being much simpler and hence easier to understand.

Thanh-Nghi Do and his colleagues are interested in random forest approaches for very-high-dimensional data with dependencies. They introduce a new oblique decision tree method with SVM-based split functions that work on randomly chosen predictors. Comparative experiments show that the proposed approach makes on very-high-dimensional data clearly better in terms of precision, recall and accuracy than random forests of C4.5.

The contribution of Nguyen-Khang Pham and his associates deals with large scale content-based image retrieval. The authors propose a solution based on Correspondence Analysis (CA) of SIFT local descriptors and introduce an original incremental CA algorithm that scales to huge databases. Response time is further improved by accounting of contextual dissimilarities during the search process. The efficiency of the proposed process is assessed through a series of tests performed on a database of more than 1 million of images.

In the last paper of this first group, Emanuel Aldea and Isabelle Bloch examine structural representations of images for machine learning and image categorization. Resorting to a graph representation in which edges describe spatial relations, they derive metrics between images by means of a graph kernel approach that explicitly accounts for spatial interactions. The authors extend their approach to the case of fuzzy spatial relations and study its behavior in the context of discriminative learning by means of a series of experimentations.

**Part II** presents five papers concerned with *unsupervised learning* issues.

The first of them by Gilles Hubert and Olivier Teste proposes a new OLAP operator in the context of multidimensional databases that proves very useful to facilitate multigranular analyses. Multigranular analyses aim at looking at the same data at different aggregation levels, which usually supposes to run multiple analyses. The proposed tool permits to switch from one granularity level to the other on the fly during the analysis.

The paper by Sébastien Lefèvre is concerned with image segmentation, an issue that can be seen as a data clustering problem with spatial constraints. Such problems are classically solved by running first a unconstraint clustering

method and submitting then the results to additional spatial post-processing. The author proposes here a new solution able to perform image segmentation in a same single round of analysis.

Nistor Grozavu and his associates propose two Self-Organizing-Map-based algorithms for the selection of relevant features through unsupervised weighting. The proposed methods provide also as a byproduct the characterization of clusters. The interest of the methods is demonstrated through a series of experimental results.

The next paper by Guillaume Cleuziou deals with overlapping clustering and presents two extensions of overlapping $k$-means (OKM). The first one generalizes the $k$-medoids method to overlapping clustering and proves useful in organizing non metric data from their proximity matrix. The second one, suitable for metric data, is a weighted version of OKM that allows for non-spherical clusters.

The last paper of this second group by Romain Bourqui and his co-authors deals also with overlapping clusters but in a dynamic social network setting. Such networks are modeled as graphs and the aim is to decompose it into similar sets of nodes. The paper provides a very efficient algorithm that can detect major changes in the network as it evolves over time.

**Part III** includes two papers on *data streaming* and two on *security*.

Nesrine Gabsi and colleagues present a new approach to build historical summaries of data streams. It is based on a combination of sampling and clustering algorithms. The benefit of this combination is empirically demonstrated.

The paper by Lionel Vinceslas and associates is about the mining of sequential patterns in data streams for which it proposes an algorithm that works online using a deterministic finite automaton as a summary structure.

The two next papers are about intrusion detection. Goverdhan Singh and colleagues are concerned by the rate of false alarms in outlier-based intrusion detection systems. They attempt to reduce that rate by looking at the repetition of intrusions from one system to another and propose solutions for separating the outliers from the normal behavioursin a streaming environment and for comparing the outliers of two systems.

Nischal Verma and associates address the problem of intrusion detection on Internet applications and propose a new secure collaborative approach. The main advantage of the proposed method is both to detectnew attacks by using information stored in different sites and to ensure that private data will not be disclosed.

**Part IV** The last four papers are about *ontologies and semantic*.

The first one by Fayçal Hamdi and his co-authors proposes a two promising ontology-partitioning methods designed to take alignment objective into account in the partitioning process.

The paper by Farid Cerbah is concerned with the automatic construction of rich semantic models or ontologies from relational databases. An important limitation of such automatic processes is that that they most often end up with flat models that simply mirror the definition schemas of the source databases. The paper shows how relevant categorisation patterns can be identified within the data by combining lexical filtering and entropy-based criteria.

The article by Alina Dia Miron and her associates is concerned with formal languages for describing ontologies. It considers OWL Description Logic for which it adapts semantic analysis techniques that permit to exploit individual spatio-temporal annotations to limit the scope of the queries and thus increase efficiency.

The last paper by Alain Lelu and Martine Cadot attempts to find links and anti-links between presence-absence attributes. By mean of a randomization approach the proposed method checks if the co-occurrences in a series of randomized data sets is significantly above (anti-link) or below (link) than the co-occurrences in the original data set. The scope of the method is illustrated on a collection of texts.

## Acknowledgments

The editors would like to thank the chapter authors for their insights and contributions to this book.

The editors would also like to acknowledge the members of the review committee and the associated referees for their involvement in the review process of the book. Their in depth reviewing, criticisms and constructive remarks significantly contributed to the high quality of the retained papers.

A special thank goes to Matthias Studer who has efficiently composed and laid out the manuscript.

Finally, we thank Springer and the publishing team, and especially T. Ditzinger and J. Kacprzyk, for their confidence in our project.

Nantes, Geneva, Lyon
February 2010

Fabrice Guillet
Gilbert Ritschard
Djamel Abdelkader Zighed
Henri Briand

# Organization

## Review Committee

All published chapters have been reviewed by at least 2 referees.

| | |
|---|---|
| Jiawei Han | Univ. of Illinois, USA - Honorary Member |
| Tomas Aluja-Banet | UPC, Barcelona, Spain |
| Nadir Belkhiter | Univ. Laval, Québec, Canada |
| Samy Bengio | Google Inc., Mountain View California, USA |
| Younès Bennani | Univ. Paris 13, France |
| Sadok Ben Yahia | Univ. Tunis, Tunisia |
| Paula Brito | Univ. of Porto, Portugal |
| Gilles Falquet | Univ. of Geneva, Switzerland |
| Georges Gardarin | Univ. of Versailles Saint-Quentin, France |
| Mohand-Saïd Hacid | Univ. of Lyon I, France |
| Robert Hilderman | University of Regina, Canada |
| Yves Kodratoff | LRI, University of Paris-Sud, France |
| Pascale Kuntz | LINA, University of Nantes, France |
| Ludovic Lebart | ENST, Paris, France |
| Philippe Lenca | Enst-Bretagne, Brest, France |
| Philippe Leray | Univ. of Nantes, France |
| Monique Noirhomme-Fraiture | FUNDP, Namur, Belgium |
| Stan Matwin | Univ. of Ottawa, Canada |
| Pascal Poncelet | LIRMM, Univ. of Montpellier, France |
| Jan Rauch | University of Prague, Czech Republic |
| Ansaf Salleb-Aouissi | Columbia Univ., New York, USA |
| Gilbert Saporta | CNAM, Paris, France |
| Florence Sédes | Univ. of Toulouse 3, France |
| Dan Simovici | Univ. of Massachusetts Boston, USA |
| Gilles Venturini | Univ. of Tours, France |
| Jeff Wijsen | Univ. of Mons-Hainaut, Belgium |

## Associated Reviewers

Emanuel Aldea,
Romain Bourqui,
Farid Cerbah,
Guillaume Cleuziou,
Marta Franova,
Céline Fiot,
Nesrine Gabsi,
Nistor Grozavu,
Fayçal Hamdi,
Gilles Hubert,
Alain Lelu,
Sébastien Lefèvre,
Cécile Low-Kam,
Florent Masseglia,
Alina-Dia Miron,
Nguyen-Khang Pham,
Stefano Perabò,
Matthias Studer,
Olivier Teste,
Thanh-Nghi Do,
Lionel Vinceslas,
Nicolas Voisine,

## Manuscript Coordinator

Matthias Studer (Univ. of Geneva, Switzerland)

# Contents

**Part I: Supervised Learning and Information Retrieval**

**Discrepancy Analysis of Complex Objects Using Dissimilarities** ..... 3
*Matthias Studer, Gilbert Ritschard, Alexis Gabadinho, Nicolas S. Müller*

**A Bayes Evaluation Criterion for Decision Trees** ..... 21
*Nicolas Voisine, Marc Boullé, Carine Hue*

**Classifying Very-High-Dimensional Data with Random Forests of Oblique Decision Trees** ..... 39
*Thanh-Nghi Do, Philippe Lenca, Stéphane Lallich, Nguyen-Khang Pham*

**Intensive Use of Correspondence Analysis for Large Scale Content-Based Image Retrieval** ..... 57
*Nguyen-Khang Pham, Annie Morin, Patrick Gros, Quyet-Thang Le*

**Toward a Better Integration of Spatial Relations in Learning with Graphical Models** ..... 77
*Emanuel Aldea, Isabelle Bloch*

**Part II: Unsupervised Learning**

**Multigranular Manipulations for OLAP Querying** ..... 97
*Gilles Hubert, Olivier Teste*

**A New Approach for Unsupervised Classification in Image Segmentation** ..... 113
*Sébastien Lefèvre*

**Cluster-Dependent Feature Selection through a Weighted Learning Paradigm** ..... 133
*Nistor Grozavu, Younès Bennani, Mustapha Lebbah*

**Two Variants of the Okm for Overlapping Clustering** ..... 149
*Guillaume Cleuziou*

**A Stable Decomposition Algorithm for Dynamic Social Network Analysis** ..... 167
*Romain Bourqui, Paolo Simonetto, Fabien Jourdan*

**Part III: Security and Data Streaming**

**An Hybrid Data Stream Summarizing Approach by Sampling and Clustering** ..... 181
*Nesrine Gabsi, Fabrice Clérot, Georges Hébrail*

**SPAMS: A Novel Incremental Approach for Sequential Pattern Mining in Data Streams** ..... 201
*Lionel Vinceslas, Jean-Emile Symphor, Alban Mancheron, Pascal Poncelet*

**Mining Common Outliers for Intrusion Detection** ..... 217
*Goverdhan Singh, Florent Masseglia, Céline Fiot, Alice Marascu, Pascal Poncelet*

**Intrusion Detections in Collaborative Organizations by Preserving Privacy** ..... 235
*Nischal Verma, François Trousset, Pascal Poncelet, Florent Masseglia*

**Part IV: Ontologies and Semantic**

**Alignment-Based Partitioning of Large-Scale Ontologies** ..... 251
*Fayçal Hamdi, Brigitte Safar, Chantal Reynaud, Haïfa Zargayouna*

**Learning Ontologies with Deep Class Hierarchies by Mining the Content of Relational Databases** ..... 271
*Farid Cerbah*

**Semantic Analysis for the Geospatial Semantic Web** ..... 287
*Alina Dia Miron, Jérôme Gensel, Marlène Villanova-Oliver*

**Statistically Valid Links and Anti-links Between Words and Between Documents: Applying TourneBool Randomization Test to a Reuters Collection** ..... 307
*Alain Lelu, Martine Cadot*

**List of Contributors** ..... 325

**Author Index** ..... 337

# Part I
# Supervised Learning and Information Retrieval

# Discrepancy Analysis of Complex Objects Using Dissimilarities

Matthias Studer, Gilbert Ritschard, Alexis Gabadinho, and Nicolas S. Müller

**Abstract.** In this article we consider objects for which we have a matrix of dissimilarities and we are interested in their links with covariates. We focus on state sequences for which pairwise dissimilarities are given for instance by edit distances. The methods discussed apply however to any kind of objects and measures of dissimilarities. We start with a generalization of the analysis of variance (ANOVA) to assess the link of complex objects (e.g. sequences) with a given categorical variable. The trick is to show that discrepancy among objects can be derived from the sole pairwise dissimilarities, which permits then to identify factors that most reduce this discrepancy. We present a general statistical test and introduce an original way of rendering the results for state sequences. We then generalize the method to the case with more than one factor and discuss its advantages and limitations especially regarding interpretation. Finally, we introduce a new tree method for analyzing discrepancy of complex objects that exploits the former test as splitting criterion. We demonstrate the scope of the methods presented through a study of the factors that most discriminate Swiss occupational trajectories. All methods presented are freely accessible in our TraMineR package for the R statistical environment.

**Keywords:** Distance, Dissimilarities, Analysis of Variance, Decision Tree, Tree Structured ANOVA, State Sequence, Optimal Matching.

## 1 Introduction

The analysis of dissimilarities is used in a wide range of areas. It includes biology with the analysis of genes and proteins (sequence alignment), ecology with the comparison of ecosystems, sociology, network analysis where similarity is a central

Matthias Studer · Gilbert Ritschard · Alexis Gabadinho · Nicolas S. Müller
Department of Econometrics and Laboratory of Demography, University of Geneva, Switzerland
e-mail: matthias.studer@unige.ch

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 3–19.
springerlink.com 

notion or the automatic analysis of texts to name just a few. When analyzed objects are not directly measurable or complex, such as sequences or ecosystems for instance, it may be convenient to think in terms of dissimilarities between objects. Having such dissimilarities, it is customary to perform a cluster analysis to get a reduced number of groups for facilitating interpretation. Once the groups are identified, it is common practice to measure the relationship between these objects and other variables of interest by using, for instance, association test or logistic regression.

However, by focusing on clusters we loose indeed information, which may lead to unfair conclusions, particularly for borderline objects. Similarly, it is possible that some associations become less significant through this reduction of information. The latter is not controlled and grouping choices, usually made on statistical ground, may hide others alternatives that might show more interesting associations with some explanatory factors.

In this article we present a set of methods to analyze dissimilarities directly, i.e. without any prior clustering. They will allow us to measure the relationship between, on the first hand, one or more covariates and, secondly, objects described using dissimilarities. We begin by studying the link with a single variable building on the test introduced by Anderson (2001). We extend then the analysis by introducing a new test of the homogeneity of object discrepancy and propose, for the case of state sequences, a new way to display the results. As a second step, we present the method from McArdle and Anderson (2001) which enables us to include several variables at the same time. Finally, we introduce a method based on induction trees that leads to a better interpretation of the results. The method is similar to the one presented in Geurts *et al.* (2006) but is more general since it is not limited to distances that can be expressed as kernels. The criteria is also similar to the one used by Piccarreta and Billari (2007) in an unsupervised setting. Finally, we give a short overview on how to perform the presented methods in R by means of TraMineR. The scope of the discussed methods is illustrated throughout the article by applying them on occupational trajectory data.

## 2 The Illustrative Data Set

Let us start with a short application issue that will serve as illustration throughout this article. We consider the study of occupational trajectories and expose the problematic so that examples and their interpretations will be clearer for the reader. We are interested in the construction of professional trajectories and factors that may influence it. We focus on the study of working rates following the work of Levy *et al.* (2006). We know that, while men's trajectories are relatively homogeneous and exhibit three main phases, namely "education", "full time work" and "retirement", those of women are much more varied. Thus, their average curve of working rates has a camel shape with a decrease in working rate when children are very young

and a recovery thereafter. This average curve results however from very distinct trajectories. Some women stop working completely or reduce their working rates and then some of them return to work while others do not. In addition, some women go back and forth between work and at home activity.

Besides the effects of sex on the trajectories, we are interested in testing the differences in trajectories between generations (2 categories), family types — number of children (4 cat.) and marital status (4 cat.) — and socio-economic situations — father social status (10 cat.), income (4 cat.) and education (3 cat.). We are also interested to test whether trajectories of younger generations are significantly more diverse than those of older ones, and thus show a pluralization of trajectories.

To answer these questions, we use the data from the biographical retrospective survey conducted by the Swiss Household Panel[1] in 2002. We know, for each individual and every year, his occupational situation distinguishing between the following states: full time work, part-time work, negative break (eg., unemployment), positive break (eg., travel), at home and training. We focus on the period between ages 25 and 40 which is the key period regarding professional career deployment. We retain all cases without missing data, that is 1560 trajectories. Since all retained individuals are aged 40 at the survey time they are all born before 1962.

## 3 Measuring Association Using Dissimilarities

We now present a method based on the ANOVA principle to evaluate the association between, on the one hand, objects characterized by a matrix of dissimilarities and, secondly, a categorical variable. We take as a starting point the method introduced by Anderson (2001) for analyzing ecosystems. We retain the more geometric approach of Batagelj (1988) in its generalization of the Ward criterion. Finally, we apply these methods on our example.

### *3.1 General Principles*

Following the ANOVA principles, we seek to determine the part of the variance that is "explained" by a given partition. The ANOVA is based on the notion of "sum of squares" that is the sum of the squared Euclidian distances between each value and the mean. This sum of squares, or inertia, can also be expressed as the average of the pairwise squared Euclidian distances ($d^2_{e,ij}$). These relationships are formalized by Eq. (1).

$$SS = \sum_{i=1}^{n}(y_i - \bar{y})^2 = \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}(y_i - y_j)^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i+1}^{n} d^2_{e,ij} \tag{1}$$

[1] http://www.swisspanel.ch

The concept of sum of squares can be generalized to other dissimilarity measures in two alternative ways. Anderson (2001) proposes to replace the Euclidian distance $d_{e,ij}$ in Eq. (1) with any possibly non-Euclidian measure of dissimilarity $d_{ij}$ yielding:

$$SS^{**} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i+1}^{n} d_{ij}^2 \tag{2}$$

However, we prefer to substitute the non-Euclidean dissimilarity $d_{ij}$ for the squared Euclidean distance $d_{e,ij}^2$ rather than for the distance itself as proposed by Batagelj (1988). We argue shortly for this choice in Sec. 3.2 below. The retained generalization of $SS$ reads thus:

$$SS^{*} = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=i+1}^{n} d_{ij} \tag{3}$$

We use this expression for measuring the discrepancy of our complex objects. Indeed, using $SS = SS^*$ in the definition $s^2 = \frac{1}{n}SS$ of the sample variance we get a fairly intuitive measure of the object discrepancy. Since the variance is theoretically defined for Euclidean distances, we prefer the term "discrepancy" for this more general setting. Interestingly, the discrepancy $s^2$ is equal to half the average pairwise dissimilarity, that is:

$$s^2 = \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} \tag{4}$$

When generalizing the notion of sum of squares to non-Euclidean measures of dissimilarity, the Huygens theorem, Eq. (5), that states that the total sum of squares ($SS_T$) is the between sum of squares ($SS_B$) plus the residual within sum of squares ($SS_W$) remains valid (Batagelj, 1988).

$$SS_T = SS_B + SS_W \tag{5}$$

We can thus apply the analysis of variance (ANOVA) machinery to our complex objects.

The terms in Eq. (5) can all be derived from formula (3). The total sum of squares ($SS_T$) and the within sum of squares ($SS_W$) are computed directly with formula (3), $SS_W$ being simply the sum of the within sums of squares of each subgroup. The between sum of squares $SS_B$ is then obtained by taking the difference between the $SS_T$ and $SS_W$. Using Eq. (5) we can assess the share of discrepancy explained by a categorical or discretized continuous variable. In the spirit of ANOVA, this reduction of discrepancy is due to a difference in the positioning of the gravity centers (or centroids) of the classes. This interpretation holds for any kind of distance even though the concept of class center is not clearly defined for complex non numeric objects (Batagelj, 1988). It is likely that the gravity centers will not belong to the

object space, exactly as the mean of integer values may be a real non integer value. Hence, conceptually, we look for the part of the discrepancy that is explained by differences in group positioning and we measure this part with the $R^2$ formula (6). Alternatively, we may consider the $F$ that compares the explained discrepancy to the residual discrepancy. The $F$ formula is given in Eq. (7), where $n$ is the number of cases and $m$ the number of parameters.

$$R^2 = \frac{SS_B}{SS_T} \tag{6}$$

$$F = \frac{SS_B/(m-1)}{SS_W/(n-m)} \tag{7}$$

The statistical significance of the association, i.e. of the explained part of discrepancy cannot be assessed with the $F$ test as in classical ANOVA. Indeed, the $F$ statistic (7) does not follow a Fisher distribution with our complex objects for which the normality assumption is hardly defendable. We consider therefore a permutation test (Anderson, 2001; Moore *et al.*, 2003). This test works as follows. At each step we change the complex object assigned to each case by means of a randomly chosen permutation, which is equivalent to jointly permute the content of the rows and columns of the distance matrix. We thus get a $F_{perm}$ value for each permutation. Repeating this operation $p$ times we end up with an empirical non parametric distribution of $F$ that characterizes its distribution under independence, i.e. assuming the objects are assigned to the cases independently of their profile in terms of explanatory factors. From this distribution, we can assess the significance of the observed $F_{obs}$ statistic by evaluating the proportion of $F_{perm}$ that are higher than $F_{obs}$. It is generally admitted that 5000 permutations are necessary to assess a significance threshold of 1% and 1000 for a threshold of 5%.

## *3.2 Generalization Conditions*

As mentioned above, we can generalize Eq. (1) either by substituting the dissimilarity $d$ for the Euclidean distance $d_e$ or for its square $d_e^2$. In this subsection, we justify our preference for the latter solution, i.e. equation (3). Firstly, in the Euclidian case, the second equality in Eq. (1) which links the sum of deviations to the mean to the sum of pairwise differences follows from properties of signed deviances and pairwise differences which do not hold for unsigned distances. Secondly, with this choice, the non negativity of the contribution of any object to the total discrepancy automatically results when the dissimilarity satisfies the triangle inequality.

In the Euclidian case, the equality (1) can be established by showing first the following result (Späth, 1975):

$$\sum_{i=1}^{n}(y_i - x)^2 = \sum_{i=1}^{n}(y_i - \bar{y})^2 + n(\bar{y} - x)^2 \tag{8}$$

Indeed, we have:

$$y_i - x = (y_i - \bar{y}) + (\bar{y} - x) \tag{9}$$

$$(y_i - x)^2 = (y_i - \bar{y})^2 + 2(y_i - \bar{y})(\bar{y} - x) + (\bar{y} - x)^2$$

$$\sum_{i=1}^{n}(y_i - x)^2 = \sum_{i=1}^{n}(y_i - \bar{y})^2 + n(\bar{y} - x)^2 + 2\sum_{i=1}^{n}(y_i - \bar{y})(\bar{y} - x) \tag{10}$$

Since, $\sum_{i=1}^{n}(y_i - \bar{y}) = 0$, the last term in (10) vanishes which yields Eq. (8). The equality (1) results then by setting $x = y_j$ in (8) and summing over $j = 1, \cdots, n$.

Clearly, equality (9) does not hold if we replace differences $y_i - x$, $y_i - \bar{y}$ and $\bar{y} - x$ with non negative dissimilarities. Likewise, the last term in (10) would not vanish with non negative dissimilarities. Using the second solution (3), we do not have to care about the deviation between objects. We just postulate that there exists a signed deviation measure in the object space.

We now turn to our second argument regarding the contribution of an object $x$ to the total discrepancy. This contribution $d_{x\tilde{g}}$ can be seen as the dissimilarity between $x$ and its (possibly virtual) gravity center $\tilde{g}$. Using the same scheme (3) of generalization, it can be obtained by substituting $d_{x\tilde{g}}$ to $(\bar{y} - x)^2$ in Eq. (8) and by isolating this term, which yields (Batagelj, 1988):

$$\begin{aligned} d_{x\tilde{g}} &= \frac{1}{n}\Big(\sum_{i=1}^{n} d_{xi} - SS\Big) \\ &= \frac{1}{2n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\big(2 \cdot d_{ix} - d_{ij}\big) \end{aligned} \tag{11}$$

This contribution to the discrepancy is non negative when the dissimilarity measure respects the triangle inequality. Indeed, according to Eq. (11), $d_{x\tilde{g}}$ is minimal when each $d_{ij}$ is maximal. Under the triangle inequality $d_{ij}$ cannot exceed $d_{xi} + d_{xj}$ and hence, $d_{x\tilde{g}}$ reaches its minimum when $d_{ij} = d_{xi} + d_{xj}$ for all $i$ and $j$. This minimum is zero which implies $d_{x\tilde{g}} \geq 0$. The non negativity of the contribution of $x$ cannot be deduced from the triangle inequality property of the dissimilarity if we use definition (2) of $SS$, i.e. if we replace the squared Euclidean distance with the squared similarity.

With the retained approach, negative contributions to the discrepancy can occur with semi-metric dissimilarities, that is when the triangle inequality does not hold. The "dissimilarity" $d_{x\tilde{g}}$ becomes negative when adding $x$ reduces the discrepancy between the other objects. This can be the case when the distance between two objects, say $y$ and $z$, becomes shorter when we can pass through $x$, i.e. when $d_{yz} > d_{yx} + d_{xz}$. Such situation is quite usual in social network analysis. For instance, let us consider a social network between $x$, $y$ and $z$ where the dissimilarity is equal to 1 for two people that meet often and is equal to 10 when they never meet. The dissimilarity $d_{x\tilde{g}}$ would then be negative if $x$ often meets $y$ and $z$ while $y$ never meets $z$. From a social network perspective, we would say that $x$ plays a cohesive role in the network.

Though a negative contribution to the discrepancy makes sense for social networks, it is not the case for most applications. Hence, the results should be

interpreted with caution when the dissimilarity measure is only semi-metric. In particular, one should be ready to admit and give sense to negative contributions to the discrepancy.

### 3.3 Application

We now illustrate the proposed test on our example data about the study of occupational trajectories. We use optimal matching (OM) for measuring the dissimilarities between trajectories that are indeed represented as state sequences. The OM dissimilarity, also known as the edit distance, is the minimal cost of transforming one sequence into the other using two types of transformation operations, namely indel (insert or delete) and substitution of elements. The transformation cost is determined by assigning indel and substitution costs. For our example, we computed the OM distances with an indel cost set to 1 and substitution costs at 2. Notice that the OM dissimilarity respects the triangle inequality. Indeed, dissimilarity being the minimal cost for transforming a sequence $y$ into $z$, we necessarily have $d_{yz} \leq d_{yx} + d_{xz}$.

The discrepancy of the occupational trajectories of the whole data set is 0.501 which is equal to half of the average edit distance (1.02). It is 0.118 for men and 0.614 for women indicating that women's trajectories exhibit wider variety.

Table 1 summarizes the results of the discrepancy analysis for the whole population as well as for men and women separately. In each case we considered individually each of the available predictive factors. The p-values of the tests are based on 1000 permutations.

**Table 1** Association test with occupational trajectories

| | Total | | | Men | | | Women | | |
|---|---|---|---|---|---|---|---|---|---|
| Variable | $F$ | $R^2$ | Sig | $F$ | $R^2$ | Sig | $F$ | $R^2$ | Sig |
| Sex | 477.995 | 0.235 | **0.000** | | | | | | |
| Father soc. status | 1.578 | 0.009 | **0.029** | 2.085 | 0.026 | **0.005** | 1.205 | 0.013 | 0.163 |
| Income | 1.349 | 0.003 | 0.182 | 3.086 | 0.013 | **0.006** | 3.553 | 0.013 | **0.000** |
| Education | 18.486 | 0.023 | **0.000** | 20.632 | 0.054 | **0.000** | 6.287 | 0.015 | **0.000** |
| Cohort | 17.037 | 0.011 | **0.000** | 6.330 | 0.009 | **0.001** | 14.911 | 0.018 | **0.000** |
| Children | 13.704 | 0.026 | **0.000** | 1.006 | 0.004 | 0.391 | 25.740 | 0.085 | **0.000** |
| Marital status | 9.744 | 0.018 | **0.000** | 1.783 | 0.007 | **0.047** | 18.078 | 0.061 | **0.000** |

Not surprisingly, sex explains the biggest part of the discrepancy of trajectories with a $R^2$ that reaches 0.235. In other words, the sex variable explains 23.5% of the discrepancy. The relationship is statistically significant since the $F_{obs} = 477.995$ was never attained amongst the thousand permutations. As for the other covariates, results show that the Father's social status and Education impact primarly male trajectories while women's trajectories are more strongly influenced by familial factors such as the number of children and the marital status. than female trajectories.

In summary, these first results show that the occupational trajectory is significantly influenced by most of the considered predictive variables. From the high significance of the significance tests, differences in the positioning of the gravity centers of groups of sequences clearly exist. Nevertheless, it is difficult to understand and interpret these differences at this stage.

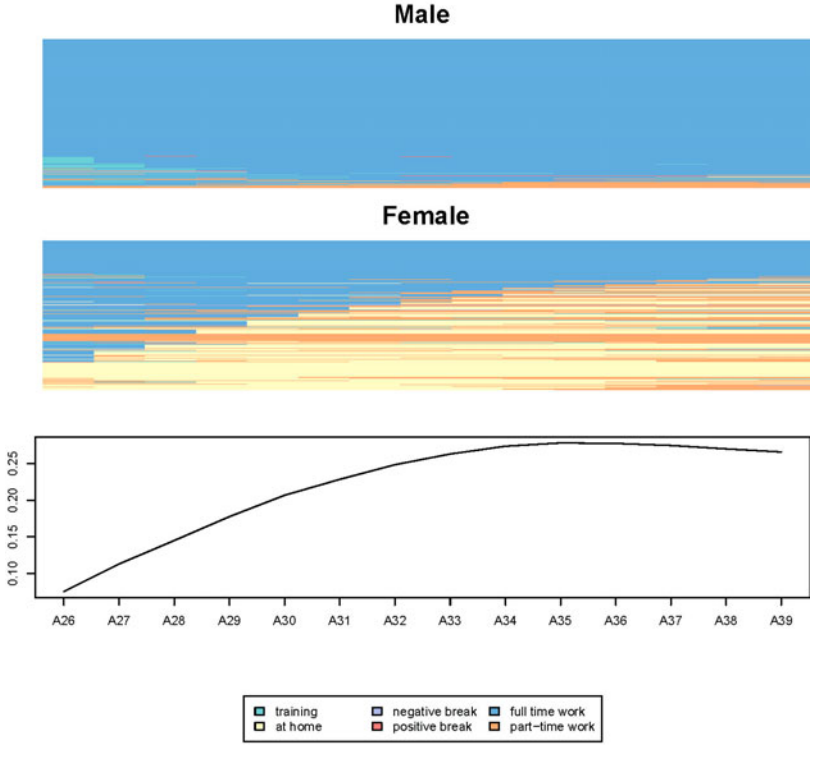



**Fig. 1** Differences of trajectories according to sex

Figure 1, which presents a new way of displaying the differences between groups of sequences, should help interpretation. The first two charts show men and women trajectories using index-plots (Scherer, 2001). In these figures, each sequence is represented by a time line split into segments colored according to the corresponding occupational state.

To improve readability of the index-plots, we ordered the sequences according to the first dimension of a PCA (Principal Coordinate Analysis) (Gower, 1966). If ordering sequences by an underlying dimension facilitates the interpretation of the index-plot, the plots provide conversely useful information for interpreting the PCA axis. For instance, we observe in our case that the sequences are organized in a

continuum ranging from full-time trajectories to trajectories where we stay at home during the whole sequence. The axis can thus be read as a Full-time - At home axis.

The final chart exhibits the evolution of the strength of association between the categorical covariate and a sliding two period long sub-sequence of the trajectory. For each unit of time, we extracted a sub-sequence of two consecutive states for which we calculated the distance matrix and the share of discrepancy explained by the covariate. This representation helps at identifying the periods over which the sequences are most differentiated by gender. It appears that gender differences reach their peak around 35 years old.

## 4 Homogeneity of Discrepancy

In some situations, it may be of interest to test whether the discrepancies within the groups differ significantly. From a geometric point of view, we are interested in measuring differences in the diameter of the distribution of sequences within each group. In classical analysis of variance, we could use a Bartlett's test (Snedecor and Cochran, 1989) that supposes equal variances under $H_0$ or, in other words, the homogeneity of variances. This test is based on the statistical distribution of the statistic $T$ defined by Eq. (12), where $s_i^2$ stands for the discrepancy within group $i$. All terms in this equation can be calculated with the formulas already introduced. As for the $F$, it is not possible in our non-Gaussian case to assume that this statistic $T$ has a known distribution. We use therefore again permutation tests to assess the significance of differences in discrepancy.

$$T = \frac{(n-m)\ln\left(\sum_{i=1}^{m}\frac{(n_i-1)}{(n-m)}s_i^2\right) - \sum_{i=1}^{m}(n_i-1)\ln(s_i^2)}{1+\frac{1}{3(m-1)}\left[\sum_{i=1}^{m}\frac{1}{n_i-1}-\frac{1}{n-m}\right]} \tag{12}$$

In the previous section, we found that men's discrepancy is 0.118 against 0.614 for women. This relatively high difference is confirmed by the $T_{obs}$ which is 460.017, a value that was attained by none of the thousand permutations. This allows us to state that the discrepancies differ significantly with the sex of the respondent. More interestingly from a sociological point of view, the discrepancy of the people born after 1945 is significantly higher than those born earlier. We thus have clear evidence that the diversity of occupational trajectories increased for younger generations.

## 5 Multi-factor Discrepancy Analysis

In Sec. 3.3 we examined the bivariate association between the trajectory and each of the covariates considered independently. We consider here the generalization to the multi-factor case and adopt for that the framework of the general multivariate analysis of variance. Several authors have considered such analyses

from pairwise distances (Excoffier *et al.*, 1992; Gower and Krzanowski, 1999; Anderson, 2001; Zapala and Schork, 2006). We adopt the approach and formalism of McArdle and Anderson (2001) who conducted a multi-factor analysis of ecosystems on the bases the pairwise semi-metric distance of Bray-Curtis. However, as for the simple discrepancy analysis and unlike McArdle and Anderson (2001) we substitute the pairwise dissimilarity measure for the squared Euclidean distance rather than for the distance itself.

Formally, we consider the multivariate regression model: $\mathbf{Y} = \mathbf{X}\beta + \varepsilon$, where $\mathbf{Y}$ is the $n \times t$ matrix with $n$ observed values of $t$ response variables and $\mathbf{X}$ the $n \times m$ matrix with the values of $m$ predictors including a first column of ones corresponding to the constant.

In the Euclidean case, the sum over the $t$ response variables of their sums of squares can be derived by means of the same Gower matrix as that used in PCA (Gower, 1966). Similarly to McArdle and Anderson (2001), we generalize this analysis to any type of dissimilarities. Let $\mathbf{1}$ be a vector of ones of length $n$, $\mathbf{I}$ the identity matrix and $\mathbf{A}$ a matrix with generic element $a_{ij} = -\frac{1}{2}d_{ij}$, where $d_{ij}$ is the dissimilarity between cases $i$ and $j$, which we substitute for the squared Euclidean distance in the original Gower's formulation. The Gower matrix reads as follows

$$\mathbf{G} = \left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'\right)\mathbf{A}\left(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}'\right) \tag{13}$$

with in our case a matrix $\mathbf{A}$ that results from the available pairwise dissimilarities. The total sum of squares $SS_T$ is equal to the trace of $\mathbf{G}$. McArdle and Anderson (2001) show that the explained sum of squares $SS_B$ and the residual sum of squares $SS_W$ can be written as

$$SS_B = tr\big(\mathbf{HGH}\big) \tag{14}$$

$$SS_W = tr\big[(\mathbf{I}-\mathbf{H})\mathbf{G}(\mathbf{I}-\mathbf{H})\big] \tag{15}$$

where $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$ is the idempotent matrix usually known as "hat" matrix in linear regression. Using these two quantities we can derive a global pseudo-$R^2$ and a global pseudo-$F$ statistic by applying Eqs. (6) and (7). Formula (14) and (15), however, allow us to account of any number of covariates and specifically of categorical factors through their contrast or indicator coding.

As in the single discrepancy analysis, the $F$ distribution is not relevant for the pseudo-$F$ and we consider again permutations tests for assessing its significance.

We may also consider the contribution of each covariate to the total discrepancy reduction. As with multi-factor ANOVA there are different ways of looking at these individual contributions. Shaw and Mitchell-Olds (1993) distinguish for instance a Type I and a Type II method. Type I is incremental. Covariates are successively added to the model and the contribution of each covariate is measured by the $SS_B$ increase that results when it is introduced. With this method the measured impact of each factor depends on the order in which they are introduced. With Type II, known to be robust in the absence of interaction effects, the contribution of each covariate is measured by the reduction of $SS_B$ that occurs when we drop it out from the full

model, i.e. from the model with all covariates. We retain this second method and hence compute the following $F$ for each covariate $v$

$$F_v = \frac{(SS_{B_c} - SS_{B_v})/p}{SS_{W_c}/(n-m-1)} \tag{16}$$

where the $SS_{B_c}$ and $SS_{W_c}$ are the explained and residual sums of squares of the full model, $SS_{B_v}$ the explained sum of squares of the model after removing variable $v$, and $p$ the number of indicators or contrasts used to encode the covariate $v$.

Let us look at what this gives for our illustrative example. Table 2 shows the results for two models, the complete model with all variables and a model obtained after removing non significant covariates through a backward stepwise process.

**Table 2** Multi-Factor Discrepancy Analysis

| | Full Model | | | Backward Model | | |
|---|---|---|---|---|---|---|
| Variable | $F_v$ | $\Delta R^2_v$ | Sig | $F_v$ | $\Delta R^2_v$ | Sig |
| Sex | 477.196 | 0.218 | 0.000 | 488.627 | 0.224 | 0.000 |
| Education | 8.230 | 0.008 | 0.000 | 10.986 | 0.010 | 0.000 |
| Income | 0.868 | 0.001 | 0.542 | | | |
| Father's soc. status | 1.167 | 0.005 | 0.241 | | | |
| Cohort | 11.586 | 0.005 | 0.000 | 13.670 | 0.006 | 0.000 |
| Children | 9.887 | 0.014 | 0.000 | 10.313 | 0.014 | 0.000 |
| Marital status | 4.621 | 0.006 | 0.000 | 5.073 | 0.007 | 0.000 |
| | $F_{\text{tot}}$ | $R^2_{\text{tot}}$ | Sig | $F_{\text{tot}}$ | $R^2_{\text{tot}}$ | Sig |
| Global | 29.557 | 0.297 | 0.000 | 63.602 | 0.291 | 0.000 |

From the global statistics, the set of covariates provide overall significant information about the diversity of occupational trajectories.

In the full model, the sex remains the most significant covariate. If we remove this variable, the $R^2$ of the model ($= 0.297$) decreases by 0.218. This difference is significant since we have $F_{sex} = 477.196$, a value never attained with a thousand permutations. On the contrary, the income is for instance not significant. Removing it from the model reduces the $R^2$ by only 0.001 and results in a $F_{income}$ value of 0.868, which was exceeded for $0.542 \cdot 1000 = 542$ of the thousand permutations. Likewise, the father's social status loses its significance in the multi-factor case. Indeed, it becomes non-significant as soon as we control for the education level, these two variables being strongly correlated and education being more significant.

The multi-factor approach provides information about the proper effect of the covariates on the occupational trajectory, that is the part of the its total effect that is not accounted for by already introduced factors. It is in that sense complementary to the single univariate discrepancy analysis that informs on the raw effect of each covariate. Nevertheless, while the method permits us to know which effects

are significant, it does not tell us much about what the effects are, i.e. about how occupational trajectories may change with the value of the covariates. We propose for that a tree approach which can be seen as an extension of the graphical display shown in Fig. 1.

## 6 Tree Structured Analysis

This section introduces a new method based on the principle of induction trees for analyzing the discrepancy of objects described by a dissimilarity matrix. Induction trees work as follows (Breiman *et al.*, 1984; Kass, 1980). They start with all individuals grouped in an initial node. Then, they recursively partition each node using values of a predictor. At each node, the predictor and the split are chosen in such a way that the resulting child nodes differ as much as possible from one another or have, more or less equivalently, lowest within discrepancy. The process is repeated on each new node until some stopping criterion is reached.

Recursive partitioning is known to provide an easily comprehensible view of how each newly selected covariate nuances the effect of covariates introduced at earlier levels. This requires indeed to display suitable information about the distribution in each node. We could represent the centrotype, i.e. the observed object that minimizes the dissimilarity (11) with the group gravity center. It would be more instructive to also render the within group discrepancy. Though this is not obvious for any kind of complex objects, displaying index-plots as those used in Fig. 1 provides a good solution for state sequences.

Beside the displayed node content, the originality of our approach resides in the use of a splitting criterion derived from the pairwise dissimilarities, namely the univariate pseudo-$R^2$ that we described in Sec. 3. We select thus at each node the predictor and binary split for which get the highest pseudo-$R^2$, i.e. the split that accounts for the greatest part of the object discrepancy. An alternative would be to use the significance of the univariate pseudo-$F$. However, since this significance must be determined through permutation tests we would end-up with an excessive time complexity if we had to repeat it for each predictor and possible split. We consider therefore the $F$ significance only as a stopping criteria, i.e. we stop growing a branch when we get a non-significant $F$ for the selected split. This requires to run the permutations only once at each node, which remains tractable.

Using the pseudo-$R^2$ as splitting criterion condemns us to build binary trees. Indeed, the $R^2$ does not penalize for the number of groups and would hence always select the maximal number of groups if we allowed n-ary splits. The $R^2$ adjusted for the number of groups as it is used in multiple regression would not be a satisfactory solution since it is known to insufficiently penalize complexity. On the other hand, information criteria such as the BIC seem hardly derivable in our setting where we do not know the distribution of our statistics ($R^2$, $F$ or $SS_W$) under the independence hypothesis.

It is worth mentioning that our tree building procedure resembles that proposed in Geurts *et al.* (2006). However, our formulation is more general since it works

with any kind of metric and non metric dissimilarities, while Geurts *et al.* (2006)'s solution is restricted to dissimilarities that can be derived through the kernel trick. For growing a tree from semi-metric dissimilarities we should indeed be ready to accept and give sense to possible negative contributions to the variance.

Before looking at the example, let us add a few words about computational aspects. First, we can highlight that it is not necessary to recompute $SS_W$ from scratch for each possible binary split that can be derived from a same predictor. Our algorithm makes use of partial results first collected into a symmetric $m \times m$ matrix $\mathbf{E}$, where $m$ is the number of different observed values of the predictor. Each element $e_{k\ell}$ of $\mathbf{E}$ is defined as $e_{k\ell} = \sum_{i \in k} \sum_{j \in \ell} d_{ij}$, that is as the sum of dissimilarities between on the one hand, cases that take the $k$-th value of the predictor and, on the other hand, those that take the $\ell$-th value. The residual sum of squares for a group of values $G$ is then equal to $SS_{G,res} = \frac{1}{n_G}(\sum_{k \in G} \sum_{\ell \geq k, \ell \in G} e_{k\ell})$. Reusing this way the same partial sums of dissimilarities may save a great amount of computation time especially for categorical predictors with few different values.

Secondly, we may exploit the fact that the $R^2$ can only decrease when merging categories. From matrix $\mathbf{E}$ we can compute the $R^2_{ori}$ that measures the part of discrepancy explained by the predictor in its original form, i.e. with all its distinct values. It then follows that this $R^2_{ori}$ is an upper bound for the best $R^2$ that would result from a binary split based on the considered predictor. Hence, when the $R^2_{ori}$ of the current predictor does not exceed the $R^2$ of the previously found best split, it becomes unnecessary to test the splits for the current predictor.

The global quality of the tree can be assessed through the association strength between the objects and the leaf (terminal node) membership. The global multi-factor pseudo-$F$ gives us a way of testing the statistical significance of the obtained segmentation and the global pseudo-$R^2$ the part of the total discrepancy that is explained by the tree.

Figure 2 shows the dissimilarity tree grown for our example of occupational trajectories. The used stopping criteria are a $p$-value of 1% for the $F$ test, a minimal leaf size of 100 and a maximal depth of 5. In each node we see the plot of the individual sequences as well as the node size and the discrepancy within the node (var). At the bottom of each parent node we indicate the retained split predictor with the associated $R^2$ while the definition of the binary split may be inferred from the indication at the top of the child nodes.

The overall tree $R^2$ is 0.302, which is higher than for the models in Table 2. The tree has thus a better explanatory power. We get this higher value by retaining only 4 predictors against 5 for the backward model. This may be explained by interaction effects that the tree automatically accounts for and that were not considered in the multi-factor discrepancy analysis. We thus can point out here that birth cohort and number of children interact in their effect on female occupational trajectories while birth cohort interacts with education in their effect on men trajectories. This automatic detection of interaction is indeed a fundamental property of all induction trees.

By looking at the displayed individual sequences, we are now able to gain knowledge about what the effect of the predictors are. Clearly, men are characterized by

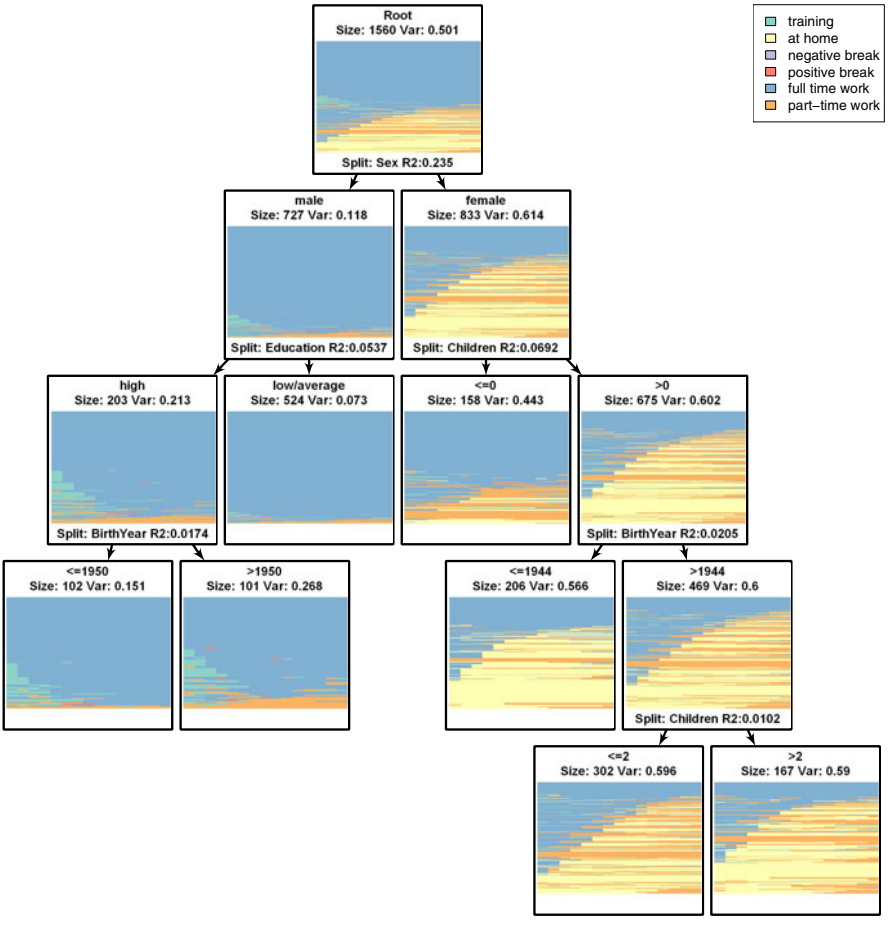



**Fig. 2** Regression tree based on pairwise dissimilarities between sequences

full time trajectories while part time and at home are typically found in women's trajectories. Among men, the choice of part-time seems to be related with higher education. For women, occupational trajectories are more diversified. Those who had at least one child have higher chance to experience part time work when they were born after 1945. This birth cohort effect is, however, less pronounced among those women who had more than two children.

## 7 Discrepancy Analysis in R with TraMineR

The methods presented in this article are all implemented in TraMineR (Gabadinho *et al.*, 2009) our free package for the R statistical environment (R Development Core Team, 2008). We shortly show here how simple it is to use them. Assume that we

have the following R objects defined in our environment: *dm* a matrix of dissimilarities between cases, *mydata* a *data.frame* with the covariates and *mysequences* an object containing the state sequences.

Univariate discrepancy analysis and test for homogeneity of discrepancy is performed by calling the *dissassoc* function. This function takes three arguments: a dissimilarity matrix, a factor and the number of permutation ($R = 1000$ by default). The results presented in Sec. 3 were obtained with the following code:

```
R> dissassoc(dm, group = mydata$sex, R = 1000)
```

Likewise, we generated the bottom part of Fig. 1 by means of function *seqdiff* with the code below.

```
R> mysequences.diff <- seqdiff(mysequences, group = mydata$sex)
R> plot(mysequences.diff)
```

The multi-factor results given in Table 2 were obtained with the *dissmfac* function. The model is specified with a classical R formula in which the left hand side is the dissimilarity matrix. The *data* argument specifies the *data.frame* containing the covariates.

```
R> dissmfac(
+  dm ~ sex + cohort + education + fathsoc + income + children + marital,
+  data = mydata, R = 1000)
```

Tree structured analysis of dissimilarities is carried out with the *disstree* function. The dissimilarity matrix and the predictors are passed to the function in the same way as in *dissmfac*. Stopping criteria can be set with the following arguments: *minSize* for the minimum node size, *maxdepth* for the maximum tree depth and *pval* for the minimum required *p*-value. The *R* option permits to control the number of permutations used for computing the significance.

```
R> mytree <- disstree(
+  dm ~ sex + cohort + education + fathsoc + income + children + marital,
+  data = mydata, minSize = 100, maxdepth = 5,  R = 1000, pval = 0.01)
R> print(mytree)
```

The resulting tree can then be plotted by calling the *dot* program of GraphViz[2], which is an open source graph visualization software (Gansner and North, 1999). Assuming GraphViz is on the path, we get a tree similar to that of Fig. 2 but with density plots instead of the index-plots just with the steps below. The plot is generated in file *mytree.dot.svg*.

```
R> seqtree2dot(mytree, filename = "mytree", seqs = mysequences,
+  plottype = "seqdplot")
R> shell("dot -Tsvg -O mytree.dot")
```

[2] http://www.graphviz.org/

## 8 Conclusion

The aim of this article was to propose tools for investigating how complex objects characterized by their pairwise dissimilarities are related to covariates or predictive attributes. The methods proposed are inspired from the classical ANOVA framework. The basic trick consists in extending results that express the classical sum of squares $SS$ in terms of pairwise squared Euclidean distances to the case of any possibly non metric dissimilarity. We designate this general setting as discrepancy analysis. We proposed first a pseudo-$R^2$ and a pseudo-$F$ test for the univariate case in which each covariate is examined separately. For this same univariate case we discussed also a way of testing the homogeneity of the discrepancies among groups. We then discussed the multi-factor case where we assess the impact of a covariate by controlling for the effect of the other factors. Eventually, we introduced an original tree structured method for discrepancy analysis. For both the univariate and tree structured settings we considered also the question of depicting the effect of the covariates. The difficulty is here to find a suited way of representing the distribution of the objects. We showed that index-plots prove useful when objects are of state sequences. However, more general solutions that could be used for any type of objects would here be necessary and we are presently working on that.

The work presented leaves certainly place to improvements on several aspects. For instance, we plan to further explore alternatives to the $R^2$ splitting criteria used in dissimilarity trees. We are looking for a way to use $p$-values of pseudo-$F$ statistics and for a penalized criteria that would permit $n$-ary splits.

**Acknowledgements.** This work is part of the Swiss National Science Foundation research project FN-1000015-122230 “Mining Event Histories: Towards New Insights on Personal Swiss Life Courses”.

## References

Anderson, M.J.: A new method for non-parametric multivariate analysis of variance. Austral Ecology 26, 32–46 (2001)

Batagelj, V.: Generalized Ward and related clustering problems. In: Bock, H. (ed.) Classification and related methods of data analysis, pp. 67–74. North-Holland, Amsterdam (1988)

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification And Regression Trees. Chapman and Hall, New York (1984)

Excoffier, L., Smouse, P.E., Quattro, J.M.: Analysis of Molecular Variance Inferred from Metric Distances among DNA Haplotypes: Application to Human Mitochondrial DNA Restriction Data. Genetics 131, 479–491 (1992)

Gabadinho, A., Ritschard, G., Studer, M., Müller, N.S.: Mining Sequence Data in R with the TraMineR package: A User’s Guide. Technical report, Department of Econometrics and Laboratory of Demography, University of Geneva, Geneva (2009), `http://mephisto.unige.ch/traminer/`

Gansner, E.R., North, S.C.: An Open Graph Visualization System and Its Applications to software engineering. Software - Practice and Experience 30, 1203–1233 (1999)

Geurts, P., Wehenkel, L., d'Alché Buc, F.: Kernelizing the output of tree-based methods. In: Cohen, W.W., Moore, A. (eds.) ICML. ACM International Conference Proceeding Series, vol. 148, pp. 345–352. ACM, New York (2006)

Gower, J.C.: Some Distance Properties of Latent Root and Vector Methods Used in Multivariate Analysis. Biometrika 53(3/4), 325–338 (1966), http://www.jstor.org/stable/2333639

Gower, J.C., Krzanowski, W.J.: Analysis of distance for structured multivariate data and extensions to multivariate analysis of variance. Journal of the Royal Statistical Society: Series C (Applied Statistics) 48(4), 505–519 (1999)

Kass, G.V.: An exploratory technique for investigating large quantities of categorical data. Applied Statistics 29(2), 119–127 (1980)

Levy, R., Gauthier, J.-A., Widmer, E.: Entre contraintes institutionnelle et domestique : les parcours de vie masculins et féminins en Suisse. Cahiers canadiens de sociologie 31(4), 461–489 (2006)

McArdle, B.H., Anderson, M.J.: Fitting Multivariate Models to Community Data: A Comment on Distance-Based Redundancy Analysis. Ecology 82(1), 290–297 (2001), http://www.jstor.org/stable/2680104

Moore, D.S., McCabe, G., Duckworth, W., Sclove, S.: Bootstrap Methods and Permutation Tests. In: The Practice of Business Statistics: Using Data for Decisions, W. H. Freeman, New York (2003)

Piccarreta, R., Billari, F.C.: Clustering work and family trajectories by using a divisive algorithm. Journal of the Royal Statistical Society A 170(4), 1061–1078 (2007)

R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2008) ISBN 3-900051-07-0, http://www.r-project.org

Scherer, S.: Early Career Patterns: A Comparison of Great Britain and West Germany. European Sociological Review 17(2), 119–144 (2001)

Shaw, R.G., Mitchell-Olds, T.: Anova for Unbalanced Data: An Overview. Ecology 74(6), 1638–1645 (1993), http://www.jstor.org/stable/1939922

Snedecor, G.W., Cochran, W.G.: Statistical methods, 8th edn. Iowa State University Press (1989)

Späth, H.: Cluster analyse algorithmen. R. Oldenbourg Verlag, München (1975)

Zapala, M.A., Schork, N.J.: Multivariate regression analysis of distance matrices for testing associations between gene expression patterns and related variables. Proceedings of the National Academy of Sciences of the United States of America 103(51), 19430–19435 (2006)

# A Bayes Evaluation Criterion for Decision Trees

Nicolas Voisine, Marc Boullé, and Carine Hue

**Abstract.** We present a new evaluation criterion for the induction of decision trees. We exploit a parameter-free Bayesian approach and propose an analytic formula for the evaluation of the posterior probability of a decision tree given the data. We thus transform the training problem into an optimization problem in the space of decision tree models, and search for the best tree, which is the maximum a posteriori (MAP) one. The optimization is performed using top-down heuristics with pre-pruning and post-pruning processes. Extensive experiments on 30 UCI datasets and on the 5 WCCI 2006 performance prediction challenge datasets show that our method obtains predictive performance similar to that of alternative state-of-the-art methods, with far simpler trees.

**Keywords:** Decision Tree, Bayesian Optimization, Minimum Description Length, Supervised Learning, Model Selection.

## 1 Introduction

Building decision trees from training data is a problem which has begun to be treated in 1963 by Morgan and Sonquist. The first method is a regression tree which predicts a numerical variable (Morgan and Sonquist, 1963). Following this seminal work, the decision trees and regression trees problem of building has long been popular in machine learning.

Decision tree is a predictive model of a categorical variable and regression tree is a predictive model of a numerical variable. We may refer to the overviews Kohavi and Quinlan (2002), Breiman *et al.* (1984); Murthy (1998); Breslow and Aha (1997) for more details about the main decision-tree methods.

Nicolas Voisine · Marc Boullé · Carine Hue
Orange Labs, 2, avenue Pierre Marzin Lannion, France
e-mail: {nicolas.voisine,marc.boulle,
carine.hue}@orange-ftgroup.com

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 21–38.
springerlink.com 

A decision tree is a classifier expressed as a hierarchical partition of the learning space. The partitions are represented by connected nodes. A node having children is called internal node and is defined by a segmentation rule. Other nodes are called leaves and represent a decision process by assigning the majority class to each instance of the node. The two first referenced algorithms are CHAID (Kass, 1980) and ID3 (Quinlan, 1986). However, the CART (Breiman *et al.*, 1984) and C4.5 (Quinlan, 1993) decision trees are the benchmark methods with the highest reported performance.

The induction of an optimal decision tree from a data set is NP-hard (Naumov, 1991). Thus, learning the optimal decision tree requires exhaustive search and is limited to very small data sets. As a result, heuristic methods are required to build decision trees. These methods could be divided into two groups: global and top-down. The last group has the academic preference and referenced decision trees use top-down heuristics.

There are two kinds of top-down decision trees. First ones are based on a pre-pruning procedure (cf. CHAID and ID3), partitioning at each level of the tree the training (sub) set into subsets according to a selected segmentation variable. The choice of the variable among all the variables is made according to a segmentation criterion which provides the best partition. The procedure starts at the root of the tree and stops at the terminal nodes (leaves) when the criterion can no more be improved. The choice of the variable, the number of segments and the definition of the segmentation characterize the process of segmentation. The segmentation for numerical variables is called discretization and the segmentation for categorical variables is called grouping. There is not usually a global criterion to optimize segmentation process; each node splitting is optimized regardless of the others. The main decision trees (ID3, CHAID, CART, and C4.5) exploit a criterion based on information theory or statistical decision theory for evaluating segmentation. For example, C4.5 uses the *gain ratio* measure based on entropy, CART uses the *Gini Index* based on information gain measure and CHAID uses the CHI-Squared statistic with a threshold to take the best decision. However, pre-pruning suffers from the horizon effect (Breiman *et al.*, 1984). The issue of pre-pruning algorithms is to stop the development of nodes until the decision tree is sufficiently accurate and to limit overfitness. Since Breiman work, new algorithms based on a post-pruning (CART, C4.5) have been studied. The construction of decision trees by post-pruning consists of two steps. The first step is to build a tree by continuing the process of segmentation as deep as possible, even if the tree overfits the data. The second step prunes the decision tree by removing nodes which minimize a pruning criterion. Learning time is longer than in the case of pre-pruning algorithms, but the performance of the tree is improved (cf. C4.5). Some pruning criteria are based on the estimated error rate of classification (C4.5). Other pruning criteria are based on a validation set (CART). Both approaches need to define heuristic parameters. A third, less-used approach exploits the principle of Minimum Description Length (Quinlan and Rivest, 1989; Wallace and Patrick, 1993).

Nowadays, decision trees are a mature class of models for which is just expected slight improvement of performance. Nevertheless, the reduction of the size of the trees and the automation of learning process are still important issues.

The decision tree performance mainly depends of the structure of trees. Too small trees obtain poor performance (Breiman *et al.*, 1984). Too large trees overfit the training dataset and their performances collapse on the test dataset. Improving decision trees requires better segmentation criterion and pruning criterion. The post-pruning methods are likely to select noisy variables which are not pruned in the pre-pruning step. This might be frequent in the case of large numbers of variables. The main issue of building decision trees is to select the best variables, to segment them correctly and to decide when to stop. The reference methods (C4.5, CART, ID3 and CHAID) use several parameters to learn their optimal tree : parameters for the choice of variables, discretization of numerical variables, grouping of categorical variables, and settings of the pruning criterion. None of these methods offers comprehensive and consistent criterion, taking into account the structure of the tree, selection of variables, segmentation and the performance of the tree. Wallace and Patrick as a result of the work of Rivest and Quinlan use a MDL approach to define a global pruning criterion taking into account the tree structure and the distribution of the classes in the leaves (Wallace and Patrick, 1993). Their lookahead algorithm pre-prunes decision trees by selecting sub-trees which minimize MDL criterion. This MDL criterion does not provide the best pruning, Quinlan and Rivest who had given the idea have not integrated in final C4.5 decision trees. MDL criteria have been exploited both as a selection criterion of segmentation variable and post-pruning criterion. However, MDL approach is a promising way with theoretical foundation to reduce the size of decision trees and to improve learning automation. But referenced MDL approaches remain incomplete, since they do not take into account all the trees parameters.

In this article, we propose a complete criterion by using a Bayesian approach according to a parsimony principle close to the Minimum Description Length approach. The aim is to transform the learning problem in a simple optimization process of one single parameter-free criterion. The MODL approach has already proved its interest in the selection of variables, the supervised discretization of numerical variables (Boullé, 2006), grouping of categorical variables (Boullé, 2005) and supervised classification model, with the Selective Naive Bayes (Boullé, 2007). Our goal is to develop a decision tree using the MODL approach, to evaluate and compare its performance with benchmark decision trees : J48(C4.5 (Quinlan, 1993)) and SimpleCART(CART (Breiman *et al.*, 1984)) from the WEKA software (Garner, 1995) which is an academic reference. The article is organized as follows. Section 2 summarizes the MODL approach in the case of supervised discretization. Section 3 describes the extension of this approach to decision trees. Section 4 describes optimization algorithms. Section 5 reports comparative evaluations of the method. Finally, section 6 concludes the article.

## 2 The MODL Approach

For the convenience of the reader, this Section summarizes the MODL approach in the case of supervised discretization of numerical variables (Boullé, 2006).

The objective of supervised discretization is to induce a list of intervals which partitions the numerical domain of a continuous input variable, while keeping the information relative to the output variable. A trade-off must be found between information quality (homogeneous intervals in regard to the output variable) and statistical quality (sufficient sample size in every interval to ensure generalization).

In the MODL approach, the discretization is turned into a model selection problem. First, a space of discretization models is defined. The parameters of a specific discretization model are the number of intervals, the bounds of the intervals and the frequencies of the output values in each interval. Then, a prior distribution is proposed on this model space. This prior exploits the hierarchy of the parameters: the number of intervals is first chosen, then the bounds of the intervals and finally the frequencies of the output values. The prior is uniform at each stage of the hierarchy. Finally, we assume that the multinomial distributions of the output values in each interval are independent from each other. A Bayesian approach is applied to select the best discretization model, which is found by maximizing the probability $p(Model|Data)$ of the model given the data. Using the Bayes rule and since the probability $p(Data)$ is constant under varying the model, this is equivalent to maximizing $p(Model)p(Data|Model)$.

Let $N$ be the number of instances, $J$ the number of output values, $I$ the number of input intervals. $N_i$ denotes the number of instances in the interval $i$ and $N_{ij}$ the number of instances of output value $j$ in the interval $i$. In the context of supervised classification, the number of instances $N$ and the number of classes $J$ are supposed to be known. A discretization model $M$ is then defined by the parameter set $\left\{I, \{N_i\}_{1\leq i\leq I}, \{N_{ij}\}_{1\leq i\leq I, 1\leq j\leq J}\right\}$.

Using the definition of the model space and its prior distribution, Bayes formula can be used to calculate the exact prior probabilities of the models and the probability of the data given a model. Taking the negative log of the probabilities, this provides the evaluation criterion given in Formula 1.

$$\log N + \log \binom{N+I-1}{I-1} + \sum_{i=1}^{I} \log \binom{N_i+J-1}{J-1} + \sum_{i=1}^{I} \log \frac{N_i!}{N_{i1}!N_{i2}!\ldots N_{iJ}!} \qquad (1)$$

The first term of the criterion stands for the choice of the number of intervals and the second term for the choice of the bounds of the intervals. The third term corresponds to the parameters of the multinomial distribution of the output values in each interval and the last term represents the conditional likelihood of the data given the model, using a multinomial term. Therefore "complex" models with large numbers of intervals are penalized.

Once the evaluation criterion is established, the problem is to design a search algorithm in order to find a discretization model that minimizes the criterion. In Boullé (2006), a standard greedy bottom-up heuristic is used to find a good discretization. In order to further improve the quality of the solution, the MODL algorithm performs post-optimizations based on hill-climbing search in the neighbourhood of a discretization. The neighbors of a discretization are defined with combinations of interval splits and interval merges. Overall, the time complexity of the algorithm is $O(JN\log N)$.

The MODL discretization method for supervised classification provides the most probable discretization given the data. Extensive comparative experiments report high performance (Boullé, 2006). The case of value grouping of categorical variables is treated in Boullé (2005) using a similar approach.

## 3 MODL Decision Trees

In this section, we apply the MODL approach to decision trees by defining explicitly a family of models and by introducing a global evaluation criterion of trees resulting from a Bayesian approach of model selection.

### *3.1 Definition*

A decision tree is a classification model which aims at predicting a categorical output variable from a set of numerical or categorical input variables. One advantage of decision trees is that they provide understandable models, based on decision rules. The issue is to induce a tree with high predictive performance while keeping its size as small as possible. This turns into a difficult problem of finding a trade-off between the performance of the model and the complexity of the structure of the tree, in order to ensure a good generalization of the model.

The MODL approach for decision trees consists in selecting the model with the highest probability given the data from a family of decision trees. As for the case of discretization (cf. Section 2), we apply a Bayesian approach to select the decision tree with the highest posterior probability $p(Tree|Data)$, which is equivalent to maximize:

$$p(Tree)p(Data|Tree)$$

where $p(Tree)$ is the prior probability of the tree and
$p(Data|Tree)$ is the likelihood of the data given the model.

Let us introduce the following notations:

- $N$ : number of instances,
- $J$ : number of output values,
- $T$ : a model of decision tree,
- $\mathbb{K}$ : set of $K$ input variables,

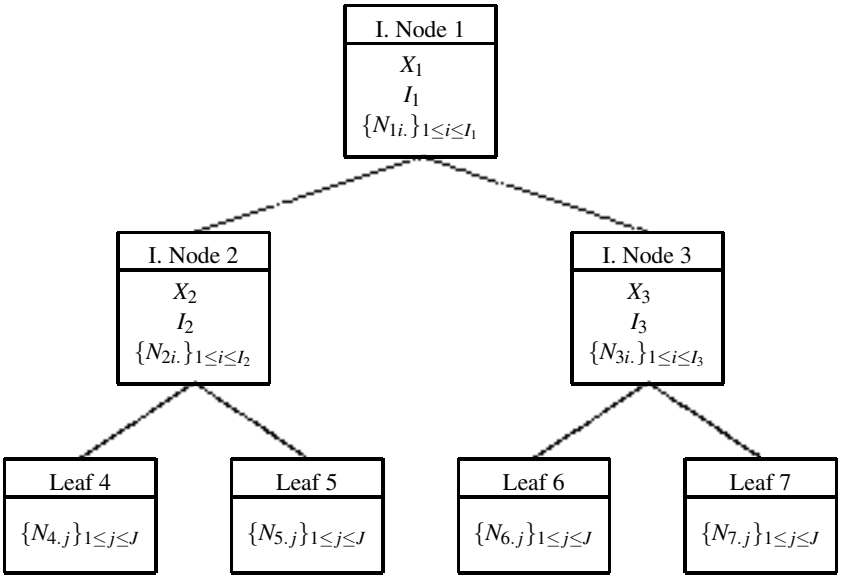



**Fig. 1** Example of decision tree. The internal nodes (I. Node) represent the decision rules and the leaves represent the distribution of the output values.

- $\mathbb{K}_T$ : subset of $K_T$ input variables used by tree $T$,
- $\mathbb{S}_T$ : set of internal nodes of tree $T$,
- $\mathbb{L}_T$ : set of terminal nodes (leaves) of tree $T$,
- $X_s$ : segmentation variable of node $s$,
- $N_{s.}$ : number of instances in node $s$,
- $V_{X_s}$ : number of values of variable $X_s$ in node $s$, in the categorical case,
- $I_s$ : number of child nodes of node $s$,
- $N_{si.}$ : number of instances in the $i^{th}$ child of node $s$,
- $N_{l.j}$ : number of instances of output value $j$ in leaf $l$.

A decision tree model is defined by its structure, the distribution of the instances in this structure and the distribution of output values in the leaves (cf. Figure 1). The structure of the decision tree model consists of the set of internal nodes $\mathbb{S}_T$ (nodes with at least two children), the set of leaves and the relations between these nodes.

The distribution of the instances in this structure is defined by the partition of the segmentation variable in each internal node and by the distribution of the output values in each leaf. A decision tree model $T$ is thus defined by:

- the subset of variables $\mathbb{K}_T$ used by model $T$, that is the number of selected variables $K_T$ and the choice of the $K_T$ variables among $K$,
- the number of child nodes $I_s$,

  - if $I_s = 1$ then the node is a leaf,
  - if $I_s > 1$ then the node is an internal node,
- the distribution of the instances in each internal node $s$:
  - the segmentation variable $X_s$,
  - the number of parts (intervals or groups of values) $I_s$,
  - the distribution of the instances on this parts (child nodes) $\{N_{si.}\}_{1 \leq i \leq I_s}$,
- the distribution of the output values in each leaf $l$: $\left\{N_{l.j}\right\}_{1 \leq j \leq J}$.

### *3.2 Evaluation Criterion*

The evaluation criterion we propose here is the negative logarithm of the posterior tree probability given the data. As the data probability is constant whatever the model, the criterion is defined as

$$c(Tree) = -\log p(Tree)p(Data|Tree)$$

We choose the prior model probability $p(Tree)$ by exploiting the hierarchy of the modelization parameters. This hierarchy enables to describe dependence relationships between parameters. The prior choice comes from hierarchical extensions of the Bayesian approach. In the case of a complex parameter set, the uncertainty of high level parameters is expressed first, then, conditionally, the uncertainty on low level parameters. Bayes law enables us to express $p(Tree)$ according to a parsimony principle close to the Minimum Description Length approach and according to prior distributions for these parameters.

There are many ways to define such parameters hierarchy. The first would consist in defining the structure then the segmentations, then the class distribution for the leaves. In this article, we propose to exploit the implicit tree hierarchy by defining the model at the root level independently of its children. Then, in a recursive way, the nodes are described from the root children to the leaves. The prior probability of a MODL decision tree is thus defined as :

$$\begin{aligned} p(Tree) = p(\mathbb{K}_T) \times & \\ & \prod_{s \in \mathbb{S}_T} p(I_s)p(X_s|\mathbb{K}_T)p(N_{si.}|\mathbb{K}_T, X_s, N_{s.}, I_s) \\ & \prod_{l \in \mathbb{L}_T} p(I_l)p(N_{l.j}|\mathbb{K}_T, N_{l.}) \end{aligned} \tag{2}$$

The first line in equation 2 represents the prior probability of variable selection. The second line is related to internal node probability and the last line represents leaf node probability.

**Prior Probability of Variable Selection**

For the variable selection parameters, we reuse the prior introduced by Boullé (2007) in the case of the selective naive Bayes classifier. We propose a hierarchic prior, by first choosing the number of selected variables and second choosing the subset of selected variables. For the number $K_T$ of variables, we propose to use a uniform prior between 0 and $K$ variables, representing $(K+1)$ equiprobable alternatives. For the choice of the $K_T$ variables, we assign the same probability to every subset of $K_T$ variables. The number of combinations $\binom{K}{K_T}$ seems the natural way to compute this prior, but it has the disadvantage of being symmetric. Beyond $K/2$ variables, every new variable makes the selection more probable. Thus, adding irrelevant variables is favored, provided that this has an insignificant impact on the likelihood of the model. As we prefer simpler models, we propose to use the number of combinations with replacement $\binom{K+K_T-1}{K_T}$.It thus gives :

$$P(\mathbb{K}_T) = \frac{1}{K+1} \frac{1}{\binom{K+K_T-1}{K_T}}$$

**Prior Probability of an Internal Node**

Knowing the selected variables and the parent nodes, the internal node can be defined by status (either internal node or leaf), segmentation parameters (the segmentation variable, the segmentation numbers and distribution of instances in segments. We consider that, for each internal node, the choice of the segmentation variable is independent and equal for all the selected explicative variables. To express the probability of the size of the segmentation of a given internal node, the simplest assumption of equiprobability leads to $p(I_s|K_T,X_s,N_{s.}) = \frac{1}{N_s}$ for a numerical variable and $p(I_s|K_T,X_s,N_{s.}) = \frac{1}{V_s}$ for a categorical variable. However, we obtained with such prior very cautious trees, as the higher the instances number, the lower the prior probability. That is why we propose here a prior inspired from the Minimum Description Length approach. Rissanen proposes an optimal coding of integers and gives the associated probability in Rissanen (1983). This universal prior is defined so that the small integers are more probable than the large integers, and the rate of decay is taken to be as small as possible. According to Rissanen, this prior is "universal" because its resulting code length (negative log of the probability) realizes the shortest coding of large integers. This prior is attractive even in the case of finite sets of integers, because it makes small integers preferable to large integers with the slightest possible difference. The optimal length, in bits, of an integer $I_s$ is :

$$C_{Ris}(I_s) = log_2(2.865) + log_2(I_s) + log_2(log_2(I_s)) + ...$$

We then obtain the universal prior probability of $I_s$ segments :

$$p(I_s|K_T,X_s,N_{s.}) = 2^{-C_{Ris}(I_s)}$$

Moreover, by using the fact that an internal node has at least two children, the status of the node has not to be explicitly described. Only the number of segments, either one for a leaf or between 2 and $N_s$ children for an internal node, are described.

For a numerical variable, the prior probability of the segmentation intervals is obtained similarly to the univariate MODL discretization (cf. section 2) :

$$\frac{1}{K_T} 2^{-C_{Ris}(I_s)} \frac{1}{\binom{N_s+I_s-1}{I_s-1}}$$

For a categorical variable, the prior probability is obtained similarly to the univariate MODL grouping method (Boullé, 2005) :

$$\frac{1}{K_T} 2^{-C_{Ris}(I_s)} \frac{1}{B(V_{X_s}, I_s)}$$

$B(X,Y)$ is the number of divisions of the $X$ values into $Y$ groups (with eventually empty groups). When $X = Y$, $B(X,Y)$ is the Bell number. In the general case, B(X,Y) can be written as a sum of Stirling numbers of the second kind S(X,y) :

$$B(X,Y) = \sum_{y=1}^{Y} S(X,y)$$

$S(X,y)$ stands for the number of ways of partitioning a set of X elements into y nonempty sets.

**Prior Probability of a Leaf**

To end up, it remains to define the prior of leaves probability, that is to say the class distribution for each leaf. Assuming the distributions are equiprobable, it remains to calculate the number of multinomial distributions of $N_{l.}$ instances among $J$ classes :

$$2^{-C_{Ris}(1)} \frac{1}{\binom{N_{l.}+J-1}{J-1}}$$

As internal nodes, the terms $2^{-C_{Ris}(1)}$ corresponds to the choice of the size of the segmentation, which is 1 for leaves.

**Likelihood Probability**

We have now to explicit the likelihood probability of the data given the model. The data distribution depends only of tree leaves. Knowing the multinomial model defined on one leaf, we deduce the likelihood :

$$p(Data|Tree) = \prod_{l \in L} \frac{N_{l.}!}{N_{l.1}!N_{l.2}!...N_{l.J}!}$$

**MODL Decision Tree Criterion**

Endly, taking the negative logarithm of its posterior probability, the optimal tree cost is :

$$\begin{aligned} C_{opt}(T) = & \log(K+1) + \log\binom{K+K_T-1}{K_T} + \\ & + \sum_{s \in \mathbb{S}_{T_n}} \log K_T + C_{Ris}(I_s)\log 2 + \log\binom{N_{s.}+I_s-1}{I_s-1} + \\ & + \sum_{s \in \mathbb{S}_{T_c}} \log K_T + C_{Ris}(I_s)\log 2 + \log B(V_{X_s}, I_s) + \\ & + \sum_{l \in \mathbb{L}_T} C_{Ris}(1)\log 2 + \log\binom{N_{l.}+J-1}{J-1} + \\ & + \sum_{l \in \mathbb{L}_T} \log \frac{N_{l.}!}{N_{l.1}!N_{l.2}!\ldots N_{l.J}!} \end{aligned}$$

where $\mathbb{S}_{T_n}$ and $\mathbb{S}_{T_c}$ are the internal nodes sets using respectively numerical or categorical segmentation variable. It is noteworthy that, using Stirling's approximation, the last multinomial term of the formula is asymptotically equal to the target entropy in the leaves of the tree (Cover and Thomas, 1991). Thus, the whole crietrion clearly relates to an entropy-based impurity measure, with a penalization for complex trees.

## 4 Optimization Algorithms

The induction of an optimal decision tree from a data set is NP-hard (Naumov, 1991). The exhaustive search algorithm is then excluded. In this article we exploit a pre-pruning algorithm 1 and a post-pruning algorithm 2. The pre-pruning algorithm starts with the root node and searches the best partition according to the criterion presented above. The leaves are segmented while the criterion is improved. For each leaf, the partition is performed according to the univariate MODL discretization or grouping methods, then the global cost of the tree is updated by accounting for this new partition. The partition is really completed if the global cost is improved. The optimum is then searched with successive local optimums at leaf levels. This algorithm is close to those used in ID3 and CHAID decision trees. The difference lies in the fact that the segmentation of two leaves is not conducted independently as the criterion is global. One leaf node can not be segmented unless it is the best choice of segmentation. In practice, the additivity of the criterion enables to update only the cost of the considered node. The algorithm does not guarantee to find the global optimum but its maximal complexity is $\mathcal{O}(KJN^2Log(N))$, in the case of an unbalanced tree. Its is $\mathcal{O}(KJN(LogN)^2)$ on average in the case of a balanced tree. This algorithm is deterministic and thus it always leads to the same local optimum.

**Algorithm 1.** Top-Down algorithm with pre-pruning for optimal tree search

**Require:** $T$ the root tree
**Ensure:** the tree $\hat{T}$ which minimizes the proposed criterion
$T^* \leftarrow T$
**while** criterion improvement **do**
$\hat{T} \leftarrow T^*$
**for all** leaf l of the tree **do**
$T' \leftarrow T^*$
**for all** variable $X$ of $\mathbb{K}$ **do**
Search the partition rule on the leaf $l$ according $X$ which best improves the criterion
$T_X \leftarrow T^* + P_X(l)$
**if** $c(T_X) < c(T')$ **then**
$T' \leftarrow T_X$
**end if**
**end for**
**if** $c(T') < c(T^*)$ **then**
$T^* \leftarrow T'$
**end if**
**end for**
**end while**

Unfortunately, the pre-pruning algorithm creates small and under-fitted decision trees. To go above this *horizon effect*, we have also exploited our criterion with a post-pruning algorithm (Breiman *et al.*, 1984). The post-pruning algorithm consists in two steps. The first step is the top-down building of the deepest tree by choosing the best univariate MODL partitions for each leaf, even if it does not lead to an improvement in the global criterion. The tree with the minimum cost is memorized during this step. This step ends when there are no more MODL informative variables left. Starting from the obtained deepest tree, the second step considers only nodes consisting of leaves, and prunes the node which leads the best improvement of the criterion. Only the internal node whose children are all leaves are candidates for pruning. Like in the first step, the tree with the minimum cost is memorized. This algorithm is also deterministic and it always leads to the same local optimum. At least, this algorithm guaranties to find a decision tree with a better cost than the tree resulting of algorithm 1. This means that the posterior probability of the tree can only be improved using the post-pruning algorithm.

## 5 Experiments

This section presents an experimental evaluation of our supervised decision trees methods described in the previous sections.

**Algorithm 2.** Top-Down algorithm with post-pruning for optimal tree search

**Require:** $T$ the root tree
**Ensure:** the tree $\hat{T}$ which minimizes the proposed criterion
  $T^* \leftarrow T$
  **while** there are MODL informative variables for one leaf node **do**
    $\hat{T} \leftarrow T^*$
    **for all** leaf l of the tree **do**
      $T' \leftarrow T^*$
      **for all** variable $X$ of $\mathbb{K}$ **do**
        Search the partition rule on the leaf $l$ according $X$ which best improves the criterion
        $T_X \leftarrow T^* + P_X(l)$
        **if** $c(T_X) < c(T')$ **then**
          $T' \leftarrow T_X$
        **end if**
      **end for**
      $T^* \leftarrow T'$
    **end for**
  **end while**
  **while** the tree is not reduced to its root **do**
    $\hat{T} \leftarrow T^*$
    **for all** internal node s of the tree whose children are all leaves **do**
      $T_s \leftarrow T^* - children(s)$
      **if** $c(T_s) < c(T')$ **then**
        $T' \leftarrow T_s$
      **end if**
      $T^* \leftarrow T'$
    **end for**
  **end while**

## 5.1 Experiments Setup

We conduct the experiments on two collections of data sets: 30 data sets from the repository at University of California at Irvine (Blake and Merz, 1996) and 5 data sets from the WCCI 2006 performance prediction challenge (Guyon *et al.*, 2006). These data sets represent a large diversity of number of variables, instances and classes, with numerical and/or categorical variables. A summary of some properties of these data sets is given in Table 1 for the UCI data sets and in Table 4 for the challenge data sets.

We evaluate two versions of the pre-pruning algorithm 1 and the post-pruning algorithm 2, with binary trees and N-ary trees. For binary trees, we constrain the univariate partition (discretization or grouping) of each node to build at most two subparts, related to two child nodes. On the opposite, internal nodes of N-ary trees can have more than two children. For more convenience, we call our decision tree family MTrees (MODL Trees). Our evaluated methods are:

**Table 1** UCI Data Sets. The properties of the used UCI data sets are : number of instances, number of variables, number of classes and majority accuracy

| Name | Variables | Instances | Classes | Majority Accuracy |
|---|---|---|---|---|
| Adult | 15 | 48842 | 2 | 0.76 |
| Australian | 14 | 690 | 2 | 0.56 |
| Breast | 10 | 699 | 2 | 0.66 |
| Crx | 15 | 690 | 2 | 0.56 |
| German | 24 | 1000 | 2 | 0.70 |
| Glass | 9 | 214 | 6 | 0.36 |
| Heart | 13 | 270 | 2 | 0.56 |
| Hepatitis | 19 | 155 | 2 | 0.79 |
| HorseColic | 27 | 368 | 2 | 0.63 |
| Hypothyroid | 25 | 3163 | 2 | 0.95 |
| Ionosphere | 34 | 351 | 2 | 0.64 |
| Iris | 4 | 150 | 3 | 0.33 |
| LED | 7 | 1000 | 10 | 0.11 |
| LED17 | 24 | 10000 | 10 | 0.11 |
| Letter | 16 | 20000 | 26 | 0.04 |
| Mushroom | 22 | 8416 | 2 | 0.53 |
| PenDigits | 16 | 10992 | 10 | 0.10 |
| Pima | 8 | 768 | 2 | 0.65 |
| Satimage | 36 | 6435 | 6 | 0.24 |
| Segmentation | 19 | 2310 | 7 | 0.14 |
| SickEuthyroid | 25 | 3163 | 2 | 0.91 |
| Sonar | 60 | 208 | 2 | 0.53 |
| Spam | 57 | 4307 | 2 | 0.65 |
| Thyroid | 21 | 7200 | 3 | 0.93 |
| TicTacToe | 9 | 958 | 2 | 0.65 |
| Vehicle | 18 | 846 | 4 | 0.26 |
| Waveform | 21 | 5000 | 3 | 0.34 |
| WaveformNoise | 40 | 5000 | 3 | 0.34 |
| Wine | 13 | 178 | 3 | 0.40 |
| Yeast | 9 | 1484 | 10 | 0.31 |

- MTp : MTree with post-pruning top-down algorithm.
- MTp(2) : MTree with post-pruning top-down algorithm and a binary tree structure.
- MT : MTree with pre-pruning top-down algorithm algorithm.
- MT(2) : MTree with pre-pruning top-down algorithm and a binary tree structure.
- NMT : Naive MODL tree is a top-down building of the deepest tree by choosing the best univariate MODL partition, without any pruning.

We compared our methods with J48 and SimpleCART which are implementations of C4.5 and CART in open-source data mining software WEKA (Garner, 1995). We

take as parameters those defined by default in the software. We evaluate the accuracy (ACC), the area under the ROC curve (AUC)(Fawcett, 2003), the number of nodes (internal nodes and leaves) and the training time. Provost et al. (1998) propose to use receiver operating characteristic (ROC) analysis rather than the accuracy to evaluate induction models (Provost *et al.*, 1998). The ACC criterion evaluates the accuracy of the prediction, no matter whether the conditional probability of the predicted class is 51% or 100%. The AUC criterion evaluates the ranking of the class conditional probabilities. In a two-classes problem, the AUC is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. In our experiments, we use the approach of Provost and Domingos (2001) to calculate the multi-class AUC (Provost and Domingos, 2001).

We collect and average the four criteria owing to a stratified 10-fold cross validation, for the seven evaluated methods on the thirty five data sets. In 10-fold cross-validation, the original data set is partitioned into 10 subsamples. Of the 10 subsamples, a single subsample is retained as the test data for testing the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times.

## *5.2 UCI Results*

The geometric means of the four criteria for each method are summarized in Table 2. The great diversity of the data sets increases the variance of the criterion. Therefore we prefer to support our analysis on the geometric mean, which allows comparing the criterion ratios between the various methods. The full results are also reported in Table 3.

**Table 2** Evaluation of the methods on UCI data sets : accuracy, AUC, size (number of nodes), training time and tree cost

| Method | Train data set | | Test data set | | Size | Time | $C_{opt}(T)$ |
|---|---|---|---|---|---|---|---|
| | Acc. | AUC | Acc. | AUC | | | |
| MT(2) | 0.845 | 0.914 | 0.819 | 0.889 | 17.5 | 0.5 | 524 |
| MT | 0.841 | 0.915 | 0.813 | 0.884 | 19.4 | 0.5 | 565 |
| MTp(2) | 0.840 | 0.910 | 0.822 | 0.891 | 17.4 | 0.6 | 508 |
| MTp | 0.834 | 0.905 | 0.817 | 0.890 | 19.5 | 0.6 | 547 |
| NMT | 0.879 | 0.959 | 0.762 | 0.844 | 142.3 | 0.8 | 1095 |
| sCART | 0.854 | 0.921 | 0.822 | 0.876 | 30.7 | 1.0 | × |
| J48 | 0.929 | 0.962 | 0.834 | 0.881 | 77.1 | 0.1 | × |

Overall J48 obtains the best geometric mean of accuracy and MTp(2) obtains the best geometric mean of AUC. In most of the cases, AUC and accuracy results are close (cf. Table 3). These weak differences are not surprising, since the decision trees are a mature technology and the differences of performance are often marginal.

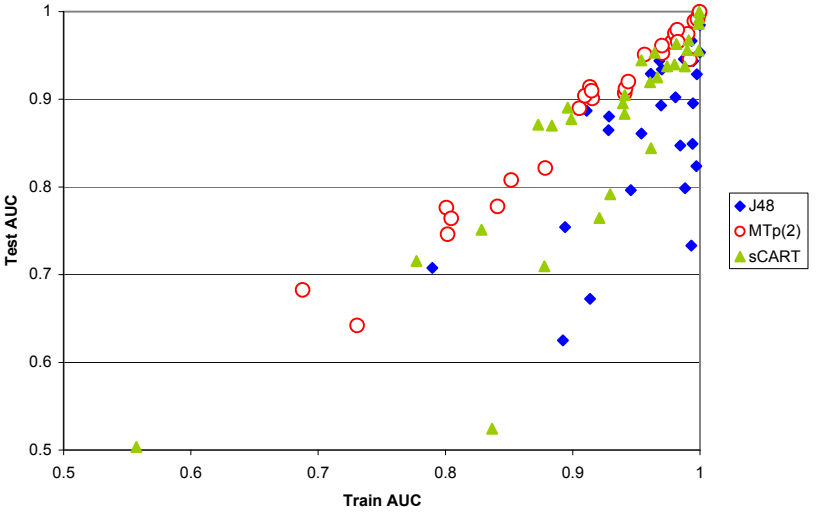



**Fig. 2** Train vs test AUC on 30 UCI data sets

On the other hand, the complexity of the tree structure is approximately four times less with MTree than with J48 and twice less than with SimpleCART. This property makes the interpretation of our trees considerably easier for the domain expert, and their deployment faster. Concerning training time, MTree is five times slower than J48 and twice faster than SimpleCART. Binary MTrees have better accuracy and AUC than N-ary MTrees. Constraining the algorithms to build binary trees leads to a better optimization of the criterion and a better predictive performance with only a slight impact on the tree size. The test results show that the tree cost of unpruned MTrees (NMT) is twice that of pruned MTrees, while the test performances (Acc and AUC) are far worse. It is noteworthy that the criterion of binary MTrees is smaller than the one of N-ary MTrees. This shows that the performance (accuracy and AUC) of the MTree trees is clearly correlated with the value of the tree criterion. MTp(2) is slightly prone to overfitting but it overfits less the data than the other methods. Figure 2 clearly shows that the differences between train and test AUC on 30 UCI data sets are lower with MTree than with J48 or SimpleCART.

A detailed analysis of the results (cf. Table 3) shows that MTree accuracy are worse with data sets having correlated variables such as Letter or image segmentation. On the other hand for marketing data sets such as Adult, MTree is slightly better while having ten times less nodes than J48.

## *5.3 Prediction Challenge Results*

This section reports the results obtained by our method on the performance prediction challenge of Guyon *et al.* (2006). The purpose of the performance prediction

**Table 3** Test accuracy, AUC and tree size of post-pruned decision tree on UCI data sets, using ten-fold cross validation

| | Accuracy | | | | AUC | | | | Tree Size | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Data Set** | **MTp(2)** | **MTp** | **sCart** | **J48** | **MTp(2)** | **MTp** | **sCart** | **J48** | **MTp(2)** | **MTp** | **sCart** | **J48** |
| Adult | 0.864 | 0.862 | 0.863 | 0.860 | 0.910 | 0.911 | 0.888 | 0.886 | 124.8 | 176.1 | 120.2 | 1099 |
| Australian | 0.852 | 0.852 | 0.857 | 0.852 | 0.904 | 0.903 | 0.878 | 0.881 | 6.2 | 6.2 | 5.8 | 46.2 |
| Breast | 0.937 | 0.957 | 0.949 | 0.946 | 0.965 | 0.969 | 0.948 | 0.948 | 9.6 | 8.7 | 15.8 | 23.4 |
| Crx | 0.861 | 0.861 | 0.852 | 0.861 | 0.914 | 0.914 | 0.866 | 0.894 | 7 | 7 | 3.6 | 27.1 |
| German | 0.692 | 0.692 | 0.750 | 0.739 | 0.682 | 0.682 | 0.722 | 0.692 | 3.2 | 3.2 | 19.4 | 140.6 |
| Glass | 0.597 | 0.649 | 0.705 | 0.659 | 0.778 | 0.811 | 0.848 | 0.793 | 7.8 | 8.2 | 20 | 47 |
| Heart | 0.733 | 0.719 | 0.785 | 0.767 | 0.808 | 0.810 | 0.792 | 0.755 | 7.2 | 7.7 | 14.2 | 33.8 |
| Hepatitis | 0.806 | 0.806 | 0.786 | 0.838 | 0.642 | 0.642 | 0.598 | 0.697 | 3 | 3 | 9.6 | 17.8 |
| HorseColic | 0.843 | 0.843 | 0.875 | 0.878 | 0.822 | 0.822 | 0.861 | 0.864 | 5.6 | 5.6 | 10 | 19.6 |
| Hypothyroid | 0.992 | 0.992 | 0.992 | 0.992 | 0.979 | 0.976 | 0.957 | 0.95 | 5.8 | 10.4 | 10.8 | 11.8 |
| Ionosphere | 0.898 | 0.889 | 0.898 | 0.915 | 0.901 | 0.908 | 0.896 | 0.895 | 5.2 | 7.9 | 8.8 | 27.4 |
| Iris | 0.953 | 0.933 | 0.953 | 0.960 | 0.975 | 0.963 | 1.000 | 0.990 | 5 | 4 | 8 | 8.4 |
| LED | 0.705 | 0.705 | 0.725 | 0.729 | 0.920 | 0.920 | 0.916 | 0.920 | 29 | 29 | 110 | 62.2 |
| LED17 | 0.735 | 0.735 | 0.735 | 0.722 | 0.951 | 0.951 | 0.957 | 0.891 | 75.6 | 75.6 | 123.8 | 890 |
| Letter | 0.768 | 0.738 | 0.869 | 0.879 | 0.975 | 0.966 | 0.965 | 0.964 | 461.2 | 531.8 | 2091.2 | 2321.6 |
| Mushroom | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 1.000 | 12.6 | 14 | 13.4 | 29.8 |
| PenDigits | 0.945 | 0.903 | 0.963 | 0.966 | 0.991 | 0.984 | 0.991 | 0.992 | 170.6 | 247 | 363.4 | 375.6 |
| Pima | 0.741 | 0.740 | 0.751 | 0.738 | 0.764 | 0.767 | 0.725 | 0.749 | 8.4 | 6.9 | 16.2 | 37.4 |
| Satimage | 0.853 | 0.852 | 0.868 | 0.873 | 0.972 | 0.971 | 0.981 | 0.979 | 76 | 72.3 | 165.2 | 551.4 |
| Segmentation | 0.938 | 0.938 | 0.958 | 0.971 | 0.989 | 0.989 | 0.998 | 0.998 | 31.2 | 46.1 | 76.6 | 82.6 |
| SickEuthyroid | 0.978 | 0.979 | 0.977 | 0.979 | 0.961 | 0.956 | 0.96 | 0.941 | 11.4 | 12.5 | 14 | 26.2 |
| Sonar | 0.735 | 0.735 | 0.712 | 0.712 | 0.746 | 0.746 | 0.722 | 0.735 | 4.8 | 4.8 | 14.2 | 29.2 |
| Spam | 0.912 | 0.911 | 0.922 | 0.935 | 0.953 | 0.955 | 0.94 | 0.943 | 44.8 | 54.1 | 131.2 | 192.2 |
| Thyroid | 0.995 | 0.992 | 0.996 | 0.997 | 0.997 | 0.996 | 0.996 | 0.99 | 14.4 | 23 | 22.4 | 30.6 |
| TicTacToe | 0.923 | 0.866 | 0.932 | 0.851 | 0.965 | 0.921 | 0.963 | 0.899 | 42.8 | 49.1 | 67.2 | 135.4 |
| Vehicle | 0.676 | 0.661 | 0.701 | 0.726 | 0.890 | 0.877 | 0.926 | 0.932 | 24.4 | 28.2 | 104.8 | 136 |
| Waveform | 0.756 | 0.745 | 0.777 | 0.759 | 0.912 | 0.904 | 0.903 | 0.845 | 72.4 | 90.4 | 136.6 | 541.8 |
| WaveformNoise | 0.744 | 0.751 | 0.767 | 0.751 | 0.907 | 0.906 | 0.903 | 0.847 | 70.4 | 84.4 | 121.4 | 580.4 |
| Wine | 0.917 | 0.917 | 0.894 | 0.939 | 0.946 | 0.951 | 0.963 | 0.975 | 8.6 | 7.4 | 9.2 | 9.8 |
| Yeast | 0.565 | 0.542 | 0.309 | 0.503 | 0.776 | 0.779 | 0.501 | 0.749 | 16.8 | 22.8 | 1.4 | 96.6 |
| **Ar. Mean** | 0.830 | 0.825 | 0.837 | 0.843 | 0.896 | 0.895 | 0.885 | 0.886 | 45.5 | 54.9 | 127.6 | 254.4 |
| **Geo. Mean** | 0.822 | 0.817 | 0.822 | 0.834 | 0.891 | 0.890 | 0.876 | 0.881 | 17.7 | 19.9 | 30.7 | 77.1 |
| **Mean Rank** | 2.6 | 2.8 | 2.3 | 1.9 | 2.0 | 2.2 | 2.6 | 2.8 | 1.3 | 1.8 | 2.7 | 4.0 |

challenge is "*to stimulate research and reveal the state-of-the-art in model selection*". Five data sets are used in the challenge (cf. Table 4). The ada data set comes from the marketing domain, the gina data set from handwriting recognition, the hiva data set from drug discovery, the nova data set from text classification and the sylva data set from ecology.

**Table 4** WCCI Challenge Data Sets. The properties of the used UCI data sets are : number of instances, number of variables, number of classes and majority accuracy

| **Name** | **Variables** | **Instances** | **Classes** | **Majority Accuracy** |
|---|---|---|---|---|
| ada | 48 | 4562 | 2 | 0.75 |
| gina | 970 | 3468 | 2 | 0.51 |
| hiva | 1617 | 4229 | 2 | 0.96 |
| nova | 16969 | 1929 | 2 | 0.72 |
| sylva | 216 | 14394 | 2 | 0.94 |

The detailed results of our evaluation are presented in Table 5. Unfortunely, we cannot report the results of simpleCART and J48 on all the data sets, since some of these data sets are too large given the Weka implementation of simpleCART and J48. Overall, MTree with post-pruning is the best method. It comes first on most of the data sets for the AUC, accuracy and tree size criteria.

**Table 5** Test accuracy, AUC and tree size of post-pruned decision tree on prediction challenge data sets, using ten-fold cross validation

| | Accuracy | | | | AUC | | | | Tree Size | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Data Set** | **MTp(2)** | **MTp** | **sCart** | **J48** | **MTp(2)** | **MTp** | **sCart** | **J48** | **MTp(2)** | **MTp** | **sCart** | **J48** |
| ada | 0.847 | 0.847 | 0.842 | 0.846 | 0.887 | 0.890 | 0.860 | 0.860 | 22.0 | 23.6 | 28.1 | 224.0 |
| gina | 0.881 | 0.863 | 0.894 | 0.867 | 0.923 | 0.913 | 0.918 | 0.862 | 47.8 | 49.1 | 64.4 | 247.7 |
| hiva | 0.966 | 0.966 | - | 0.955 | 0.622 | 0.622 | - | 0.659 | 6.0 | 6.0 | - | 64.4 |
| nova | 0.866 | 0.866 | - | - | 0.817 | 0.817 | - | - | 17.6 | 17.6 | - | - |
| sylva | 0.989 | 0.989 | 0.991 | 0.990 | 0.991 | 0.991 | 0.981 | 0.954 | 26.2 | 41.4 | 41.0 | 105.2 |

## 6 Conclusion

The Bayesian criterion presented in this article gives a complete criterion to evaluate a decision tree by taking into account the structure of the tree, the choice of the explanatory variables, the segmentation in each internal node and the distributions of the classes in each leaf. This corresponds to an exact analytic evaluation of the posterior probability of a tree given the data and results in a parameter-free evaluation criterion. We have tested two optimization algorithms. The first one is a pre-pruning heuristic and the second one is a post-pruning heuristic which leads to a better optimization and obtain the better performance. Evaluations on 30 UCI data sets show that MTrees obtains state of the art performance while being much less complex. The evaluation on prediction challenge data sets show that our method gets the best results and builds the less complex decision trees. It is also noteworthy that binary trees are better on average than N-ary trees. Therefore designing new algorithms is a promising direction to get better performance. Another direction of research is to use MODL trees with random forest or Bayesian Model Averaging.

## References

Blake, C., Merz, C.: UCI repository of machine learning databases (1996), `http://www.ics.uci.edu/mlearn/MLRepository.html`

Boullé, M.: A Bayes optimal approach for partitioning the values of categorical attributes. Journal of Machine Learning Research 6, 1431–1452 (2005)

Boullé, M.: MODL: a Bayes optimal discretization method for continuous attributes. Machine Learning 65(1), 131–165 (2006)

Boullé, M.: Compression-Based Averaging of Selective Naive Bayes Classifiers. Journal of Machine Learning Research 8, 1659–1685 (2007)

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification And Regression Trees. Chapman and Hall, New York (1984)

Breslow, L.A., Aha, D.W.: Simplifying decision trees: A survey. Knowl. Eng. Rev. 12(1), 1–40 (1997), http://dx.doi.org/10.1017/S0269888997000015

Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & sons, Chichester (1991)

Fawcett, T.: ROC Graphs: Notes and Practical Considerations for Researchers. Technical Report HPL-2003-4, HP Laboratories (2003)

Garner, S.R.: WEKA: The Waikato Environment for Knowledge Analysis. In: Proc. of the New Zealand Computer Science Research Students Conference, pp. 57–64 (1995)

Guyon, I., Saffari, A., Dror, G., Bumann, J.: Performance Prediction Challenge. In: International Joint Conference on Neural Networks, pp. 2958–2965 (2006), http://www.modelselect.inf.ethz.ch/index.php

Kass, G.: An exploratory technique for investigating large quantities of categorical data. Applied Statistics 29(2), 119–127 (1980)

Kohavi, R., Quinlan, R.: Decision tree discovery. In: Handbook of Data Mining and Knowledge Discovery, pp. 267–276. University Press (2002)

Morgan, J., Sonquist, J.A.: Problems in the analysis of Survey data, And a proposal. Journal of the American Statistical Association 58, 415–435 (1963)

Murthy, S.K.: Automatic construction of decision trees from data: A multi-disciplinary survey. Data Mining and Knowledge Discovery 2, 345–389 (1998)

Naumov, G.E.: NP-completeness of problems of construction of optimal decision trees. Soviet Physics 34(4), 270–271 (1991)

Provost, F., Domingos, P.: Well-trained PETs: Improving Probability Estimation Trees. Technical Report CeDER #IS-00-04, New York University (2001)

Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 445–553 (1998)

Quinlan, J., Rivest, R.: Inferring decision trees using the minimum description length principle. Inf. Comput. 80(3), 227–248 (1989)

Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1, 81–106 (1986)

Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)

Rissanen, J.: A universal prior for integers and estimation by minimum description length. Annals of Statistics 11(2), 416–431 (1983)

Wallace, C., Patrick, J.: Coding Decision Trees. Machine Learning 11, 7–22 (1993)

# Classifying Very-High-Dimensional Data with Random Forests of Oblique Decision Trees

Thanh-Nghi Do, Philippe Lenca, Stéphane Lallich, and Nguyen-Khang Pham

**Abstract.** The random forests method is one of the most successful ensemble methods. However, random forests do not have high performance when dealing with very-high-dimensional data in presence of dependencies. In this case one can expect that there exist many combinations between the variables and unfortunately the usual random forests method does not effectively exploit this situation. We here investigate a new approach for supervised classification with a huge number of numerical attributes. We propose a random oblique decision trees method. It consists of randomly choosing a subset of predictive attributes and it uses SVM as a split function of these attributes. We compare, on 25 datasets, the effectiveness with classical measures (e.g. precision, recall, F1-measure and accuracy) of random forests of random oblique decision trees with SVMs and random forests of C4.5. Our proposal has significant better performance on very-high-dimensional datasets with slightly better results on lower dimensional datasets.

**Keywords:** High Dimensional Data, Random Forests, Oblique Decision Trees, Support Vector Machines.

Thanh-Nghi Do
Institut Telecom; Telecom Bretagne, UMR CNRS 3192 Lab-STICC, Université européenne de Bretagne, France and Can Tho University, Vietnam
e-mail: tn.do@telecom-bretagne.eu

Philippe Lenca
Institut Telecom; Telecom Bretagne, UMR CNRS 3192 Lab-STICC, Université européenne de Bretagne, France
e-mail: philippe.lenca@telecom-bretagne.eu

Stéphane Lallich
Université de Lyon, Laboratoire ERIC, Lyon 2, France
e-mail: stephane.lallich@univ-lyon2.fr

Nguyen-Khang Pham
IRISA, Rennes, France and Can Tho University, Vietnam
e-mail: pnguyenk@irisa.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 39–55.
springerlink.com 

## 1 Introduction

Since the nineties the machine learning community studies how to combine multiple classifiers into an ensemble of classifiers to build models that are more accurate than a single one. The purpose of ensemble classifiers is to reduce the variance and/or the bias in learning algorithms. Bias is the systematic error term (independent of the learning sample) and variance is the error due to the variability of the model with respect to the learning sample randomness. Buntine (1992) introduced Bayesian techniques for tree averaging to reduce the variance in learning methods. Stacking method (Wolpert, 1992) aims at minimizing the bias of learning algorithms. Freund and Schapire (1995) proposed Boosting to simultaneously reduce the bias and the variance while the Bagging method proposed by Breiman (1996) reduces the variance of a learning algorithm without increasing its bias too much.

The random forests approach proposed by Breiman (2001) has been one of the most successful ensemble methods. Random forests algorithm creates a collection of unpruned decision trees (built so that at each node the best split is done from a randomly chosen subset of attributes) from bootstrap samples (sampling with replacement from the original dataset). The generalization error of a forest depends on the strength of the individual trees in the forest and on the dependence between them. Random forest algorithm constructs unpruned trees for keeping low bias and uses the randomization for controlling high diversity between trees in the forest. Two classifiers are diverse if they make different errors on new data points (Dietterich, 2000a). Random forests approach gives high accuracy compared with state-of-the-art supervised classification algorithms, including AdaBoost (Freund and Schapire, 1995) and SVM (Vapnik, 1995). As mentioned in Breiman (2001) random forests method is fast, robust to noise and does not overfit unlike AdaBoost algorithm which is sensitive to noisy data (Dietterich, 2000b). Random forests algorithm has been shown to build accurate models with practical relevance for classification, regression and novelty detection (Breiman, 2001).

The tree construction of habitual random forests only picks, at each node, a single attribute for node splitting. Thus the individual trees are less efficient when dealing with data having dependencies among attributes, as it could be the case with very-high-dimensional datasets.

In this paper which is an extended version of Do *et al.* (2009), we propose to use linear proximal SVMs (Fung and Mangasarian, 2001) for performing multivariate node splitting during the tree construction (in order to use dependencies between attributes), producing individual classifiers that are stronger than in the usual forests. Numerical test results on UCI (Asuncion and Newman, 2007), Statlog (Michie *et al.*, 1994) and very-high-dimensional Bio-medical (Jinyan and Huiqing, 2002) datasets show that our random forests of oblique decision trees are often more accurate than random forests of C4.5 (Quinlan, 1993) and SVM in terms of precision, recall, F1-measure and accuracy (van Rijsbergen, 1979). In particular our proposal has significant better performance on very-high-dimensional data with better -but not significant- results on lower dimensional datasets.

The paper is organized as follows. Section 2 briefly introduces random forests and our random forests of oblique decision trees for classification. The experimental results are presented in Section 3. We then conclude in Section 4.

## 2 Random Forests of Oblique Decision Trees

In early bagging approach proposal of Breiman (1996), an ensemble of decision trees is built from bootstrap samples drawn with replacement from the original dataset. Then, the predictions of these trees are aggregated, by a majority vote in classification tasks or by an average for regression problems. Ho (1995) also proposed the random subspace method which randomly selects a subset of attributes for growing each tree. Amit and Geman (2001) used a random selection of attributes for the search of the best split at each node. Finally, these approaches were extended and formalized in the term of random forests by Breiman (2001).

### *2.1 Random Forests*

The random forests algorithm of Breiman (2001) aims at creating a collection of high performance decision trees with high diversity between individual trees in the forest. He proposed to use two strategies to keep low bias and low dependence between trees in the forest. For reaching out to low bias, he proposed to build the individual trees without pruning, i.e. which are grown to maximum depth. To the diversity control of the trees, he also proposed to uses a bootstrap replica from the original training set to construct the trees and randomly choose a subset of attributes on which to base the calculation of the best split at a decision node.

Let us consider the classification task with $m$ datapoints $x_i(i = 1,m)$ and $n$ attributes, a decision tree (denoted by DT) in a random forest of $k$ trees (denoted by RF$= \{DT_i\}_{i=1,k}$) is constructed as follows:

- The training set is a bootstrap replica of $m$ individuals, i.e. a random sampling with replacement from the original training set.
- For each node of the tree, randomly choose $n'$ attributes ($n' << n$, e.g. $n' = \sqrt{n}$) and calculate the best split based on one of these $n'$ attributes.
- The tree is grown to its maximal depth without pruning.

To classify a new individual, the prediction phase uses an unweighted majority vote of the trees for a classification task or an average-up for a regression task. Breiman proposed to use datapoints of out-of-bag (about 36.8% of the original training set are out of the bootstrap sample) to estimate important attributes and the error in the forest while adding a new tree. The random forests algorithm exhibits high accuracy. Furthermore, it is also fast, robust to noise and does not overfit. Breiman also extended random forests for unsupervised learning tasks (Breiman, 2001).

Recently, Robnik-Sikonja proposed in Robnik-Sikonja (2004) some possibilities for improving random forests. He investigated strategies to increase strength or to increase diversity of individual trees in the forest. He used several attribute evaluation measures instead of just one. He also proposed the use of weighted voting.

An another idea proposed by Geurts *et al.* (2006) aims at building totally random trees. They proposed and studied an algorithm called Extra-Trees. It uses the whole training set instead of a bootstrap replica to build the trees. At each level of the tree, the method randomly chooses an attribute to split the data (if it is an continuous attribute then the cut-point is also chosen randomly), i.e. independently of the class labels. As mentioned in Geurts *et al.* (2006) the explicit randomization of the splitting in the Extra-Trees could reduce variance more strongly than the weaker randomization schemes used by habitual random forests. Obviously, the algorithm is very fast for training, but the strength of the individual trees in the forest may be reduced. The algorithm Extra-Trees is close to the algorithm PERT (for perfect random tree ensembles) proposed in Cutler and Guohua (2001).

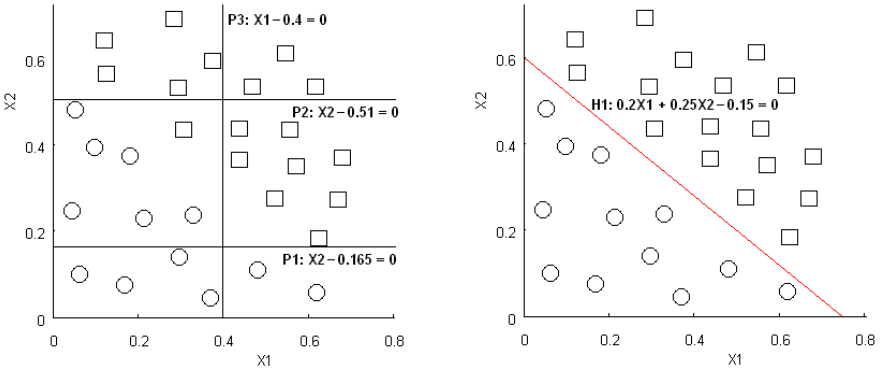



**Fig. 1** Single attribute (left) and bi-variate (right) node splitting

## *2.2 Oblique Decision Trees*

The tree construction of habitual random forests picks a single attribute for node splitting (Breiman *et al.*, 1984; Quinlan, 1993; Ho, 1995; Amit and Geman, 2001; Cutler and Guohua, 2001; Robnik-Sikonja, 2004; Geurts *et al.*, 2006). Thus, the strength of individual trees is reduced, particularly when dealing with datasets having dependencies among attributes. For example in figure 1, any univariate splitting (perpendicular to axes) can not totally separate the data into two classes but the bivariate splitting, i.e. $H_1$ (combination of two attributes) perfectly classifies the data into two classes (some ensemble methods can deal with this problem if they use a large number of trees, see for example Cutler and Guohua (2001) and Geurts *et al.*

(2006)). Therefore, the univariate splitting used by the usual tree construction is not robust in this case.

At the opposite, multivariate splitting criteria -where several attributes may participate in a single node split test-, may dramatically improve the trees performance. The problem of constructing an oblique decision tree is well known to be NP-hard (Heath, 1992). Most of the multivariate splitting criteria are based on linear combination of the input attributes. As pointed out by Rokach and Maimon (2005) finding the best linear combination can be achieved in different ways. For example, linear programming (Bennett and Mangasarian, 1994), linear discriminant analysis (Loh and Vanichsetakul, 1988; Yildiz and Alpaydin, 2005) or linear combinations of attributes (Breiman *et al.*, 1984). With multivariate splitting criteria each test is equivalent to a hyperplane with an oblique orientation to the axes. Because of the computational intractability of finding an optimal orientation for these hyperplanes, heuristic methods were proposed to produce good trees like in the algorithm OC1 (Murthy *et al.*, 1993, 1994). Indeed, the greedy approaches can deal only with low dimensional datasets due to combinatorial explosion.

The OC1 approach was extended by Wu *et al.* (1999) by modifying the splitting criterion of the basic OC1 algorithm or by post-processing OC1 output. While these modifications outperform the basic OC1 on the correctness and the robustness to noise, the optimal hyperplanes are found with standard SVMs through the resolution of a quadratic programming. Therefore, the proposed approach has a high cost for the learning task.

Our investigation aims at performing multivariate node splitting during tree construction, thus producing individual oblique classifiers that are stronger than the usual random forests. As a whole our method combines both advantages of oblique splitting and ensemble methods in an efficient manner.

## *2.3 Random Forests of Oblique Decision Trees*

Our random forest algorithm constructs a collection of oblique decision trees (denoted by RF-ODT) in the same framework of random forests proposed by Breiman (2001). The main difference is that each random oblique decision tree (ODT) in the forest (RF-ODT$= \{ODT_i\}_{i=1,k}$) uses linear SVMs for performing multivariate node splitting as proposed in Do *et al.* (2009). Our proposal is thus an hybridization of decision trees with SVMs. SVMs are here used in the growing phase to create the oblique trees.

Others works proposed hybridization in a post-growing phase, to attach classifiers to the tree's leaves as for example, genetic algorithm (Carvalho and Freitas, 2004), neural network (Zhou and Chen, 2002; Maji, 2008), SVM (Xu *et al.*, 2006) or multiple-classifier (Cohen *et al.*, 2007). Recently Simon *et al.* (2009) proposed to embedded multiple proximal SVM into a binary tree architecture for multi-classes problem.

We propose to use linear proximal SVMs (Fung and Mangasarian, 2001) to build oblique splits on randomly chosen attributes because they are very fast for training and give good accuracy when compared with standard SVMs (see for example the experiments by Do and Poulet (2006) where one million datapoints in 20-dimensional input space are classified into two classes in 13 seconds on a PC (2.4 GHz Pentium IV, 512 MB RAM)).

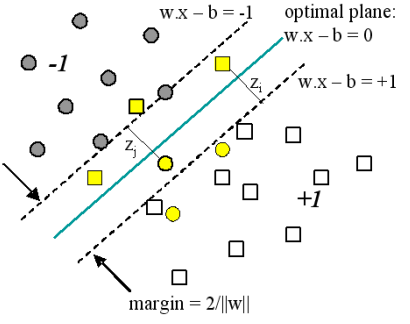



**Fig. 2** Linear separation of the datapoints into two classes

Briefly, consider the linear binary classification task depicted in figure 2, with $m$ datapoints $x_i (i = 1, m)$ in $n$ dimensions (attributes). It is represented by the $[m \times n]$ matrix $A$, having corresponding labels $y_i = \pm 1$, denoted by the $[m \times m]$ diagonal matrix $D$ of $\pm 1$ (where $D[i,i] = 1$ if $x_i$ is in class +1 and $D[i,i] = -1$ if $x_i$ is in class -1). A SVM algorithm tries to find the best plane to separate the classes, i.e. the one farthest from both class +1 and class -1. Any point falling on the wrong side of its supporting plane is considered to be an error. Therefore, SVMs simultaneously maximize the distance between two parallel supporting planes for each class and minimize the errors.

Classical SVMs pursue these goals with the quadratic program (1):

$$\min_{w,\, b,\, z} \psi(w,b,z) = (1/2)\|w\|^2 + cz \qquad (1)$$
$$s.t. : D(Aw - eb) + z \geq e$$

where $e$ is the column vector of 1, $z \in R^m$ is the non negative slack vector, and $c \in R^1$ a positive constant; $w$ and $b$ be the normal vector and the scalar of the plane respectively; $z$ and $c$ are used to tune errors and margin size.

The plane $(w,b)$ is obtained by solving the quadratic programming (1). Then, the classification function of a new datapoint $x$ based on the plane is:

$predict(x) = sign(w.x - b)$

Unfortunately, the computational cost requirements of the SVM solutions in (1) are at least the square of the number of training datapoints, making classical SVM

intractable for large datasets. The proximal SVM proposed by Fung and Mangasarian (2001) modified the quadratic programming (1) by using the equality instead of the inequality constraints and a least squares 2-norm error in the objective function $\psi$. They also changed the formulation of the margin maximization to the minimization of $1/2\|w,b\|^2$. Thus substituting for $z$ from the constraint in terms of $w$ and $b$ into the objective function $\psi$ of the quadratic programming (1) yields an unconstrained problem (2):

$$\min_{w,\ b} \Psi(w,b) = (1/2)\|w,b\|^2 + (c/2)\|e - D(Aw - eb)\|^2 \tag{2}$$

In the optimal configuration for (2), the gradient with respect to $w$ and $b$ should be zero. This yields the linear equation system of $(n+1)$ variables $(w_1,\ w_2,\ \ldots,\ w_n,\ b)$ as follows:

$$(w_1,\ w_2,\ \ldots,\ w_n,\ b)^T = (\frac{1}{c}I + E^T E)^{-1} E^T De \tag{3}$$

where $E = [A \quad -e]$, $I$ denotes the identity matrix.

The proximal SVM formulation (3) requires thus only the solution of linear equations of $(n+1)$ variables $(w_1, w_2,\ \ldots, w_n, b)$ instead of the quadratic programming (1). Its complexity is linear with the number of training datapoints. If the dimension of input space is small enough (less than $10^4$), even if there are millions of datapoints, the proximal SVM algorithm is able to classify them in an efficient manner. Numerical test results have shown that this algorithm gives similar accuracy compared to standard SVM like LibSVM (Chang and Lin, 2001) but the proximal SVMs are much faster than standard SVMs. Another non-standard SVM, the Least-Squares SVM proposed by Suykens and Vandewalle (1999) also replaces standard SVM optimization inequality constraints with equalities; so its performance is very close to the proximal SVM. Our proposal is thus efficient in comparison with methods discussed in this paper.

Therefore, we propose to use proximal SVMs for performing multivariate node splitting during oblique trees construction. Furthermore, at a decision node of the tree, the $n'$ randomly chosen attributes ($n' << n$, e.g. $n' = \sqrt{n}$) brings out to low-dimensional problems where the proximal SVM algorithm is very fast compared with other ones. In addition, the costs re-balancing method (Veropoulos *et al.*, 1999) is also used to deal with the class imbalance problem at multivariate node splitting. The random forests of oblique trees algorithm builds an ensemble of unpruned oblique trees according to the classical top-down procedure (see Figure 3). Note that our algorithm not only improves the strength of the individual trees in the forest using oblique splitting during tree construction but also keeps the high diversity between them as it is done with usual random forests method. It means that our forests create a collection of random oblique trees using a bootstrap replica from the original training set to construct oblique trees and a random subset of attributes on which to build multivariate splitting at each node.

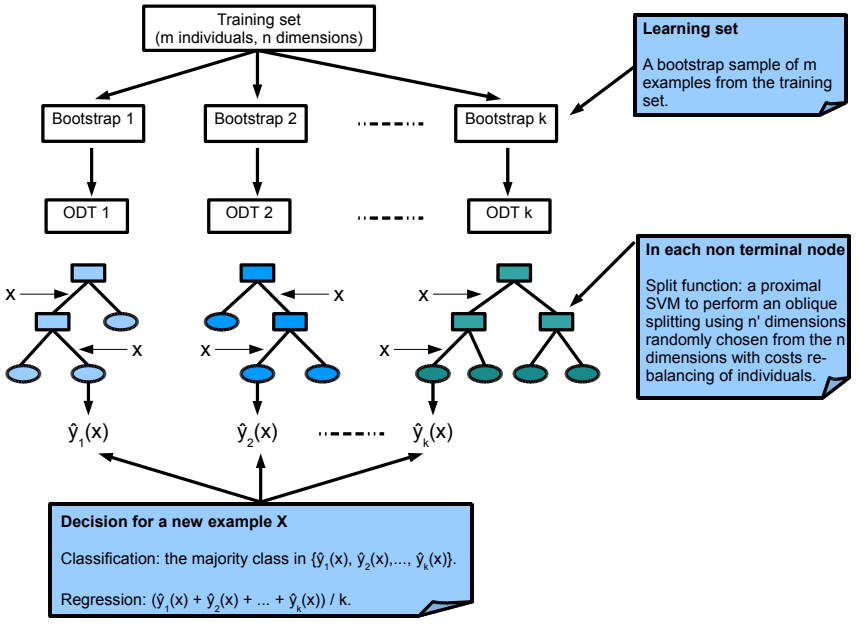



**Fig. 3** Random Forest of Oblique Decision Trees

## 3 Evaluation

We here report the comparisons of the performance of random forests of oblique trees, of random forests of C4.5 and of SVMs. Random forests of oblique decision trees (RF-ODT) and random forests of C4.5 decision trees (using C4.5 program by Quinlan (1993)) have been implemented in C++ and C respectively. We also use the highly efficient standard SVM algorithm LibSVM (Chang and Lin, 2001).

In order to test the statistical signification of the observed results we used both the Student test and the sign test. Indeed, these two tests give complementary point of views on the data.

### *3.1 Experiments Setup*

The experimental setup used fifteen very-high-dimensional datasets from the Biomedical repository (Jinyan and Huiqing, 2002) and ten standard datasets from UCI (Asuncion and Newman, 2007) and Statlog (Michie *et al.*, 1994) repositories.

In order to evaluate performance for binary classification tasks, we pre-processed multi-class (more than two classes, denoted by an asterisk in the tables 1 and 2) datasets as two-class problems. In the tables 1 and 2 the fourth column shows how

**Table 1** Description of very-high-dimensional datasets

| ID | Datasets | #Datapoints | #Dimensions | Classes | Protocols |
|---|---|---|---|---|---|
| 1 | Colon Tumor | 62 | 2000 | tumor, normal | loo |
| 2 | ALL-AML-Leukemia | 72 | 7129 | ALL, AML | trn-tst |
| 3 | *MLL-Leukemia | 72 | 12582 | MLL, rest | trn-tst |
| 4 | Breast Cancer | 97 | 24481 | relapse, non-relapse | trn-tst |
| 5 | Duke Breast Cancer | 42 | 7129 | cancer, normal | loo |
| 6 | Prostate Cancer | 136 | 12600 | cancer, normal | trn-tst |
| 7 | Lung Cancer | 181 | 12533 | cancer, normal | trn-tst |
| 8 | Central Nervous System | 60 | 7129 | positive, negative | loo |
| 9 | Translation Initiation Site | 13375 | 927 | positive, negative | 10-fold |
| 10 | Ovarian Cancer | 253 | 15154 | cancer, normal | loo |
| 11 | Diffuse Large B-Cell Lymphoma | 47 | 4026 | germinal, activated | loo |
| 12 | *Subtypes of Acute Lymphoblastic (Hyperdip) | 327 | 12558 | Hyperdip, rest | trn-tst |
| 13 | *Subtypes of Acute Lymphoblastic (TEL-AML1) | 327 | 12558 | TEL-AML1, rest | trn-tst |
| 14 | *Subtypes of Acute Lymphoblastic (T-ALL) | 327 | 12558 | TEL-ALL, rest | trn-tst |
| 15 | *Subtypes of Acute Lymphoblastic (Others) | 327 | 12558 | Others, diagnostic groups | trn-tst |

**Table 2** Description of standard datasets

| ID | Datasets | #Datapoints | #Dimensions | Classes | Protocols |
|---|---|---|---|---|---|
| 16 | Bupa | 345 | 6 | 1, 2 | 10-fold |
| 17 | Breast Cancer Wisconsin | 569 | 30 | M, B | 10-fold |
| 18 | Pima | 768 | 8 | 1, 2 | 10-fold |
| 19 | *Segment | 2310 | 19 | 1, rest | 10-fold |
| 20 | Spambase | 4601 | 57 | spam, non-spam | 10-fold |
| 21 | *Opticdigits | 5620 | 64 | 0, rest | trn-tst |
| 22 | *Satimage | 6435 | 36 | 1, rest | trn-tst |
| 23 | *Pendigits | 10992 | 16 | 9, rest | trn-tst |
| 24 | *Letters | 20000 | 16 | A, rest | 10-fold |
| 25 | *Shuttle | 58000 | 9 | 1, rest | trn-tst |

we convert multi-class to two-class ( for example with the OpticDigits dataset, the digit "0" is mapped to the +1 class and the remaining digits are considered as the -1 class). The performance of the classification algorithms is analyzed in terms of precision, recall, F1-measure and accuracy. The test protocols are presented in the last column of the tables 1 and 2. With datasets having training set $(trn)$ and testing set $(tst)$ available, we used the training data to tune the parameters of the algorithms for obtaining a good accuracy in the learning phase. For Random forests, we tuned the number of trees in the forests and the number of random attributes at each node splitting. For LibSVM, we tuned the positive constant $c$ for tradeoff of errors and the margin size ($c = 10^5$ for the 15 high dimensional datasets). Then the obtained model is evaluated on the test set. If the training set and testing set are not available then we used cross-validation protocols to evaluate the performance. With datasets having less than three hundred datapoints, the test protocol is leave-one-out cross-validation $(loo)$. It involves using a single datapoint from the dataset as the testing data and the remaining datapoints as the training data. This is repeated such that each datapoint in the dataset is used once as the testing data. With dataset having more than three hundred datapoints, 10-fold cross-validation is used to evaluate the performance. The dataset is partitioned into 10 folds. A single fold is retained as

the validation set, and the remaining 9 folds are used as training data. The cross-validation process is then repeated 10 times (folds). The results from the 10 folds are then averaged to produce the final result.

### 3.2 Classification Results on Very-High-Dimensional Datasets

For dealing with very-high-dimensional datasets we varied the size of the forest ($k$) from 50 to 500 trees and the number of random attributes ($n'$) for each node of the tree from 100 to 500. Then the good parameter values were chosen (table 3). With the standard SVM algorithm LibSVM, the linear kernels are appropriate for very-high-dimensional datasets having a very large number of dimensions and few datapoints.

The main result of the carried out experiments (table 4) is that RF-ODT outperforms RF-C4.5 and LibSVM. Overall, using paired Student ratio test, one can see that RF-ODT significantly improves the mean accuracy (table 5) of 3.6 percent points compared to RF-C4.5 (p-value $= 0.0462$) and 6.4 points compared to

**Table 3** Parameter values of random forest algorithms

| ID | Datasets | #Random dimensions | #Trees |
|---|---|---|---|
| 1 | Colon Tumor | 100 | 200 |
| 2 | ALL-AML-Leukemia | 100 | 300 |
| 3 | *MLL-Leukemia | 500 | 100 |
| 4 | Breast Cancer | 100 | 500 |
| 5 | Duke Breast Cancer | 200 | 500 |
| 6 | Prostate Cancer | 250 | 100 |
| 7 | Lung Cancer | 250 | 100 |
| 8 | Central Nervous System | 500 | 100 |
| 9 | Translation Initiation Site | 150 | 200 |
| 10 | Ovarian Cancer | 500 | 100 |
| 11 | Diffuse Large B-Cell Lymphoma | 150 | 200 |
| 12 | *Subtypes of Acute Lymphoblastic (Hyperdip) | 150 | 500 |
| 13 | *Subtypes of Acute Lymphoblastic (TEL-AML1) | 150 | 500 |
| 14 | *Subtypes of Acute Lymphoblastic (T-ALL) | 200 | 100 |
| 15 | *Subtypes of Acute Lymphoblastic (Others) | 500 | 500 |
| 16 | Bupa | 4 | 50 |
| 17 | Breast Cancer Wisconsin | 10 | 50 |
| 18 | Pima | 5 | 50 |
| 19 | *Segment | 10 | 50 |
| 20 | Spambase | 20 | 50 |
| 21 | *Opticdigits | 20 | 50 |
| 22 | *Satimage | 6 | 200 |
| 23 | *Pendigits | 8 | 50 |
| 24 | *Letters | 8 | 50 |
| 25 | *Shuttle | 5 | 50 |

**Table 4** Classification results on very-high-dimensional datasets

| Dataset | Precision | | | Recall | | | F1-measure | | | Accuracy | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ID | LibSVM | RF-C4.5 | RF-ODT | LibSVM | RF-C4.5 | RF-ODT | LibSVM | RF-C4.5 | RF-ODT | LibSVM | RF-C4.5 | RF-ODT |
| 1 | 68.18 | 76.19 | **82.61** | 75.00 | 72.73 | **86.36** | 71.43 | 74.42 | **84.44** | 80.65 | 82.26 | **88.71** |
| 2 | **100** | 95.24 | 95.24 | 95.00 | **100** | **100** | 97.44 | **97.56** | **97.56** | **97.06** | **97.06** | **97.06** |
| 3 | 75.00 | **100** | **100** | **100** | **100** | **100** | 85.71 | **100** | **100** | 93.33 | **100** | **100** |
| 4 | 69.23 | 83.33 | **84.62** | 75.00 | 83.33 | **91.67** | 72.00 | 83.33 | **88.00** | 63.16 | 78.94 | **84.21** |
| 5 | 85.00 | **94.12** | 90.00 | **94.44** | 80.00 | 90.00 | 89.47 | 86.49 | **90.00** | **90.48** | 88.10 | **90.48** |
| 6 | 73.53 | 75.76 | **100** | **100** | **100** | 96.00 | 84.75 | 86.21 | **97.96** | 73.53 | 76.47 | **97.06** |
| 7 | 88.26 | **93.75** | **93.75** | **100** | **100** | **100** | 93.75 | **96.77** | **96.77** | 98.66 | **99.33** | **99.33** |
| 8 | 47.62 | 45.46 | **61.91** | 55.56 | 23.81 | **61.91** | 51.28 | 31.25 | **61.91** | 68.33 | 63.33 | **73.33** |
| 9 | 83.13 | **92.58** | 90.78 | **84.42** | 73.83 | 79.75 | 83.77 | 82.15 | **84.91** | 92.15 | 92.30 | **93.20** |
| 10 | **100** | 98.78 | **100** | **100** | **100** | **100** | **100** | 99.39 | **100** | **100** | 99.21 | **100** |
| 11 | 91.30 | **95.65** | 92.00 | 87.50 | 91.67 | **95.83** | 89.36 | 93.62 | **93.88** | 89.36 | **93.62** | **93.62** |
| 12 | 95.46 | 95.24 | **100** | **95.46** | 90.91 | **95.46** | 95.46 | 93.02 | **97.67** | 98.21 | 97.32 | **99.11** |
| 13 | **100** | **100** | **100** | **100** | 96.30 | 96.30 | **100** | 98.11 | 98.11 | **100** | 99.11 | 99.11 |
| 14 | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** | **100** |
| 15 | 92.59 | **100** | **100** | 39.68 | 29.63 | **55.56** | 55.56 | 45.71 | **71.43** | 64.29 | 83.93 | **89.29** |

**Table 5** Accuracy comparison on very-high-dimensional datasets

| Accuracy | LibSVM | RF-C4.5 | RF-ODT | RF-ODT vs LibSVM | RF-ODT vs RF-C4.5 |
|---|---|---|---|---|---|
| mean | 87.28 | 90.07 | 93.63 | **6.35** | **3.57** |
| standard deviation | 13.65 | 22.31 | 21.69 | 9.41 | 6.32 |
| student ratio | | | | 2.61 | 2.19 |
| p-value | | | | **0.0204** | **0.0462** |
| result of RF-ODT | | | | gain* | gain* |
| RF-ODT win | | | | 10 | 9 |
| RF-ODT tie | | | | 4 | 6 |
| RF-ODT defeat | | | | 1 | 0 |
| p-value | | | | 0.0117 | 0.0039 |
| result of RF-ODT | | | | gain* | gain** |

LibSVM (p-value = 0.0204). The comparison dataset by dataset using the sign test shows that on the 15 datasets, RF-ODT systematically prevails on RF-C4.5 (9 wins, 6 ties, 0 defeat, p-value = 0.0039) and is beaten once only by LibSVM (10 wins, 4 ties, 1 defeat, p-value = 0.0117).

For a more detailed assessment of the performance of RF-ODT facing RF-C4.5 and LibSVM, in addition to the error rate, we also calculated the precision, the recall and the F1-measure (van Rijsbergen, 1979). The precision for a class is the number of datapoints correctly labeled as belonging to the class divided by the total number of datapoints labeled as belonging to the class. The recall for a class is the number of datapoints correctly labeled as belonging to the class divided by the total number of elements that actually belong to the class. The F1-measure is a synthesis of the precision and the recall, which is defined as the harmonic mean of these both quantities. Compared to the arithmetic mean, the harmonic mean has the particularity to be more sensitive to the minimum of the precision and the recall.

Regarding the comparison of RF-ODT with RF-C4.5, one can see that the gain ensured by RF-ODT is above all due to the increase of the recall (table 7) which is significantly improved of 7.1 percent points on average (Student p-value = 0.0296).

**Table 6** Precision comparison on very-high-dimensional datasets

| Precision | LibSVM | RF-C4.5 | RF-ODT | RF-ODT vs LibSVM | RF-ODT vs RF-C4.5 |
|---|---|---|---|---|---|
| mean | 84.62 | 89.74 | 92.73 | 8.11 | 2.99 |
| standard deviation | 15.30 | 14.69 | 10.38 | 9.26 | 7.68 |
| student ratio | | | | 3.39 | 1.51 |
| p-value | | | | **0.0044** | 0.1540 |
| result of RF-ODT | | | | gain** | |
| RF-ODT win | | | | 11 | 6 |
| RF-ODT tie | | | | 3 | 6 |
| RF-ODT defeat | | | | 1 | 3 |
| p-value | | | | **0.0063** | 0.5078 |
| result of RF-ODT | | | | gain** | |

**Table 7** Recall comparison on very-high-dimensional datasets

| Recall | LibSVM | RF-C4.5 | RF-ODT | RF-ODT vs LibSVM | RF-ODT vs RF-C4.5 |
|---|---|---|---|---|---|
| mean | 86.80 | 83.06 | 90.17 | 3.37 | 7.11 |
| standard deviation | 18.37 | 24.94 | 14.13 | 6.98 | 11.37 |
| student ratio | | | | 1.87 | 2.42 |
| p-value | | | | 0.0828 | **0.0296** |
| result of RF-ODT | | | | | gain* |
| RF-ODT win | | | | 6 | 8 |
| RF-ODT tie | | | | 6 | 6 |
| RF-ODT defeat | | | | 3 | 1 |
| p-value | | | | 0.5078 | 0.0391 |
| result of RF-ODT | | | | | gain* |

**Table 8** F1-measure comparison on very-high-dimensional datasets

| F1-measure | LibSVM | RF-C4.5 | RF-ODT | RF-ODT vs LibSVM | RF-ODT vs RF-C4.5 |
|---|---|---|---|---|---|
| mean | 84.67 | 84.54 | 90.84 | **6.18** | **6.31** |
| standard deviation | 15.61 | 27.15 | 22.65 | 6.90 | 9.88 |
| student ratio | | | | 3.47 | 2.47 |
| p-value | | | | **0.0038** | **0.0269** |
| result of RF-ODT | | | | gain** | gain* |
| RF-ODT win | | | | 12 | 10 |
| RF-ODT tie | | | | 2 | 5 |
| RF-ODT defeat | | | | 1 | 0 |
| p-value | | | | 0.0034 | 0.0020 |
| result of RF-ODT | | | | gain** | gain** |

The comparison of the recalls, dataset by dataset, shows that RF-ODT is beaten only once by RF-C4.5 (8 wins, 6 ties, 1 defeat, p-value $= 0.0391$). The empirical gain on the precision (table 6), which worth 3 percent points, due to the excellent performance of RF-ODT on the datasets 6 and 8, is not significant. The dataset by dataset results (6 wins, 6 ties, 3 defeat, p-value $= 0.5078$) support those comments. As results in table 8, the F1-measure obtained from RF-ODT is significantly improved

by 6.3 points on average compared with the F1-measure obtained from RF-C4.5 (p-value = 0.0269).

The comparison dataset by dataset gives a very significant advantage to RF-ODT (sign-test p-value = 0.0020) which is never defeated by RF-C4.5. Indeed, RF-ODT is the winner 10 times out of 15 and there is equality 5 times out of 15.

The comparison of RF-ODT with LibSVM gives an opposite result. The superiority of RF-ODT on LibSVM is mainly due to the increase of the precision. In fact, RF-ODT improves the LibSVM precision of 8.1 percent points on average (table 6), which is very significant (p-value = 0.0044). Out of the 15 datasets, RF-ODT is beaten only once by LibSVM (11 wins, 3 ties, 1 defeat, p-value = 0.0063). RF-ODT improves the LibSVM recall of 3.4 percent points on average (table 7), which is not quite significant. However, we note that the 6 wins (especially on datasets 1, 4 and 15) are larger than the 3 defeats. Overall, the F1-measure is improved (table 8) by 6.2 points on average (p-value = 0.0038), which is very significant. The dataset by dataset comparison supports this conclusion (12 wins, 2 ties, 1 defeat, p-value = 0.0034).

### *3.3 Classification Results on Standard Datasets*

It is interesting to complement the above experiments by comparing the accuracies of RF-ODT and RF-C4.5 on standard benchmarks. Ten datasets (table 2) are used, each of them having a number of variables comprised between 6 and 64 and a ratio between the number of dimensions and the number of datapoints which does not exceed 5%.

These experiments (table 9) suggest that RF-ODT is at least as effective as RF-C4.5 when the datasets are standard. Indeed, RF-ODT improves the RF-C4.5 accuracy of 0.6 percent points on average, but this advantage is tiny and not significant. The differences between the accuracies of RF-ODT and RF-C4.5 on each dataset are small, except for Bupa dataset where RF-ODT ensures an increase of 4 percent points. However, it must be noticed that RF-ODT wins 8 times out of 10 against RF-C4.5, which is almost significant (p-value = 0.0547).

**Table 9** Accuracy comparison of random forests on standard datasets

| Accuracy | RF-ODT vs RF-C4.5 |
|---|---|
| mean | **0.69** |
| standard deviation | 3.10 |
| student ratio | 0.70 |
| p-value | **0.5001** |
| result of RF-ODT | non significant |
| RF-ODT win | 8 |
| RF-ODT tie | 0 |
| RF-ODT defeat | 2 |
| p-value | 0.1094 |
| result of RF-ODT | almost significant |

**Table 10** Execution time

| Dataset | Execution time | | | Dataset | Execution time | | |
|---|---|---|---|---|---|---|---|
| ID | RF-C4.5 | RF-ODT | RF-C4.5/RF-ODT | ID | RF-C4.5 | RF-ODT | RF-C4.5/RF-ODT |
| 1 | 2.96 | **0.66** | 4,48 | 14 | **4.00** | 4.30 | 0,93 |
| 2 | 3.96 | **3.42** | 1,16 | 15 | **74.68** | 86.20 | 0,87 |
| 3 | **5.98** | 17.97 | 0,33 | 16 | **0.21** | 0.28 | 0,75 |
| 4 | 11.38 | **6.58** | 1,73 | 17 | 1.87 | **0.15** | 12,47 |
| 5 | 7.41 | **6.81** | 1,09 | 18 | 0.93 | **0.83** | 1,12 |
| 6 | 3.09 | **2.49** | 1,24 | 19 | 1.87 | **0.20** | 9,35 |
| 7 | **1.30** | 2.50 | 0,52 | 20 | 10.08 | 3.65 | 2,76 |
| 8 | **3.33** | 5.88 | 0,57 | 21 | 2.79 | **1.07** | 2,61 |
| 9 | 797.10 | **671.00** | 1,19 | 22 | 7.83 | **3.74** | 2,09 |
| 10 | 29.97 | **22.99** | 1,30 | 23 | 4.90 | **1.35** | 3,63 |
| 11 | 2.55 | **1.73** | 1,47 | 24 | 8.62 | **1.56** | 5,53 |
| 12 | 19.82 | **10.37** | 1,91 | 25 | 28.07 | **6.56** | 4,28 |
| 13 | 17.70 | **11.96** | 1,48 | Mean | 42.10 | **34.97** | 1,20 |

The whole experiments confirm the validity of our approach: RF-ODT at least match RF-C4.5 on standard datasets and outperforms RF-C4.5 on very high dimensional databases, which is the pursued aim.

## *3.4 Execution Time*

Before to compare the RF-ODT execution time and the RF-C4.5 one, let us analyse the theoretical complexity of the both algorithms. Considering $m$ datapoints and $n$ attributes, RF-C4.5 constructs the $k$ trees of the forest with the complexity $O(kqn'mlog(m))$, where $q = 1$ if the attributes are nominal, $q = 2$ if the attributes are numerical and $n' << n$. Our RF-ODT algorithm has complexity $O(kp(n'+m)n'^2)$, $p$ being the average depth of the different trees.

In practice, the size of oblique trees is smaller than the size of C4.5 ones. An oblique tree generally has a smaller depth. In addition, at each node of each RF-ODT tree, the subset of attributes is reduced. For these reasons, the RF-ODT execution time is faster than the RF-C4.5 one. The experiments to compare the RF-ODT and RF-C4.5 execution time were performed on a PC (Pentium 2,4 GHz, 1 Go RAM, Linux Mandriva 2008). Execution time is given in Table 10.

In addition, we evaluated our proposal on Forest cover type is which a very large dataset (Asuncion and Newman, 2007). It comprises 495141 datapoints for the learning set and 45141 datapoints for the test set. We constructed random forest of 30 trees to learn each of the two largest classes (Spruce-Fire: 211840 datapoints and Lorgepole-Pine: 283301 datapoints with 54 attributes). The learning time of RF-ODT is 801.61 seconds with a precision of 99.98%, while the learning time of RF-C4.5 is 17484 seconds with a precision of 99.57%. Therefore our RF-ODT is 22 times faster than the usual RF-C4.5, while it slightly improves the precision (0.41%).

## 4 Conclusion and Future Works

We presented random forests of oblique decision trees that achieve high performances for classification tasks. The main ideas are to use linear proximal SVMs for performing multivariate node splitting during tree construction, producing individual classifiers that are stronger than in a classical forests. Numerical test results on standard datasets and very-high-dimensional datasets have shown that our random forests of oblique decision trees algorithm is usually more accurate in terms of precision, recall, F1-measure, accuracy compared with random forests of C4.5 and SVM. It has significant better performance on very-high-dimensional data with better -but not significant- results on lower dimensional datasets. In addition our proposal is very efficient and it can be parallelized. A parallel implementation that exploits the multicore processors can greatly speed up the learning tasks.

Extension of the proposed approach for imbalanced datasets, multi-class classification, regression problems and feature selection tasks are under progress. In the near future we intend to provide more empirical test on large benchmarks and comparisons with other oblique trees methods.

## References

Amit, Y., Geman, D.: Shape Quantization and Recognition with Randomized Trees. Machine Learning 45(1), 5–32 (2001)

Asuncion, A., Newman, D.: UCI Repository of machine learning databases (2007), http://www.ics.uci.edu/~mlearn/{MLR}epository.html

Bennett, K.P., Mangasarian, O.L.: Multicategory Discrimination via Linear Programming. Optimization Methods and Software 3, 27–39 (1994)

Breiman, L.: Bagging Predictors. Machine Learning 24(2), 123–140 (1996)

Breiman, L.: Random Forests. Machine Learning 45(1), 5–32 (2001)

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.: Classification and Regression Trees. Wadsworth International, Belmont (1984)

Buntine, W.: Learning Classification Trees. Statistics and Computing 2, 63–73 (1992)

Carvalho, D., Freitas, A.: A hybrid decision tree/genetic algorithm method for data mining. Information Sciences 163(1-3), 13–35 (2004)

Chang, C.C., Lin, C.J.: LIBSVM – A Library for Support Vector Machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm

Cohen, S., Rokach, L., Maimon, O.: Decision-tree instance-space decomposition with grouped gain-ratio. Information Sciences 177(17), 3592–3612 (2007)

Cutler, A., Guohua, Z.: PERT – Perfect Random Tree Ensembles. Computing Science and Statistics 33, 490–497 (2001)

Dietterich, T.G.: Ensemble Methods in Machine Learning. In: Multiple Classifier Systems, pp. 1–15 (2000a)

Dietterich, T.G.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and Randomization. Machine Learning 40(2), 139–157 (2000b)

Do, T.-N., Lallich, S., Pham, N.-K., Lenca, P.: Un nouvel algorithme de forêts aléatoires d'arbres obliques particulièrement adapté à la classification de données en grandes dimensions. In: Ganascia, J.G., Gançarski, P. (eds.) Extraction et Gestion des Connaissances 2009, Strasbourg, France, pp. 79–90 (2009)

Do, T.N., Poulet, F.: Classifying one Billion Data with a New Distributed SVM Algorithm. In: Proceedings RIVF-2006: the 4th IEEE International Conference on Computer Science, Research, Innovation and Vision for the Future, pp. 59–66 (2006)

Freund, Y., Schapire, R.: A Decision-theoretic Generalization of On-line Learning and an Application to Boosting. In: Computational Learning Theory: Proceedings of the Second European Conference, pp. 23–37 (1995)

Fung, G., Mangasarian, O.: Proximal Support Vector Classifiers. In: Proceedings KDD 2001: Knowledge Discovery and Data Mining, pp. 77–86 (2001)

Geurts, P., Ernst, D., Wehenkel, L.: Extremely randomized trees. Machine Learning 63(1), 3–42 (2006)

Heath, D.: A Geometric Framework for Machine Learning. Ph.D. thesis, Johns Hopkins University, Baltimore (1992)

Ho, T.K.: Random Decision Forest. In: Proceedings of the Third International Conference on Document Analysis and Recognition, pp. 278–282 (1995)

Jinyan, L., Huiqing, L.: Kent Ridge Bio-medical Data Set Repository. Technical report (2002), http://datam.i2r.a-star.edu.sg/datasets/krbd/

Loh, W.-Y., Vanichsetakul, N.: Tree-structured classification via generalized discriminant analysis (with discussion). Journal of the American Statistical Association 83, 715–728 (1988)

Maji, P.: Efficient design of neural network tree using a new splitting criterion. Neurocomputing 71(4-6), 787–800 (2008)

Michie, D., Spiegelhalter, D.J., Taylor, C.C. (eds.): Machine Learning, Neural and Statistical Classification. Ellis Horwood (1994)

Murthy, S., Kasif, S., Salzberg, S.: A System for Induction of Oblique Decision Trees. Journal of Artificial Intelligence Research 2(1), 1–32 (1994)

Murthy, S., Kasif, S., Salzberg, S., Beigel, R.: OC1: Randomized Induction of Oblique Decision Trees. In: Proceedings of the Eleventh National Conference on Artificial Intelligence, pp. 322–327 (1993)

Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)

Robnik-Sikonja, M.: Improving Random Forests. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) ECML 2004. LNCS (LNAI), vol. 3201, pp. 359–370. Springer, Heidelberg (2004)

Rokach, L., Maimon, O.: Top-Down Induction of Decision Trees Classifiers - A Survey. IEEE Transactions on Systems, Man and Cybernetics. Part C: Applications and Reviews 35(4), 476–487 (2005)

Simon, C., Meessen, J., De Vleeschouwer, C.: Embedding proximal support vectors into randomized trees. In: European Symposium on Artificial Neural Networks, Advances in Computational Intelligence and Learning, pp. 373–378 (2009)

Suykens, J., Vandewalle, J.: Least Squares Support Vector Machines Classifiers. Neural Processing Letters 9(3), 293–300 (1999)

van Rijsbergen, C.V.: Information Retrieval. Butterworth (1979)

Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Heidelberg (1995)

Veropoulos, K.: Campbell, C. and Cristianini, N, Controlling the sensitivity of support vector machines. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 55–60 (1999)

Wolpert, D.: Stacked Generalization. Neural Networks 5, 241–259 (1992)
Wu, W., Bennett, K., Cristianini, N., Shawe-Taylor, J.: Large Margin Trees for Induction and Transduction. In: Proceedings of the Sixth International Conference on Machine Learning, pp. 474–483 (1999)
Xu, Q., Pei, W., Yang, L., He, Z.: Support Vector Machine Tree Based on Feature Selection. In: King, I., Wang, J., Chan, L., Wang, D.L. (eds.) ICONIP 2006. LNCS, vol. 4232, pp. 856–863. Springer, Heidelberg (2006)
Yildiz, O., Alpaydin, E.: Linear Discriminant Trees. International Journal of Pattern Recognition and Artificial Intelligence 19(3), 323–353 (2005)
Zhou, Z.-H., Chen, Z.-Q.: Hybrid decision tree. Knowledge-Based Systems 15(8), 515–528 (2002)

# Intensive Use of Correspondence Analysis for Large Scale Content-Based Image Retrieval

Nguyen-Khang Pham, Annie Morin, Patrick Gros, and Quyet-Thang Le

**Abstract.** In this paper, we investigate the intensive use of Correspondence Analysis (CA) for large scale content-based image retrieval. Correspondence Analysis is a useful method for analyzing textual data and we adapt it to images using the SIFT local descriptors. CA is used to reduce dimensions and to limit the number of images to be considered during the search step. An incremental algorithm for CA is proposed to deal with large databases giving exactly the same result as the standard algorithm. We also integrate the Contextual Dissimilarity Measure in our search scheme in order to improve response time and accuracy. We explore this integration in two ways: (i) off-line (the structure of image neighborhoods is corrected off-line) and (ii) on-the-fly (the structure of image neighborhoods is adapted during the search). The evaluation tests have been performed on a large image database (up to 1 million images).

**Keywords:** Correspondence Analysis, Large Scale Image Retrieval, Image Indexing, Incremental Algorithm, SIFT.

Nguyen-Khang Pham
University of Rennes I, IRISA, Campus de Beaulieu, 35042 RENNES Cedex,
France and Cantho University, 1 Ly Tu Trong, Cantho, Vietnam
e-mail: Pham.Nguyen_Khang@irisa.fr, pnkhang@cit.ctu.edu.vn

Annie Morin
University of Rennes I, IRISA, Campus de Beaulieu, 35042 RENNES Cedex, France
e-mail: Annie.Morin@irisa.fr

Patrick Gros
INRIA Rennes - Bretagne Atlantique, Campus de Beaulieu, 35042 RENNES Cedex, France
e-mail: Patrick.Gros@inria.fr

Quyet-Thang Le
Cantho University, 1 Ly Tu Trong, Cantho, Vietnam
e-mail: lqthang@cit.ctu.edu.vn

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 57–76.
springerlink.com 

## 1 Introduction

The goal of Content-Based Image Retrieval (CBIR) systems is to operate on collections of images and to extract relevant images in response to visual queries. This task is not easy because of two gaps: the *sensory gap* and the *semantic gap* (Smeulders *et al.*, 2000). While the *sensory gap* is the gap between the object in the real world and a picture of this same object which is subject to accidental distortions, background clutter, occlusion, etc., the *semantic gap* is the gap between low-level content and higher-level concepts.

Recently, the use of local descriptors has drastically increased the power of image analysis techniques. Global descriptors are computed on the whole image while local descriptors are extracted from specific points. Using local descriptors allows to find images which share one or several visual elements with the query. First, the methods based on a voting algorithm have been used for image retrieval. These methods describe an image as a *set of local descriptors* computed at some particular points called *interest points*. Given a query, each of its descriptors is matched independently with the descriptors of the images in the database, based on a distance which measures the similarity between the descriptors. Finally, the image similarity measure is computed by counting the number of matching descriptors (Amsaleg and Gros, 2001; Berrani *et al.*, 2003; Lowe, 1999, 2004a; Mikolajczyk and Schmid, 2001, 2004a; Mohr *et al.*, 1998; Schaffalitzky and Zisserman, 2003; Schmid and Mohr, 1997; Tuytelaars and Gool, 1999). Later, the methods initially developed for textual data analysis such as *tf***idf* weighting (Salton and Buckley, 1988), LSA (Latent Semantic Analysis) (Deerwester *et al.*, 1990), PLSA (Probabilistic Latent Semantic Analysis) (Hofmann, 1999), LDA (Latent Dirichlet Allocation) (Blei *et al.*, 2003) have been adapted to images (Bosch *et al.*, 2006; Lienhart and Slaney, 2007; Sivic *et al.*, 2005; Sivic and Zisserman, 2003). In textual data analysis, these methods use the bag-of-words model. The input of such methods is a two-way table, often called contingency table, crossing documents and words. Applying them to images requires to replace documents by images and to define "*visual words*".

Among these methods, LSA, PLSA, and LDA are very costly in time and in memory when dealing with huge image databases. As a consequence, we will focus, in this paper, on the use of Correspondence Analysis for large scale image retrieval and especially on four aspects: (i) CA provides a better distance between images compared to *tf***idf* and PLSA (Pham and Morin, 2008); (ii) we propose an incremental version of CA algorithm (which computes exactly the same results as the usual algorithm) to deal with very large databases; (iii) We introduce a very efficient retrieval scheme using a relevant indicator of CA: *the quality of representation*. This scheme is based on inverted files (Nistér and Stewénius, 2006; Sivic and Zisserman, 2003) that avoid comparing the query to all images in the database. Nevertheless, our inverted files are not built directly from visual words but from topics which have been found by CA; and (iv) we also present the integration of the Contextual Dissimilarity Measure (CDM) (Jegou *et al.*, 2007) in our retrieval scheme.

The organization of the paper is the following: section 2 deals with the construction of visual words and its use for image representation. Section 3 is devoted to a short presentation of CA. In section 4, we describe an incremental version of the CA algorithm and in section 5, we use the previous results to large scale image retrieval. Finally, we show some experimental results, before concluding.

## 2 Representation of Images

### *2.1 Construction of Visual Words*

The words computed from images are called *visual words* and form a vocabulary of $N$ words. This computation is made in two steps: (i) computation of local descriptors for a set of images and (ii) vector quantization of those descriptors into clusters which are called, by definition, *visual words*. The computation of the local descriptors in an image is also a two step process: *interest point detection* and *descriptor computation*.

- *Interest point detection* - The Harris detector (Harris and Stephens, 1988) has been used in Schmid and Mohr (1997) for image retrieval. The interest points extracted from this detector are rotationally invariant whereas scale invariant interest point detectors have been proposed in Lindeberg (1998); Lowe (1999, 2004b); Mikolajczyk and Schmid (2001). The interest points are extracted at different scales. The automatic scale selection is carried out by selecting the extrema of a function over scales (e.g., Difference-of-Gaussian or normalized Laplacian-of-Gaussian). To achieve invariance with respect to affine transformations, Mikolajczyk *et al.* have proposed an affine-adapted Harris detector and an iterative algorithm for detecting affine and scale invariant interest points (Mikolajczyk and Schmid, 2002, 2004a).
- *Descriptor computation* - The descriptor of each interest point is computed at the selected scale in the region around the interest point. Many different descriptors have been proposed in research literature: shape context (Belongie *et al.*, 2002), steerable filters (Freeman and Adelson, 1991), SIFT (Lowe, 2004b), PCA-SIFT (Ke and Sukthankar, 2004), GLOH (Mikolajczyk and Schmid, 2005). A comparative performance evaluation of various descriptors is reported in Mikolajczyk and Schmid (2005). Among descriptors, the SIFT descriptor is the most used due to its discriminating power (Bosch *et al.*, 2006; Lienhart and Slaney, 2007; Pham and Morin, 2008; Sivic *et al.*, 2005; Sivic and Zisserman, 2003; Willamowski *et al.*, 2004). Each SIFT descriptor is a 128-dimensional vector.

Once local descriptors have been computed, the next step is to quantize them into clusters using the well known $k$-means algorithm, though other methods (k-medoids, histogram binning, etc) could be used too. Each cluster corresponds to a visual word. After building the visual vocabulary, each local descriptor is assigned to the closest cluster. At the end of the process, an image is described by the list of visual words

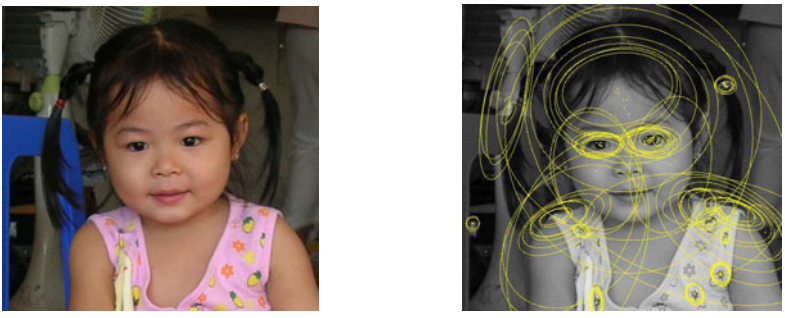


**Fig. 1** Detected interest points by an Hessian-Affine detector

its descriptors belong to. We then obtain a two-way table crossing the images and the visual words, called *contingency table*.

## 3 Correspondence Analysis

CA is an exploratory data analysis technique designed for simple two-way tables containing some measure of correspondence between the rows and the columns. It was developed by Benzécri (Benzecri, 1973; Greenacre, 1984; Lebart, 1984) for textual data analysis. CA applied on a table crossing documents and words allows to answer the following questions: Are there similarities between some words ? Between some documents ? Are there some relationships between words and documents ? CA such as factorial methods is based on the *eigen decomposition* of a matrix. The rows and columns of a data matrix (i.e. contingency table) are assumed to be points in a high-dimensional Euclidean space, and the method aims to redefine the dimensions of the space so that the principal dimensions capture the greatest variance possible (i.e. the inertia of the projected points is maximum), allowing for representations of both words and documents in a same lower-dimensional space (called *factor space*).

Besides, CA provides some relevant indicators: every word or document represented as a point in the low dimensional factor space can be characterized by its *contribution* to the inertia of an axis or its *quality of representation* (Greenacre, 2007; Morin, 2004).

Let $F = \{f_{ij}\}_{M,N}(f_{ij} \geq 0)$ be a contingency table with dimensions $M \times N$ ($N < M$, $M$ is the number of documents and $N$ is the number of words). We normalize $F$ and get $X = \{x_{ij}\}_{M,N}$ by:

$$s = \sum_{i=1}^{M} \sum_{j=1}^{N} f_{ij} \tag{1}$$

$$x_{ij} = \frac{f_{ij}}{s}\ ,\ \forall i = 1,2,\ldots,M; j = 1,2,\ldots,N \tag{2}$$

Let's note:

$$p_i = \sum_{j=1}^{N} x_{ij} \ , \ \forall i = 1,2,\ldots,M \quad q_j = \sum_{i=1}^{M} x_{ij} \ , \ \forall j = 1,2,\ldots,N \tag{3}$$

$$P = \begin{pmatrix} p_1 & & 0 \\ & p_2 & \\ & \ddots & \\ 0 & & p_M \end{pmatrix} \quad Q = \begin{pmatrix} q_1 & & 0 \\ & q_2 & \\ & \ddots & \\ 0 & & q_N \end{pmatrix} \tag{4}$$

In order to determine the best lower-dimensional space where the data are to be projected (images and visual words), we compute the eigenvalues and the eigenvectors of the matrix $V$ of dimension $(N \times N)$:

$$V = X^T P^{-1} X Q^{-1} \tag{5}$$

where $X^T$ is the transposed matrix of $X$.

We then obtain the eigenvalues $\lambda$ and the eigenvectors $\mu$:

$$\lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{pmatrix}, \ \mu = \begin{pmatrix} \mu_{11} & \mu_{12} & \ldots & \mu_{1N} \\ \mu_{21} & \mu_{22} & \ldots & \mu_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{N1} & \mu_{N2} & \ldots & \mu_{NN} \end{pmatrix}$$

where $1 = \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_N \geq 0$.

After removing the first eigenvalue (i.e. the trivial eigenvalue which is equal to one), we only keep the $K(K < N)$ largest positive eigenvalues and the associated eigenvectors[1]. These $K$ eigenvectors define an *orthonormal basis* of the $K$-dimensional space. The number of dimensions of the problem is thus reduced from $N$ to $K$. The documents are projected in the new space:

$$Z = P^{-1} X A \tag{6}$$

where $P^{-1}X$ represents the *row profiles*[2] and $A = Q^{-1}\mu$ is the *transition matrix* associated to CA. The words are also projected in the same space:

[1] Like other dimension reduction methods, $K$ is chosen empirically (e.g., by the way of cross-validation).

[2] CA is based on relative values. The sample size is not important for the construction of the *factor space*. The data table can be expressed as proportions (percentages) relative to the row or column margins. The rows (columns) containing the relative frequencies for the singles words (documents) are called *row profiles* (*column profiles*).

$$W = Q^{-1} X^T Z \, \mathbf{diag}(\lambda)^{-\frac{1}{2}} \tag{7}$$

$$\text{where } \mathbf{diag}(\lambda) = \begin{pmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ & & \ddots \\ 0 & & \lambda_K \end{pmatrix} \tag{8}$$

A new document (e.g., the query) $r = [r_1 \; r_2 \; \cdots \; r_N]$ will be projected in the *factor space* through the transition formula (6):

$$\hat{r}_i = \frac{r_i}{\sum_{j=1}^{N} r_j}, \; \forall i = 1, 2, \ldots, N \tag{9}$$

$$\text{and } Z_r = \hat{r} A \tag{10}$$

This CA will used to index images. By applying CA on the contingency table crossing the images and visual words (i.e. we consider images as documents and visual words as words), we obtain a new representation of images (i.e. CA-based representation): the matrix $Z$. The similarity of a query and an image is thus measured on the new representation. For instance, the cosine similarity between a query $r$ and the image $i$ of the database is computed by:

$$\text{similarity}(r, i) = \cos(Z_r, Z_i) \tag{11}$$

$$= \frac{\sum_{k=1}^{K} Z_{rk} Z_{ik}}{\|Z_r\| \|Z_i\|} \tag{12}$$

where $Z_{rk}$ and $Z_{ik}$ are respectively the coordinates of the query $r$ and the image $i$ on axis $k$, $\|Z_r\|$ and $\|Z_i\|$ are the Euclidean norm of $Z_r$ and $Z_i$ respectively.

## 4 Incremental CA Algorithm

As mentioned in section 3, the CA problem requires to compute the eigenvectors and eigenvalues of a particular matrix $V$ (formula 5). In the case of large scale databases, the matrix $X$ is too large to be stored entirely into memory. For instance, let us assume that each image is described by a 5 000-dimensional vector, then a database of one million images occupies an amount of $1\,000\,000 \times 5\,000 \times 4$ bytes $\approx 18$ GB in memory. Therefore, we need to find an incremental procedure for computing the matrix $V$, with the same result as the non-incremental algorithm.

First, let's rewrite formula 5:

$$V = V_0 Q^{-1} \quad \text{where} \tag{13}$$

$$V_0 = X^T P^{-1} X \tag{14}$$

The matrix $X$ is divided into blocks of rows. Suppose that there are $\mathbf{B}$ blocks, denoted by:

$$X = \begin{pmatrix} X_{[1]} \\ \vdots \\ X_{[\mathbf{B}]} \end{pmatrix}. \tag{15}$$

Then we compute $P_{[1]}, P_{[2]}, \ldots, P_{[\mathbf{B}]}$ and $Q_{[1]}, Q_{[2]}, \ldots, Q_{[\mathbf{B}]}$ in the same way for $Q$ and $P$ by replacing $X$ with $X_{[i]}$ for $i = 1, 2, \ldots, \mathbf{B}$. It is clear that:

$$P = \begin{pmatrix} P_{[1]} & & 0 \\ & P_{[2]} & \\ & & \ddots \\ 0 & & P_{[\mathbf{B}]} \end{pmatrix} \text{ and} \tag{16}$$

$$Q = \sum_{i=1}^{\mathbf{B}} Q_{[i]}. \tag{17}$$

If we denote:

$$V_{[i]} = X_{[i]}^T P_{[i]}^{-1} X_{[i]} \tag{18}$$

then

$$V_0 = \sum_{i=1}^{\mathbf{B}} V_{[i]}. \tag{19}$$

Both formulas 17 and 19 are the key parts for the incremental algorithm.

Once $V$ is constructed, the *eigen* problem is thus solved for a small matrix (e.g., size of $5\,000 \times 5\,000$). Since only a part of the eigenvectors is used for the projection stage, this problem can be solved efficiently by some advanced algorithms like LAPACK (Anderson *et al.*, 1999).

The mapping of images into the factor space can be done following the same algorithm: the new image representation $Z$ is computed by blocks. The main steps of incremental version of CA algorithm are described in Algorithm 1.

## 5 Large Scale Image Retrieval

In this section, we describe the two main contributions of our work: an efficient search scheme using inverted files, and the integration of the Contextual Dissimilarity Measure in our retrieval scheme. These contributions make the retrieval robust and efficient.

**Algorithm 1.** Incremental CA algorithm

```
Q = 0
V_0 = 0
for i = 1 to B do
    load block X_[i] into memory
    compute P_[i], Q_[i] from X_[i]
    Q   = Q + Q_[i]
    V_[i] = X_[i]^T P_[i]^{-1} X_[i]
    V_0 = V_0 + V_[i]
V = V_0 Q^{-1}
compute K eigenvalues λ and eigenvectors μ of V
compute the transition matrix A = Q^{-1} μ
for i = 1 to B do
    load block X_[i] into memory
    compute P_[i] from X_[i]
    Z_[i] = P_[i]^{-1} X_[i] A
```

## 5.1 Retrieval Scheme

We use the method described in Pham and Morin (2008) to accelerate the retrieval stage. It is a *two phase* algorithm. The first step consists in filtering non relevant images and the second step performs a sequential search in a *list of candidates*.

The main idea is based on a relevant indicator of CA: the *quality of representation* of an image on an axis. It has been shown that CA applied to images allows to build a correspondence between axes and image topics[3]. The better the representation of an image by an axis, the closer the relation between this image and the corresponding topic. The relevance of the database images with respect to a particular query is proportional to the number of topics they share with the query.

This filtering of non relevant images is achieved by using inverted files based on the *quality of representation* of images.

**Definition 1 (Quality of representation).** The *quality of representation* of an image $i$ on the axis $j$ is the *squared cosine* of the angle between the axis $j$ and the vector which joins the gravity center of the cloud of points-images to the image $i$:

$$\cos_j^2(i) = \frac{Z_{ij}^2}{\sum_{k=1}^{K} Z_{ik}^2} \tag{20}$$

where $Z_{ij}$ is the coordinate of the image $i$ on axis $j$ in the factor space.

[3] An image topic corresponds to a group of homogeneous images or a group of images sharing a common subject.

The closer the squared cosine of an image on a certain axis is to 1, the closer its projection on this axis is to its real position in the original space. A low *quality of representation* means that the current axis does not represent the considered image very well.

**Definition 2 (Inverted file).** Given a threshold $\varepsilon > 0$, an *inverted file* $F_j^+$ ($F_j^-$) associated to the positive (respectively negative) part of the axis $j$ is a set of images having a *quality of representation* superior to $\varepsilon$ and lying in the positive (respectively negative) part of the axis $j$.

$$F_j^+ = \{i \mid \cos_j^2(i) > \varepsilon \text{ and } Z_{ij} > 0\} \tag{21}$$

$$F_j^- = \{i \mid \cos_j^2(i) > \varepsilon \text{ and } Z_{ij} < 0\} \tag{22}$$

The threshold $\varepsilon$ is set to the average *quality of representation* (i.e. $\frac{1}{K}$ where $K$ is the number of axes). The number of inverted files is $2K$.

Note that the two parts of an axis (positive and negative) are often very different, even opposite (cf. Fig. 2).

**Algorithm 2.** Search algorithm using inverted files

1 Project $r$ into factor space (cf. Formula 10)
2 Determine topics to which $r$ belongs et take corresponding inverted files
3 Merge selected inverted files and compute the number of topics that images share with $r$
4 Filter non relevant images to construct a *list of candidates*
5 Search $k$ nearest neighbors of $r$ in the candidate list

Given a query $r$, the search procedure using inverted files is described in Algorithm 2. The search begins by filtering non relevant images. $r$ is first projected into the factor space and $Z_r$ is obtained by (10). Then, the representation quality factors of the query with respect to all axes are computed and the axes corresponding to the greater factors are selected. The inverted files attached to the selected axes are merged. As a consequence, the number of topics that an image $i$ shares with the query $r$ is equal to the number $t_i$ of occurrences of $i$ in the merged list. Thus it is possible to filter images that are less relevant, by eliminating those with a low $t_i$. The set of remaining images after the filtering step is the so-called *list of candidates* whose size is much smaller than that of the original database. Finally, the last retrieval stage considers only this *list of candidates*. This ensures the efficiency of the method.

In the previous algorithm, the number $t_i$ plays a key role with respect to the presence or absence of an image in the *list of candidates*. The choice of an appropriated threshold is done experimentally using a majority vote technique (Pham and Morin, 2008). In this case, an image $i$ will be kept in the *list of candidates* if it shares at

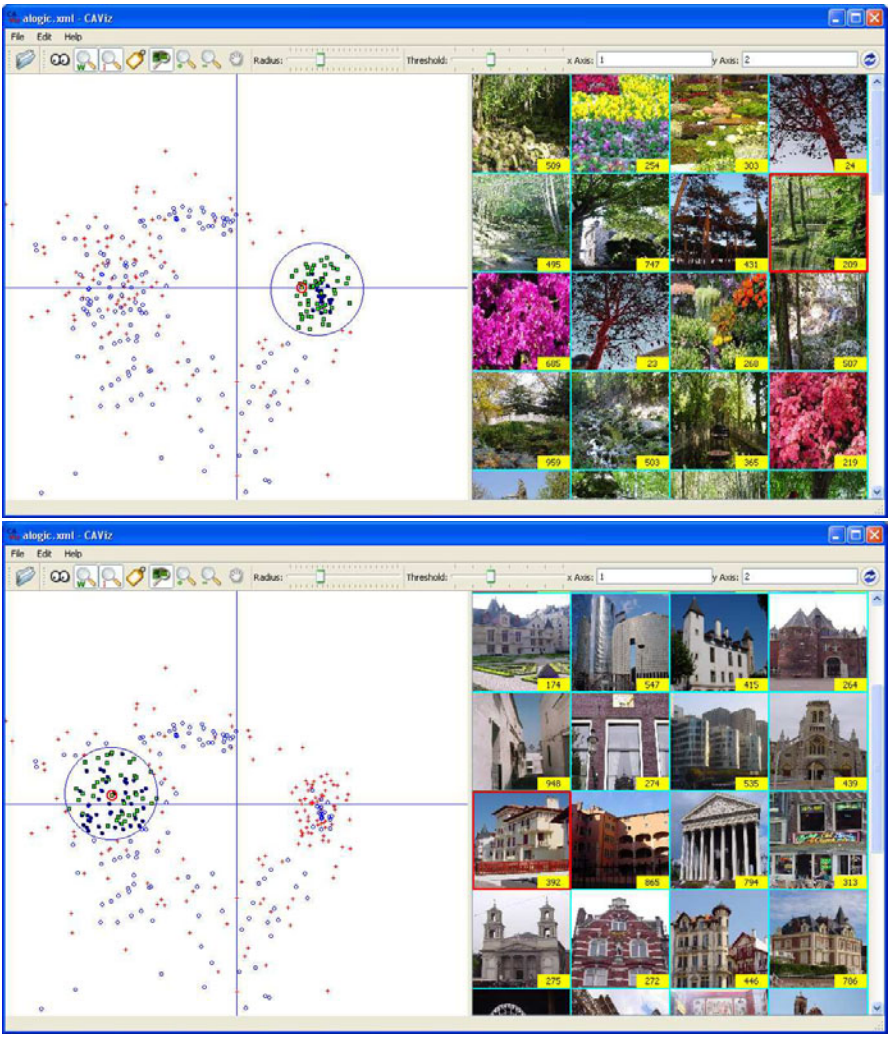



**Fig. 2** Two corresponding image topics of an axis: natural landscape (positive part) and building (negative part)

least half of its topics with the query. In case of very numerous axes, this criterion becomes too severe, possibly leading to an empty *list of candidates*.

This filtering rule should be weakened in the context of $k$-nearest neighbor searches. In this context, we return the $k$ nearest images of the query to the user who evaluates their relevance. After the merging step, the images sharing no topic with the query are rejected. However, the merged list remains very large and needs to be reduced further. The possibility that an image is the nearest neighbor of the

query is estimated by its $t_i$. Therefore, the $t_i$'s can be considered as a rough measure of similarity, allowing us to narrow down the merged list to a *list of candidates*.

The threshold $\theta$ (i.e. images whose $t_i < \theta$ will be filtered) can be chosen such that the size of the *list of candidates* is greater than a given value, **min-size**, which depends on the database's size and/or on $k$ (e.g., 0.1% of database's size, $50k$ or $100k$). Searching the $k$ nearest neighbors of the query is finally achieved by a sequential search in the *list of candidates*.

Another use of the $t_i$'s is the *iterative search* with user interaction where the search engine first takes some images with high $t_i$, performs a refined search, and presents the most relevant images to the user; at the next iteration, images with lower $t_i$ will be taken into account and merged with the remaining images of the previous iteration; a refined search will be carried out and so on... until the user stops the search.

## 5.2 Contextual Dissimilarity Measure

The Contextual Dissimilarity Measure (CDM) proposed in Jegou *et al.* (2007) is based on the integration of contextual information in the retrieval process. This measure takes into account the neighborhood structure of the points (or the images in the image retrieval context) to improve the quantity defined hereafter, and referred to as the *neighborhood symmetry rate*, of the $k$ nearest neighbors ($k$-NN) search scheme and as a consequence to improve the dissimilarity measure between images. This regularization is performed in the spirit of a local Mahalanobis distance for each image.

Let us consider the neighborhood $\mathcal{N}(i)$ of a given image $i$ obtained by a particular search framework ($\varepsilon$-search or $k$-NN search) and $|\mathcal{N}(i)|$ the cardinal of this set (which is a constant within the $k$-NN framework). The *neighborhood symmetry rate* of the search framework is obtained by:

$$S = \frac{1}{M} \sum_{i=1}^{M} \left( \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \mathrm{sym}(i,j) \right) \tag{23}$$

where $M$ is the number of images in the database and the $\mathrm{sym}(i,j) = 1$ if $i$ is a neighbor of $j$ and $j$ is a neighbor of $i$, 0 otherwise.

By definition, the *neighborhood symmetry rate* is maximized in the $\varepsilon$-search framework due to the distance symmetry property.

In order to improve the *neighborhood symmetry rate* of the $k$-NN search, each image is associated with a weight inversely proportional to the density of its neighborhood. This weighting scheme favors isolated images and penalizes images lying in dense areas.

Let the neighborhood $\mathcal{N}(i)$ of an image $i$, obtained by a $k$-NN search framework, be the set of its $k$ nearest neighbors ($\forall i, |\mathcal{N}(i)| = k$). The *neighborhood distance* $d_n(i)$ is, by definition, the mean distance of the image $i$ to all its neighbors:

$$d_n(i) = \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} d(i,j) \tag{24}$$

where $d(i,j)$ is the distance (or any dissimilarity measure) between two images $i$ and $j$.

The quantity $d_n(i)$ is computed for all images. The contextual dissimilarity measure $d_{CDM}(i,j)$ between two images $i$ and $j$ is defined by:

$$d_{CDM}(i,j) = d(i,j)\left(\frac{\bar{d_n}^2}{d_n(i)d_n(j)}\right)^{\alpha} \tag{25}$$

where $0 < \alpha < 1$ is a smoothing factor and $\bar{d_n}$ is the geometric mean of the neighborhood distances obtained by:

$$\bar{d_n} = \left(\prod_{i=1}^{M} d_n(i)\right)^{\frac{1}{M}} \tag{26}$$

where $M$ is the number of images in the database.

It is possible to use the arithmetic mean instead of the geometric mean. This leads to similar results.

Let us rewrite formula 25:

$$\begin{aligned} d_{CDM}(i,j) &= d(i,j)\left(\frac{\bar{d_n}^2}{d_n(i)d_n(j)}\right)^{\alpha} \\ &= d(i,j)\left(\frac{\bar{d_n}}{d_n(i)}\right)^{\alpha}\left(\frac{\bar{d_n}}{d_n(j)}\right)^{\alpha} \\ &= d(i,j)\delta(i)\delta(j) \end{aligned} \tag{27}$$

$$\text{where} \quad \delta(i) = \left(\frac{\bar{d_n}}{d_n(i)}\right)^{\alpha} \tag{28}$$

The $k$ nearest neighbors of a query $r$ are thus obtained by:

$$k\text{-NN}(r) = k\text{-}\underset{j}{\text{argmin}}\{d(j,r)\delta(j)\} \tag{29}$$

The $\delta(i)$'s are called *regularization terms* (or *distance update terms*) and stored within the database.

## 5.3 Integration of CDM in Retrieval Scheme

When the database becomes large, one of the disadvantages of CDM is the computation complexity of the regularization terms $\delta(i)$ which is quadratic with respect to the number of images in the database. To overcome this problem, Jegou *et al.* (2007) proposed a retrieval scheme using a clustering as a pre-processing step. The neighborhood of an image $i$ is then searched in a small number of clusters only (e.g., in 1% of the database), those whose centers (or centroids) are the nearest to $i$.

However, the choice of such a small number of clusters based on their center makes the approximate neighborhood far from the ideal because the number of neighboring clusters of an image increases exponentially with the space dimension. Therefore, if the number of selected clusters is too small, the risk of obtaining a bad neighborhood increases and the accuracy decreases.

We propose here an approach that dynamically selects a group of potential neighbors of a given image using the inverted files described in 5.1 for computing regularization terms. The approach allows us to find a good neighborhood for an image by examining only a very small group of points (e.g., 0.05% of the database). As a consequence, the computation complexity is considerably reduced while preserving the accuracy. There are two possible implementations:

- off-line: as used in Jegou *et al.* (2007), regularization terms $\delta(i)$'s are computed once before the search. This solution is appropriated to static databases where updating is rare.
- on-the-fly: during the search, regularization terms are dynamically computed on the *list of candidates* only. In this way, updating is no longer a problem.

# 6 Numerical Results

## *6.1 Image Datasets*

We performed experiments on the Nistér Stewénius dataset, namely N-S dataset (Nistér and Stewénius, 2006). This dataset consists of 2 550 scenes taken from 4 different viewpoints. Hence the dataset contains a total of 10 200 images. Figure 3 shows some images from the dataset.

We used the software **extract_feature** of Mikolajczyk and Schmid (2004b) to extract and compute local descriptors (Hessian-affine interest point detector and SIFT descriptors). The number of visual words is fixed to 5 000. The choice of $N$ (5000 in this case) is done empirically. We have experimented with different values of $N$

**Fig. 3** Images from the Nistér Stewénius dataset

(1000, 2000, 5000, and 10 000). The result slightly improved (but not very much) when $N$ increases.

To evaluate the scalability of our approach, we merged the N-S dataset with one million images downloaded from FlickR.

## 6.2 Baseline Methods

The *tf*idf* weighting (used in Sivic and Zisserman, 2003) is considered as a baseline method. We compared the CA-based search to the PLSA-based approach (Hofmann, 1999). In this case, an image is represented by its topic distribution, $P(topic \mid image)$[4].

## 6.3 Evaluation Metrics

To evaluate the method performance, we compute the *precision* for the first 3 retrieved images (P@3) because there are only 3 relevant images for a given query. We also present the Mean Average Precision (*MAP*) and the Mean Average Normalized Rank (*MANR*) of the methods.

Average Precision (*AP*) is a single valued measure computed as the area under the *precision–recall* graph and reflects performance over all recall levels. *MAP* is an algebraic mean of the *AP*'s for all queries.

The average normalized rank (ANR) for a particular query is obtained by:

$$ANR = \frac{1}{M * M_{rel}} \left( \sum_{i=1}^{M_{rel}} rank(i) - \frac{M_{rel}(M_{rel}+1)}{2} \right) \tag{30}$$

where $M$ is the number of images in the dataset; $M_{rel}$ is the number of relevant images for the query and $rank(i)$ is the rank of the $i^{th}$ relevant image. In essence, *ANR* is zero if all $M_{rel}$ relevant images are returned first. The *ANR* measure lies in the range 0 to 1, with 0.5 corresponding to random retrieval. Similar to MAP, MANR is obtained by averaging the *ANR*'s.

We also report the N-S score Nistér and Stewénius (2006). The N-S score is the number of relevant images among first 4 returned images (including the image used for the query).

## 6.4 CA versus Other Methods

Table 1 shows the performance of the different methods: *tf*idf*, PLSA and CA with different $K$ (the number of topics in the case of PLSA or the number of considered axes in the case of CA). For all methods, we perform an exhaustive search using the

[4] An implementation in Matlab of PLSA (by J. Verbeek) is found at `http://lear.inrialpes.fr/~verbeek/software.php`. The number of iterations of the EM algorithm for all experimentations is fixed to 100.

**Table 1** Performance of different methods on N-S dataset. PLSA-1: PLSA with the cosine similarity and PLSA-2: PLSA with the J-S measure

| Methods | N-S score | P@3 | MAP | MANR |
|---|---|---|---|---|
| *tf*idf* | 2.785 | 0.595 | 0.632 | 0.020 |
| PLSA-1, $K = 100$ | 2.825 | 0.608 | 0.651 | 0.010 |
| PLSA-1, $K = 200$ | 2.833 | 0.611 | 0.656 | 0.011 |
| PLSA-1, $K = 300$ | 2.745 | 0.582 | 0.628 | 0.011 |
| PLSA-1, $K = 400$ | 2.706 | 0.569 | 0.613 | 0.012 |
| PLSA-1, $K = 500$ | 2.631 | 0.544 | 0.591 | 0.014 |
| PLSA-2, $K = 100$ | 3.122 | 0.707 | 0.742 | 0.009 |
| PLSA-2, $K = 200$ | 3.154 | 0.718 | 0.762 | 0.009 |
| PLSA-2, $K = 300$ | 3.116 | 0.705 | 0.753 | 0.010 |
| PLSA-2, $K = 400$ | 3.059 | 0.686 | 0.740 | 0.011 |
| PLSA-2, $K = 500$ | 3.061 | 0.687 | 0.738 | 0.012 |
| AFC, $K = 100$ | 3.123 | 0.708 | 0.747 | 0.007 |
| AFC, $K = 200$ | 3.195 | 0.732 | 0.770 | **0.006** |
| AFC, $K = 300$ | 3.217 | 0.739 | 0.776 | **0.006** |
| AFC, $K = 400$ | 3.222 | 0.741 | 0.777 | **0.006** |
| AFC, $K = 500$ | **3.225** | **0.742** | **0.778** | **0.006** |
| AFC, $K = 600$ | **3.225** | **0.742** | **0.778** | **0.006** |

cosine similarity for measuring the similarity of images. We have also tested PLSA with another similarity measure: the Jensen-Shannon divergence (J-S measure). This measure is based on the Kullback Leibler divergence (K-L divergence). The J-S measure of two distributions $\mathbf{x}$ et $\mathbf{y}$ is obtained by:

$$d_{\mathrm{JS}}(\mathbf{x},\mathbf{y}) = \frac{1}{2}\left(d_{\mathrm{KL}}(\mathbf{x},\frac{\mathbf{x}+\mathbf{y}}{2}) + d_{\mathrm{KL}}(\mathbf{y},\frac{\mathbf{x}+\mathbf{y}}{2})\right) \tag{31}$$

where $d_{\mathrm{KL}}$ is the K-L divergence:

$$d_{\mathrm{KL}}(\mathbf{x},\mathbf{y}) = \sum_i x_i \log\frac{x_i}{y_i} \tag{32}$$

PLSA and CA considerably improve the quality of the results compared to the simple *tf*idf* weighting. For PLSA, the J-S measure gives better results than the cosine similarity. However, the computation time for J-S measure is much higher than the one for the cosine similarity (about 50-60 times slower due to the computation of logarithm functions). CA performs better than PLSA whatever the number of topics $K$.

## *6.5 Large Scale Evaluation*

For large scale evaluation, we have successively merged 10 200 images of the N-S dataset with 100 000, 200 000, 500 000 and one million images from FlickR. We

**Table 2** CA without Contextual dissimilarity measure, in comparison to the *tf*idf* clustering-based method

| Database size | min-size | recall | CA | | *tf*idf* | |
|---|---|---|---|---|---|---|
| | | | P@3 | time (ms) | P@3 | time (ms) |
| 100K + 10200 | 200 | 0.847 | 0.676 | 14.8 | 0.609 | 34.3 |
| | 300 | 0.872 | 0.681 | 15.1 | | |
| | 500 | 0.896 | 0.684 | 15.7 | | |
| 200K + 10200 | 200 | 0.841 | 0.660 | 24.6 | 0.596 | 71.4 |
| | 300 | 0.863 | 0.663 | 25.0 | | |
| | 500 | 0.885 | 0.664 | 25.5 | | |
| 500K + 10200 | 200 | 0.811 | 0.637 | 62.9 | 0.576 | 179.2 |
| | 300 | 0.832 | 0.640 | 63.8 | | |
| | 500 | 0.856 | 0.641 | 65.5 | | |
| 1M + 10200 | 200 | 0.779 | 0.623 | 172.3 | 0.559 | 355.9 |
| | 300 | 0.800 | 0.625 | 172.6 | | |
| | 500 | 0.823 | 0.627 | 172.8 | | |

compute the precision for the first 3 returned images (P@3) of the *tf*idf* and CA methods. Due to memory limitation, we could not perform PLSA on these large datasets. In these experimentations, we use our approximative search method (based on inverted files) instead of a sequential search and we compare with a clustering-based method described in Jegou *et al.* (2007) where the dataset is organized in 500 clusters (using *k*-medoids algorithm). For a given query, a sequential search is performed on 50 nearest clusters (10% of the dataset).

**CA without CDM**

Table 2 shows the results of both *tf*idf* and CA methods using the cosine similarity. The "min-size" column provides the minimum size of the *list of candidates* used to determine the threshold $\theta$ (cf. Section 5.1); the "recall" column represents the *recall* rate of the *list of candidates* (the number of relevant images divided by 3). The last columns provide the results (P@3 and response time) obtained with two methods:

1. CA retrieval using inverted files
2. *tf*idf* retrieval using clustering technique

It is clear that our method outperforms the clustering-based method in terms of both precision and response time. With a database of million images, a list of 500 candidate images (0.06% of the database) contains 82.3% of relevant images. This explains why our method performs efficiently without any reduction in the quality of the results.

**CA with CDM**

As shown in table 2, the percentage of relevant images in the lists of candidates ("recall" column) is relatively high. This means that the precision can be improved

**Table 3** Combination of CA and CDM. CA-1: combination of CA and off-line CDM; CA-2: CA with on-the-fly CDM and *tf*idf* + CDM: combination of CDM and clustering.

| Database size | min-size | CA-1 | CA-2 | *tf*idf* + MDC | time |
|---|---|---|---|---|---|
| 100K + 10200 | 200 | 0.735 | 0.735 | 0.696 | 25.8 |
| | 300 | 0.737 | 0.744 | | 36.7 |
| | 500 | 0.741 | 0.749 | | 71.5 |
| 200K + 10200 | 200 | 0.721 | 0.725 | 0.683 | 35.9 |
| | 300 | 0.724 | 0.731 | | 46.3 |
| | 500 | 0.725 | 0.735 | | 81.2 |
| 500K + 10200 | 200 | 0.699 | 0.705 | 0.660 | 75.1 |
| | 300 | 0.701 | 0.711 | | 86.5 |
| | 500 | 0.705 | 0.715 | | 121.3 |
| 1M + 10200 | 200 | 0.677 | 0.688 | 0.643 | 183.5 |
| | 300 | 0.684 | 0.696 | | 195.3 |
| | 500 | 0.688 | 0.701 | | 231.2 |

if an appropriate measure is used. This is why we integrate CDM into our retrieval scheme.

We show the precision for the first 3 returned images in table 3. CA-1 refers the off-line CDM integration and CA-2 to the CDM on-the-fly. The "*tf*idf* + CDM" is the method used in Jegou *et al.* (2007) which combines CDM and clustering. The response time for the CA with on-the-fly CDM is shown in "time(ms)" column. The response time for the CA-1 and "*tf*idf* + CDM" methods remains unchanged compared to the methods without CDM in Table 2 because regularization terms are computed off-line. With CA-1, we fixed **min-size** to 300 for the computation of the regularization terms. While with CA-2, the regularization terms are computed for all images of each *list of candidates* whose size varies from 200 to 500 as shown in the "min-size" column. For the sake of comparison, we set $|\mathcal{N}(i)|$ to 10 and $\alpha$ to 0.6 as proposed in Jegou *et al.* (2007). The results show that our method (off-line or on-the-fly) outperforms the clustering-based method. CA-2 still performs faster than "*tf*idf* + CDM" although regularization terms are computed on-the-fly. A possible explanation is the fact that the lists of candidates contain good neighbors. Therefore (i) approximate regularization terms are very close to ideal ones (which are computed exhaustively) while the clustering-based method does not lead to good regularization terms and (ii) A small number of clusters does not contain enough relevant images for a query and thus the precision decreases.

## 7 Conclusion and Future Work

We have presented, in this paper, an intensive use of Correspondence Analysis for large scale content-based image retrieval. In a first step, CA is used instead of *tf*idf* to improve the computation of distances between images. Then in order to deal with large scale databases, we have proposed an incremental version of CA algorithm

that loads only a small block of data into memory at an instant $t$. The incremental algorithm provides exactly the same results as the non incremental one. Next, one of the indicators of CA, the *quality of representation*, is used to build a retrieval scheme based on inverted files that avoids the comparison of the query with all the images in the database. Finally we proposed an improvement of this retrieval scheme by integrating the Contextual Dissimilarity Measure (Jegou *et al.*, 2007). The numerical results show that:

- CA is more effective than *tf*idf* and PLSA,
- results are still improved (better precision) as the Contextual dissimilarity measure is integrated in our retrieval scheme, either in the off-line or on-the-fly ways.

With the proposed method, only 0.06% of database is explored (in less than one eight of a second) and these 0.06% contain 82.3% of the relevant images. A first perspective would be to parallelize the CA algorithm. Another one is to explore the use of this method with very high dimensional data ($> 100\,000$).

## References

Amsaleg, L., Gros, P.: Content-based Retrieval Using Local Descriptors: Problems and Issues from a Database Perspective. Pattern Analysis and Applications, Special Issue on Image Indexation 4(2-3), 108–124 (2001)

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D.: LAPACK Users' Guide, 3rd edn. Society for Industrial and Applied Mathematics, Philadelphia (1999) ISBN 0-89871-447-8 (paperback)

Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(4), 509–522 (2002)

Benzecri, J.P.: L'Analyse de données: L'Analyse des correspondances. Dunod, Paris (1973)

Berrani, S.A., Amsaleg, L., Gros, P.: Robust content-based image searches for copyright protection. In: Proceedings of the ACM International Workshop on Multimedia Databases (MMDB 2003), pp. 70–77. ACM, New York (2003)

Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. Journal of Machine Learning Research 3, 993–1022 (2003)

Bosch, A., Zisserman, A., Munoz, X.: Scene Classification via pLSA. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3954, pp. 517–530. Springer, Heidelberg (2006)

Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harsman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science 41(6), 391–407 (1990)

Freeman, W., Adelson, E.: The Design and Use of Steerable Filters. IEEE Transactions on Pattern Analysis and Machine Intelligence 13(9), 891–906 (1991)

Greenacre, M.J.: Theory and Application of correspondence analysis. Academic Press, London (1984)

Greenacre, M.J.: Correspondence analysis in practice, 2nd edn. Chapman and Hall, Boca Raton (2007)

Harris, C., Stephens, M.: A combined corner and edge detector. In: Alvey Vision Conference, pp. 147–151 (1988)

Hofmann, T.: Probabilistic latent semantic analysis. In: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI 1999), pp. 289–296 (1999)

Jegou, H., Harzallah, H., Schmid, C.: A contextual dissimilarity measure for accurate and efficient image search. In: Proceedings of CVPR 2007, pp. 1–8 (2007)

Ke, Y., Sukthankar, R.: PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In: Proceedings of the Conference on Computer Vision and Pattern Recognition, pp. 511–517 (2004)

Lebart, L.: Multivariate Descriptive Statistical Analysis (Probability & Mathematical Statistics). John Wiley & Sons Inc., Chichester (1984)

Lienhart, R., Slaney, M.: pLSA on large scale image databases. In: IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1217–1220 (2007)

Lindeberg, T.: Feature detection with automatic scale selection. International Journal of Computer Vision 30(2), 79–116 (1998)

Lowe, D.G.: Object recognition from local scale-invariant features. In: Proceedings of the 7th International Conference on Computer Vision, Kerkyra, Greece, pp. 1150–1157 (1999)

Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004a)

Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 91–110 (2004b)

Mikolajczyk, K., Schmid, C.: Indexing based on scale invariant interest points. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV 2001), vol. 1, pp. 525–531 (2001)

Mikolajczyk, K., Schmid, C.: An affine invariant interest point detector. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 128–142. Springer, Heidelberg (2002)

Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. Proceedings of IJC V 60(1), 63–86 (2004a)

Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. International Journal of Computer Vision 60(1), 63–86 (2004b)

Mikolajczyk, K., Schmid, C.: A Performance Evaluation of Local Descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(10), 1615–1630 (2005)

Mohr, R., Gros, P., Schmid, C.: Efficient matching with invariant local descriptors. In: Amin, A., Pudil, P., Dori, D. (eds.) SPR 1998 and SSPR 1998. LNCS, vol. 1451, pp. 54–71. Springer, Heidelberg (1998)

Morin, A.: Intensive Use of Correspondence Analysis for Information Retrieval. In: Proceedings of the 26th International Conference on Information Technology Interfaces, ITI 2004, pp. 255–258 (2004)

Nistér, D., Stewénius, H.: Scalable recognition with a vocabulary tree. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 2161–2168 (2006)

Pham, N.-K., Morin, A.: Une nouvelle approche pour la recherche d'images par le contenu. In: Revue des Nouvelles Technologies de l'Information - Serie Extraction et gestion des connaissances, vol. RNTI-E-11, pp. 475–486 (2008)

Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5), 513–523 (1988)

Schaffalitzky, F., Zisserman, A.: Automated Location Matching in Movies. Computer Vision and Image Understanding 92, 236–264 (2003)

Schmid, C., Mohr, R.: Local Grayvalue Invariants for Image Retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(5), 530–535 (1997)

Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering objects and their location in image collections. In: Proceedings of the International Conference on Computer Vision, pp. 370–377 (2005)

Sivic, J., Zisserman, A.: Video Google: A Text Retrieval Approach to Object Matching in Videos. In: Proceedings of the International Conference on Computer Vision, vol. 2, pp. 1470–1477 (2003)

Smeulders, A.W.M., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-Based Image Retrieval at the End of the Early Years. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12), 1349–1380 (2000)

Tuytelaars, T., Gool, L.J.V.: Content-Based Image Retrieval Based on Local Affinely Invariant Regions. In: Huijsmans, D.P., Smeulders, A.W.M. (eds.) VISUAL 1999. LNCS, vol. 1614, pp. 493–500. Springer, Heidelberg (1999)

Willamowski, J., Arregui, D., Csurka, G., Dance, C.R., Fan, L.: Categorizing Nine Visual Classes Using Local Appearance Descriptors. In: Proceeding of the ICPR Workshop on Learning for Adaptable Visual Systems (2004)

# Toward a Better Integration of Spatial Relations in Learning with Graphical Models

Emanuel Aldea and Isabelle Bloch

**Abstract.** This paper deals with structural representations of images for machine learning and image categorization. The representation consists of a graph where vertices represent image regions and edges spatial relations between them. Both vertices and edges are attributed. The method is based on graph kernels, in order to derive a metrics for comparing images. We show in particular the importance of edge information (i.e. spatial relations) in the specific context of the influence of the satisfaction or non-satisfaction of a relation between two regions. The main contribution of the paper is situated in highlighting the challenges that follow in terms of image representation, if fuzzy models are considered for estimating relation satisfiability.

**Keywords:** Image Interpretation, Spatial Relations, Fuzzy Reasoning, Kernel Methods.

## 1 Introduction

Generic machine learning algorithms do not cope with complex data such as images directly, a preprocessing step being usually required in order for them to perform various tasks. Among the solutions used to adapt image data to algorithm inputs, we discuss in this article a representation method as a structure which encodes explicitly image parts and spatial interactions in a graphical model.

Discriminative learning algorithms that are well suited for this kind of graphical models have been created (Kashima *et al.*, 2003) and optimized (Mahé *et al.*, 2004) in view of specific applications in computational chemistry and biology. An adaptation for coping with graphical models extracted from images is required

Emanuel Aldea · Isabelle Bloch
TELECOM ParisTech, TSI Department, CNRS UMR 5141 LTCI,
46 rue Barrault, Paris 75013, France
e-mail: {emanuel.aldea,isabelle.bloch}@telecom-paristech.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 77–94.
springerlink.com 

nevertheless, since the properties of the information encoded in the graphical structure is fundamentally different than in the context of biological or chemical structured data analysis.

In a larger context, image interpretation methods use primarily the visual features of low-level or high-level interest elements. However, spatial information concerning the relative positioning of these elements is equally beneficial, as it has been shown previously in segmentation and structure recognition. Therefore, an interest for the integration of spatial information in the learning framework has emerged recently. The fact that spatial information is often perceived and expressed in a manner which is close to natural language, along with the fact that the absence of a spatial interaction is also relevant, hint at the usefulness of fuzzy spatial information for image representation. Fuzzy representations actually permit to assess at the same time the imprecision degree of a relation (e.g., "close to" or "to the left of" ) and the gradual transition between the satisfiability and the non-satisfiability of a relation.

The objective of this article is to explore the limits of spatial information representation and its integration in the learning process within the context of image classifiers that make use of graph kernels. In the first part of our work, we present the advantages that labeled graphs provide for representing images, along with the general learning strategy employed by the corresponding SVM classifier. We continue with a short reminder on the use of spatial information in some related graph representations, and on the particularities of spatial information for image representation. The results show that spatial information complements the visual features of distinctive elements in images and that adjusting the kernel functions for the fuzzy spatial representations is beneficial in terms of performance.

## 2 Knowledge Representation by Labeled Graphs

In the domain of machine learning, generic supervised statistical algorithms accept input data in the form of numerical arrays or sequences and return a numerical value or indicate a specific category. Nowadays, input data are increasingly provided in complex configurations: as trees, graphs or other relational structures. These data arise very often from health and life sciences, but also from image processing, reasoning models for forecasting and decision making, etc. We witness accordingly the apparition of complex tasks that require the extraction of relations and structural dependencies out of input data. This situation suggests the emergence of learning methods adapted for these tasks and coping with large quantities of data.

In a structured data representation by graphical models, vertices may represent for instance atoms (Kashima *et al.*, 2003), simple chemical structures with specific properties (Mahé *et al.*, 2006), proteins (Borgwardt and Kriegel, 2007), segmentation regions in images (Aldea *et al.*, 2007a; Harchaoui and Bach, 2007), while edges encode specific interactions such as interdependence and scheduling, or spatial relations (adjacency, distance, relative localization, topology). In the context and particularly for image processing tasks, key sources of imprecision must be taken into account, concerning the objects and their imprecise delimitation and the relative

essence of the interaction information, often depicted using natural language. The graph structure and labeling integrate therefore the information that we possess concerning the basic elements that form the input objects, their features, the interactions among them but equally our confidence level for these types of information.

In the case of an image, one possible approach for the extraction of a graphical model is by building an adjacency graph upon the output of image segmentation. Graph vertices are associated with image regions and are labeled according to specific region features, related to size, color, texture. Usually, these numerical values are continuous, as opposed to discrete values that we may encounter in other applications (a chemical symbol, a protein identification reference or a nucleotide). The only structural information being used is the region adjacency, implicitly encoded by the graph edges. Extensions of this basic graphical model take into account more complex spatial and topological information using a richer labeling of the edges.

The next step consists in using a Support Vector Machine (SVM) (Vapnik, 1998) to classify the structures that were extracted. Given a positive definite function $K$, denoted as the *kernel function* of the classifier, a set of training objects $\mathcal{X}$ and a set of labels $\mathcal{Y}$ associated to the elements of $\mathcal{X}$, such that $y_i \in \{-1,+1\}$ for any $x_i \in \mathcal{X}$, the output of the classifier for a new object $x$ is:

$$y(x) = \operatorname{sgn}\left(\sum_{i=1}^{|\mathcal{X}|} \alpha_i y_i K(x_i,x)\right) \tag{1}$$

where $\alpha_i$ is the Lagrange multiplier in the optimization solution associated to the training object $x_i$.

An important observation is that the classifier only needs the value of the kernel function between pairs of examples, as a similarity estimation. An additional advantage of this approch is that it allows classifying elements issued from spaces which are not naturally endowed with inner products (such as graph, tree or string spaces), as long as we use a valid kernel function.

Furthermore, we describe the specific marginalized kernels that are being used in labeled graph analysis.

## 2.1 Marginalized Kernels

Given a generic class of objects $\mathcal{X}$, we assume that the constituents $x \in \mathcal{X}$ are generated according to a latent variable model which consists of the visible variable $x$ and of a hidden variable $\theta$, being considered jointly in a pair $z = [\theta, x]$. As we need a kernel $K(x,x')$ for the visible variables, we define first a *joint kernel* $K_z(z,z')$ for the mixed pair, which is used in a *marginalized kernel* (Tsuda *et al.*, 2002) defined as the expectation of the joint kernel over all the values of the hidden variable:

$$K(x,x') = \int_{\theta,\theta' \in \Theta} p(\theta|x)p(\theta'|x')K_z(z,z')d\theta d\theta' \tag{2}$$

where $\Theta$ refers to the domain of the hidden variable. In a discrete setting, the value of the marginalized kernel is estimated with:

$$K(x,x') = \sum_{\theta,\theta' \in \Theta} p(\theta|x)p(\theta'|x')K_z(z,z') \tag{3}$$

The difficulties to be considered when estimating marginalized kernels are the computational burden which is related to the dimension of $\Theta$, and the estimation from the data of the probabilistic model $p(\theta|x)$. Therefore, the choice of the model $p(\theta|x)$ should maximize the relevance for the specific data $x \in \mathcal{X}$ under the tractability constraint of $K(x,x')$. With respect to the properties of the function $K(x,x')$, as long as the joint kernel $K_z(z,z')$ is positive semidefinite, the kernel $K(x,x')$ is also positive semidefinite, since the class of positive semidefinite kernels is closed under addition and multiplication (Genton, 2001); the kernel may also be interpreted as the inner product of the two vectors $p(\theta|x)$ and $p(\theta'|x')$.

## 2.2 Building SVM Classifiers for Graphs

The graph similarity is assessed using a marginalized kernel function and is employed in a SVM classifier. This similarity, related to a specific feature $a$, between two graphs $G$ and $G'$ extracted from images is evaluated with a kernel that sums the similarities between all possible pairs of random walks in the two graphs (Kashima *et al.*, 2003), weighted by their probability of apparition.

In reference to other applications that used this type of marginalized graph kernel (Mahé *et al.*, 2004), the labeling space for vertex features is continuous and multidimensional. The similarity function for feature values has to be less discriminative than the Dirac delta function usually employed in the discrete case. Therefore, we use Gaussian kernels with variance $\sigma^2$ in order to evaluate the similarity $k_a^{rbf}(a_1,a_2)$ between two values $a_1$ and $a_2$ of the numerical feature $a$ :

$$k_a^{rbf}(a_1,a_2) = \exp\left(-\frac{\|a_1 - a_2\|^2}{2\sigma^2}\right) \tag{4}$$

Specific kernels have been shown to be adapted for other types of features. These kernels usually employ a well known distance between features that they kernelize; examples include a $\chi^2$-kernel between histograms :

$$k_{\chi^2}(h_1,h_2) = e^{-\alpha\chi^2(h_1,h_2)} \tag{5}$$

or a L1/L2 distance based kernel between multichannel mean color levels. In case of texture features, we use a distance metric defined in Arivazhagan *et al.* (2006) on descriptors (Bernardino and Santos Victor, 2006) based on the means and standard deviations of Gabor filter energy responses.

Given the two graphs $G$ and $G'$ to compare, Equation (4) is used to evaluate the similarity $k_{v,e}(h,h')$ between two random walks $h = \{x_1,\ldots,x_n\}$ in $G$ and $h' = \{x'_1,\ldots,x'_n\}$ in $G'$, by combining the similarity functions $k_v$ for a vertex feature $v$ and $k_e$ for an edge feature $e$ along $h$ and $h'$:

$$k_{v,e}(h,h') = k_v(v_{x_1}, v_{x'_1}) \prod_{i=2}^{n} k_e(e_{x_{i-1}x_i}, e_{x'_{i-1}x'_i}) k_v(v_{x_i}, v_{x'_i}) \tag{6}$$

This general equation may be simplified if we take into account only a region feature (as it happens with adjacency graphs) or if we take into account only an edge feature. In order to simplify the computation, we fix the value of the missing function to 1 (however, the element that is not considered for similarity computation must exist, otherwise a random walk containing the element could not exist).

At this point, we can underline the link between our specific kernel and the formal marginalized model depicted in Section 2.1. The input graph $G$ represents the visible variable, and the random walk $h$ represents the hidden variable. Therefore, the graph kernel between $G$ and $G'$ is computed by adding the similarities between all possible random walk kernels $h$ and $h'$, weighted by their probability of apparition:

$$K_{v,e}(G,G') = \sum_{h} \sum_{h'} k_{v,e}(h,h') p(h|G) p(h'|G') \tag{7}$$

This function is subsequently used with a support vector machine (SVM) in order to build an image classifier. The matrix $K_{v,e}$ defines the similarity between the graphs to compare.

## 3 Spatial Relations in the Context of Graph Learning

The spatial context has been taken into account in computational biology and chemistry when representing structured data. With few exceptions (Mahé *et al.*, 2006), spatial relations being used are binary and model rigorously the presence of an interaction between two elements of the structure. Even under this binary relation model, it has been shown that information brought by the absence of interactions may increase prediction performance, in relevant applications. For example, in protein-protein interaction (PPI) networks the absence of protein interactions is relevant for disease prediction. Therefore, a complement graph $\tilde{G}$ of the initial interaction graph $G$, which encodes the absence of interactions, has been proposed Borgwardt and Kriegel (2007). The resulting composite kernel:

$$K^*(G,G') = K(G,G') + K(\tilde{G},\tilde{G}') \tag{8}$$

leads to noteworthy improvements in classification accuracies on disease outcome prediction for cancer patients.

As to the extraction of spatial information for image representations, the situation is more complex. First of all, spatial interactions present an inherent semantic variability which goes well beyond the binary case mentioned above. Secondly, the

integration of fuzzy spatial information and region feature information turns out to be more complicated than the direct method depicted in Equation 8. However, this type of spatial information has been shown to enrich the description of images and to be useful for segmentation and structure recognition purposes. Below, we examine how we can use spatial information in learning and, more specifically, categorization.

We could just add fuzzy information on the existing edges of the image representation graph, but using strict adjacency for the underlying structure may pose robustness issues. Indeed, in cases where adjacency relies on a small number of pixels, the resulting graph may differ according to the segmentation method. Adding edges that represent more than the implicit strict adjacency relation does not only help with encoding structural information, but at the same time improves the robustness of the representation.

For our application, we use a topological spatial relation represented by an extended degree of adjacency, described below. Note that other relations could be added as well, using the same framework.

### *3.1 Distance between Regions*

The distance between two regions $R_1$ and $R_2$ is computed as the minimal Euclidean distance between two points $p_i \in R_1$ and $q_j \in R_2$:

$$d(R_1, R_2) = \min_{p_i \in R_1, q_j \in R_2} (d_{Euclidean}(p_i, q_j)) \tag{9}$$

Distance, as well as orientation, may not always be relevant, for instance the distance between two regions is the same if those two regions are adjacent by only one pixel, or if a region is surrounded by another region. Therefore we propose to consider a topological feature that measures the adjacency length between two regions.

### *3.2 Adjacency Measure Based on Fuzzy Satisfiability*

One way to estimate this measure is to compute the matching between the area “near" a reference region and another region. This measure is maximal in the case where the reference region is embedded into the second, and is minimal if the two regions are far away from each other.

Fuzzy representations are appropriate to model the intrinsic imprecision of several relations (such as “near") and the necessary flexibility for spatial reasoning (Bloch, 2005). We define the region of space in which a relation to a given object is satisfied as a fuzzy set. The membership degree of each point to this fuzzy set corresponds to the satisfiability degree of the relation at that point (Bloch, 2005). Note that this representation is in the image space and thus may be more easily merged with a representation of another region.

The spatial relation “near" is defined as a distance relation. A distance relation can be defined as a fuzzy interval $f$ of trapezoidal shape on $\mathbb{R}^+$. A fuzzy subset $\mu_d$

of the image space $\mathcal{S}$ can then be derived by combining $f$ with a distance map $d_R$ to the reference object $R$: $\forall x \in \mathcal{S}$, $\mu_d(x) = f(d_R(x))$, where $d_R(x) = \inf_{y \in R} d(x,y)$. In our experiments, the fuzzy interval $f$ is defined with the following fixed values: 0, 0, 10, 30 (Figure 1a). We exemplify using a butterfly image (Figure 1b) and the result of a segmentation exhibiting four distinct regions (Figure 1c). We illustrate the distance map to the region represented by the left wing (Figure 1d) and the fuzzy subset corresponding to the relation “near the left wing" (Figure 1e) which uses the distance map and the fuzzy interval defined above. Similarly, we compute fuzzy subsets for the right wing (Figure 1f) as well as for any other regions designated by the segmentation.

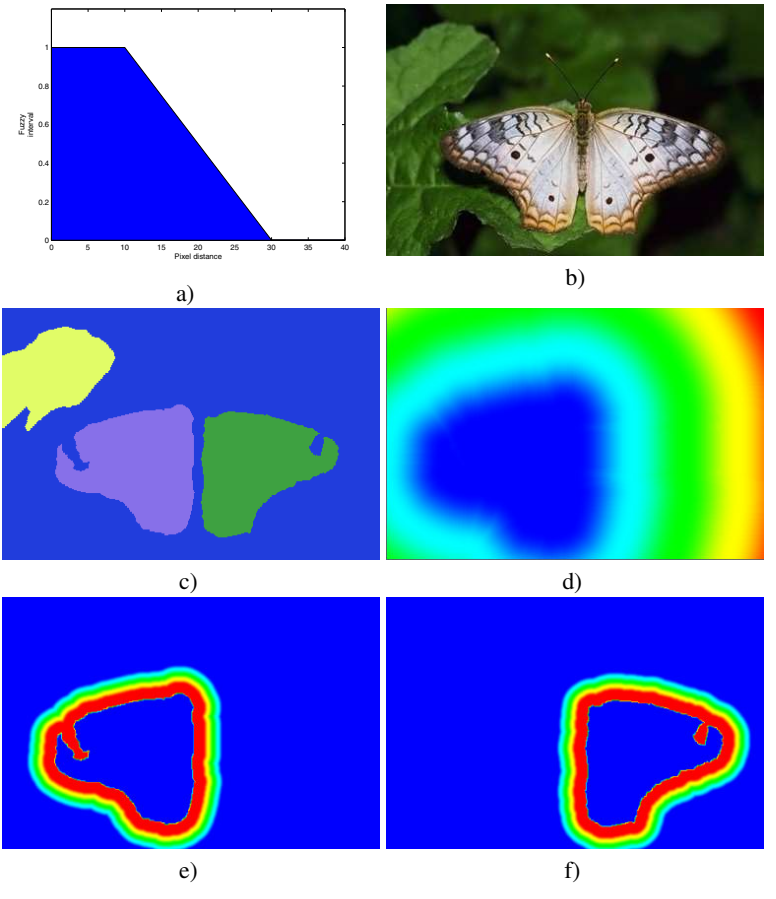



**Fig. 1** (a) Fuzzy interval for the distance relation. (b) Input image. (c) Segmentation result, four distinct regions. (d) Distance map to the region represented by the left butterfly wing. (e) Fuzzy subset corresponding to the relation “near the left wing" (red corresponds to highest values). (f) Fuzzy subset corresponding to the relation “near the right wing".

So far we have defined the area of the image in which the relation “near to" a reference object is defined. The next step consists in estimating the matching between this fuzzy representation and the other region. Among all possible fuzzy measures, we choose as a criterion a *M-measure of satisfiability* (Bouchon-Meunier *et al.*, 1996) defined as:

$$Sat(near(R_1), R_2) = \frac{\sum_{x \in \mathcal{S}} \min(\mu_{near(R_1)}(x), \mu_{R_2}(x))}{\sum_{x \in \mathcal{S}} \mu_{near(R_1)}(x)} \tag{10}$$

where $\mathcal{S}$ denotes the spatial domain. It measures the precision of the position of the object in the region where the relation is satisfied. It is maximal if the whole object is included in the kernel of $\mu_{near(R_1)}$. Note that the size of the region where the relation is satisfied is not restricted and could be the whole image space. If the object $R_2$ is crisp, this measure reduces to $\frac{\sum_{x \in R_2} \mu_{near(R_1)}(x)}{\sum_{x \in \mathcal{S}} \mu_{near(R_1)}(x)}$, i.e. the portion of $\mu_{near(R_1)}$ that is covered by the object $R_2$.

### *3.3 Adjacency Measure Based on Fuzzy Resemblance*

Beside satisfiability, we also choose a symmetric measure, the *M-measure of resemblance* (Bouchon-Meunier *et al.*, 1996) defined as :

$$Res(near(R_1), R_2) = \frac{\sum_{x \in \mathcal{S}} \min(\mu_{near(R_1)}(x), \mu_{R_2}(x))}{\sum_{x \in \mathcal{S}} \max(\mu_{near(R_1)}(x), \mu_{R_2}(x))}$$

This measure is maximal if the object and the relation are identical: this resemblance measure accounts for the positioning of the object and for the precision of the fuzzy set as well.

In Figure 1(e) and Figure 1(f) we have illustrated the fuzzy subsets corresponding to the two wings. With the fuzzy satisfiability measure defined above, we get a response of 0.100 for “right wing near the left wing" and 0.109 for “left wing near the right wing". It is equally worth noting that the two regions are disconnected with respect to the strict pixel adjacency.

In the remaining sections, we will denote by a spatial relation $R$ one of these measures of fuzzy adjacency, but we underline the fact that $R$ could be substituted for other functions that estimate the interaction between elements of the image structure. The choice of the spatial relation of adjacency for our illustration is immediate because fuzzy adjacency information extends naturally one of the most simple and pertinent relations between image regions, the strict adjacency. However, taking into account more complex spatial relations such as “parallel to” or “along”, along with their fuzzy measures of satisfiability (Vanegas *et al.*, 2009; Takemura *et al.*, 2005), is possible as long as these spatial relations are appropriate for the content of the input images.

## 4 Fuzzy Spatial Information and Discriminative Models

We can see that fuzzy spatial relations for images extend very conveniently the binary relations that are being used in other domains and achieve to merge at the same time information concerning the presence and the absence of an interaction. However, there are difficulties that arise when using this type of relations for discriminative learning.

In the context of image representation using spatial relations, related work has been done using binary relations (Deruyver *et al.*, 2009) supported by a specific ontology, using a count vector (Lebrun *et al.*, 2008) which estimates simple relative positioning, or using fuzzy spatial relations (Aldea *et al.*, 2007b). Each independent spatial relation builds in itself a novel data representation, therefore additional work may be necessary in order to make use of different spatial features simultaneously and efficiently. In this part of the article, we focus on the fact that a single fuzzy spatial relation creates *by itself* an infinite set of different representations. Rather than using multiple spatial relations for learning, we try to underline the specific challenges that a discriminative learning algorithm has when using a family of representations generated by the same fuzzy spatial relation.

Very often, there is a correlation between the value of a fuzzy spatial relation and the information gain: if the response is high, it means that the relation that the function has been designed for is much present. Consequently, low responses may be frequent (e.g. in the case of the “near" spatial relation) and may not bring the same amount of information. Discriminative learning, and discriminative learning for labeled graphs in particular, makes intensive use of similarity assessments between input objects. The similarity score between two graphs increases if these graphs exhibit many similar substructures. Complete graphs must be used if we compute a spatial relation value between all possible regions; therefore, if very low relation values are frequent (close to, or equal to 0), the graph kernel function will overincrease the graph similarity measure.

A straightforward solution to this situation is to threshold the spatial relation values, so that edges will exist only when the fuzzy adjacency estimation between two vertices is beyond a minimum value $\theta$. However, the strict adjacency graph does not necessarily belong to this set $\mathcal{G}$ of threshold graphs. In terms of *graph edit distance* (Riesen *et al.*, 2007), let us consider for an image the adjacency graph $G$ and an element $G_\theta \in \mathcal{G}$:

$$G_\theta = \{\mathcal{V}(G); (v_1; v_2) \in \mathcal{V}^2(G) | R(v_1, v_2) \geq \theta\} \tag{11}$$

where $R(v_1, v_2)$ is the generic spatial relation function between regions (vertices) $v_1$ and $v_2$.

Obviously, the vertex sets of $G$ and $G_\theta$ are identical, as representations of the same image segmentation. We denote by $\mathcal{V}(G)$ the vertex set of graph $G$, and by $\mathcal{E}(G)$ its edge set. Under these circumstances, the graph edit distance $d_{g.e.}(G, G_\theta)$ between $G$ and $G_\theta$ is generated by strictly adjacent regions that are not close enough

in terms of $\theta$-fuzzy adjacency and non strictly adjacent regions that are close in terms of $\theta$-fuzzy adjacency:

$$\begin{aligned} d_{g.e.}(G, G_\theta) &= \text{card}\{(v_1; v_2) \in \mathcal{V}^2(G) | (v_1, v_2) \in \mathcal{E}(G) \wedge R(v_1, v_2) < \theta\} \\ &+ \text{card}\{(v_1; v_2) \in \mathcal{V}^2(G) | (v_1, v_2) \notin \mathcal{E}(G) \wedge R(v_1, v_2) \geq \theta\} \quad (12) \end{aligned}$$

In practice this means that if we extend the spatial information beyond the intuitive strict adjacency between regions, we get for each spatial relation $R$ a graph set $\mathcal{G}$ instead of a single graph representation of the image, and learning using spatial relations should be adapted to this situation. More precisely, we should know which of these graphs is more appropriate for learning.

The graph or graphs $G_* \in \mathcal{G}$ that minimize $d_{g.e.}$ are the closest (structurally) to the adjacency graph G. These are the projections of $G$ in the set $\mathcal{G}$, and are ideally robust generalizations of $G$ with respect to the spatial relation $R$. However, $G_*$ and $G$ might still exhibit various differences (the edge sets $\mathcal{E}(G)$ and $\mathcal{E}(G_*)$ are not identical), therefore the structural information within might still be partially disjunct.

The element that bridges the informational gap between $G$ and $\mathcal{G}$ is the complete graph $G_f$, which includes (structurally) any element $G_\theta \in \mathcal{G}$, as well as the strict adjacency graph $G$. Ideally, the learning algorithm should exhibit the best performance with $G_f$, but then it should be able to cope well with the noise generated by a lot of similar low-information edge labels.

## 5 Experiments and Results

### 5.1 Data Set

The Internet Brain Segmentation Repository (IBSR) data set[1] contains real clinical data and is a widely used 3D healthy brain magnetic resonance image (MRI) database. It provides eighteen manually-guided expert brain segmentations, each of them being available for three different views, along reference planes: axial, sagittal and coronal. Each element of IBSR is a set of slices that cover the whole brain.

The main purpose of the data set is to provide a tool for evaluating the performance of segmentation algorithms. However, the fact that it is freely available and that it offers high quality segmentations as input data makes it also useful for our experiments.

### 5.2 Experimental Setup

Image categorization between images belonging to different views in the data set (sagittal, coronal, axial) is performed with a 100% success rate for many of

[1] The MR brain data sets and their manual segmentations were provided by the Center for Morphometric Analysis at Massachusetts General Hospital and are available at http://www.cma.mgh.harvard.edu/ibsr/

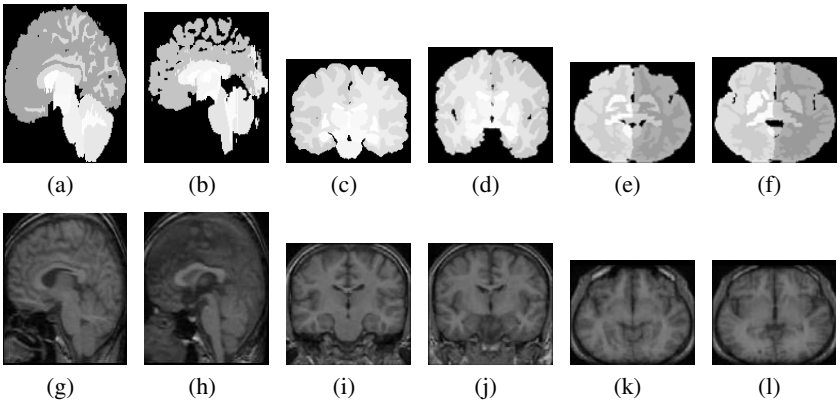



**Fig. 2** Samples from IBSR data set. 2(a), 2(b) Two slices of the sagittal view of the same 3D MRI volume representing the two categories. 2(c), 2(d) Coronal view. 2(e), 2(f) Axial view. The original images are presented below their corresponding manual segmentations.

**Table 1** Identification of the slices composing the database in each view of the 3D volume, for the three possible views: sagittal (S), coronal (C) and axial (A)

| View | Slices | Slices cat. 1 | Slices cat. 2 |
|---|---|---|---|
| S | 256 | 121, 122, 123 | 126, 127, 128 |
| C | 128 | 58, 59, 60 | 64, 65, 66 |
| A | 256 | 121, 122, 123 | 126, 127, 128 |

the features that we take into account; as a result, we build a more challenging categorization problem between images belonging to the same view; a secondary benefit of this approach is that by choosing certain slices we can control the difficulty of the task. Since the brain is made up of consecutive slices in any of the three views and the brain structure varies progressively, we want to create one category using three consecutive slices which are at the same level over all the eighteen 3D brain segmentations. A second category is being built using three consecutive slices which are positioned at a certain distance from the first block; as the distance between the two blocks of slices decreases, the difficulty of the categorization task increases. We found out that choosing a distance of only two or three slices between the training blocks, along with category intra-variability, would account for a difficult categorization task. Table 1 references the total number of slices in each 3D brain view and the indices of slices being used for defining each category; Figure 2 presents typical category elements for all views. Each brain view will provide three images for each category, thus creating a category definition of 54 images.

Concerning the graph construction and labeling, nodes are represented by manually segmented regions while edges account for spatial relations between regions.

For vertex labeling, we use normalized region visual features: the mean gray level (which is normalized according to the lightest and darkest regions in the image), the relative region area (normalized according to the total image area) and the normalized region compacity, defined as the normalized ratio between its surface and its squared perimeter. For this work specifically, we will experiment with the coronal view and with the mean gray level as region feature. Spatial relations based on adjacency measures being considered between image regions build up the edge labeling, respectively.

We perform n-fold cross validation on the training set ($n = 10$), and we repeat the classification task $m$ times ($m = 10$); the performance given below is the mean value of these $m$ executions.

### 5.3 Categorization with Strict Adjacency Structures

Given a region feature and a spatial relation within the strict adjacency graph, we use RBF kernels (Equation 4) with thresholds that are adapted to the range $[0,1]$ of these normalized features (see Figure 3), and we set up a grid search in the $\sigma$ parameter space for each of the two kernel functions. For each element of the grid, we try multiple values for the regularization parameter $C$ of the SVM, $C \in \{10^{-2}, 10^{-1}, \ldots, 10^{6}\}$. Figure 3 presents the best classification performance for each pair $(\sigma_{vertex}, \sigma_{edge})$, for the values and features specified on the axis.

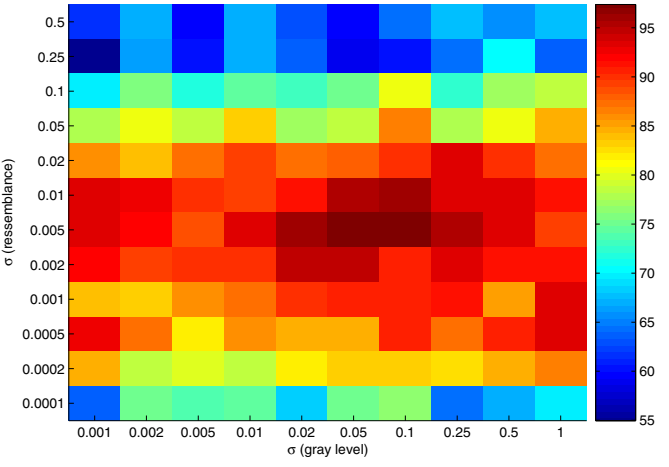
### 5.4 Categorization with Fuzzy Adjacency Structures

Next, we analyze the impact of adding structural information which is not necessarily tied to the strict adjacency between image regions. For a given segmentation and for a certain spatial relation $R$, the complete graph encodes all the possible relations between vertices, as edge labels.

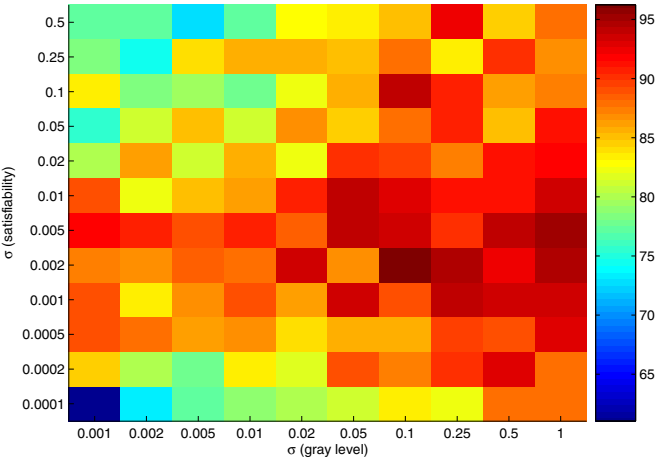
The histograms in Figure 4 present the satisfiability and resemblance values encoded within all the complete graphs in the dataset. From these figures, we notice that the first type of measure takes the maximum value more often, while the frequency of low values is very significant. For the second measure, maximum values are quite low even for adjacent regions (the maximum value in all the dataset being 0.39), and low values are very frequent, too.

In order to estimate the impact of different spatial relation thresholds $\theta$ on the structure of elements in the threshold graph set $\mathcal{G}$, we compute the number of differences between the set of strict adjacency edges and $\theta$-thresholded edges with respect to the relation $R$, using Equation 12. The difference profiles for the satisfiability and resemblence relations are presented in Figure 5(a) and Figure 5(b). Given any of the adjacency graphs $G$ in the dataset, the threshold $\theta$ that would minimize the structural difference between $G$ and $G_\theta \in \mathcal{G}$ is given by the value corresponding to the minimum in the difference profile, $\theta_*$.

For our dataset, the optimal threshold for the satisfiability measure is $\theta_*^{sat} = 0.911$. This high value proves the fact that most of the times, strictly adjacent

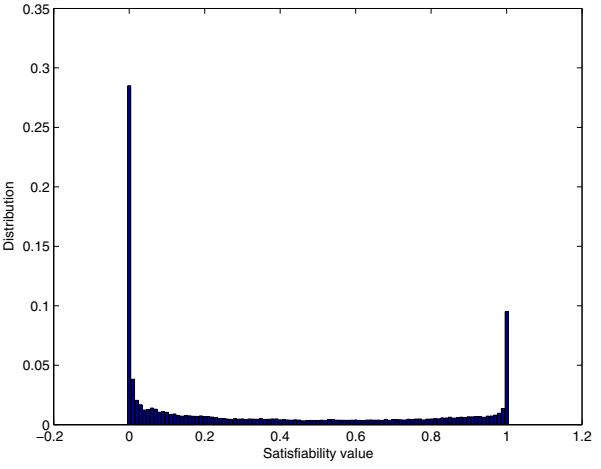



(a) Using the gray level region attribute and the resemblance measure, the best performance, 97.72%, is attained for $\sigma_{vertex} = 0.1$ and $\sigma_{edge} = 0.005$.

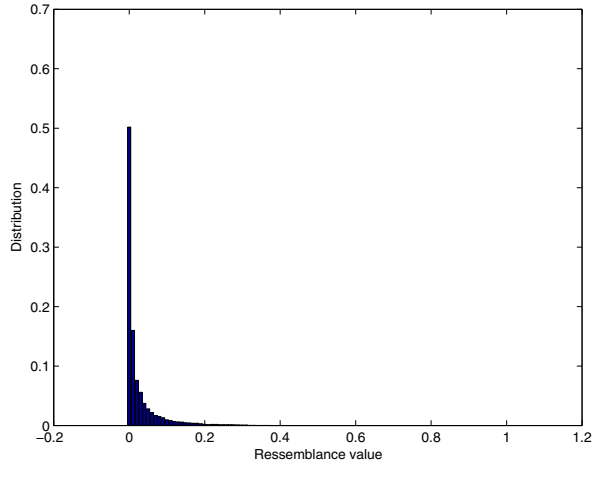



(b) Using the gray level region attribute and the satisfiability measure, the best performance, 96.51%, is attained for $\sigma_{vertex} = 0.1$ and $\sigma_{edge} = 0.002$.

**Fig. 3** Categorization performance for the gray level region attribute and two different measures of fuzzy adjacency, using grid search in the space of kernel parameters. At this point, we model input data using strict adjacency graphs.

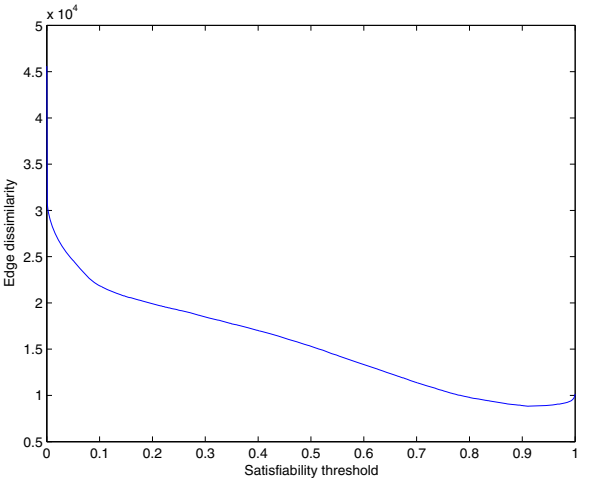



(a) Histogram of the satisfiability measure. Null values are the most frequent ones, but maximal values are frequent too.

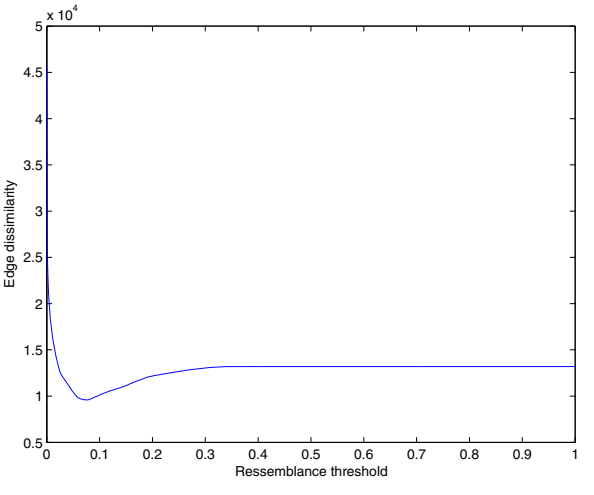



(b) Histogram of the resemblance measure. Null values are equally the most frequent ones, but this measure penalizes very fast the absence of a strong adjacency and the maximum value in all the dataset is 0.39.

**Fig. 4** Distribution of the two measures of fuzzy adjacency for all the edges in the set of complete graphs representing the dataset




(a) Satisfiability edge dissimilarity count




(b) Resemblance edge dissimilarity count

**Fig. 5** For a given measure (satisfiability or resemblance) threshold $\theta$, we show the number of different edges between the set of strict adjacency graphs of the dataset and the set of $\theta$-threshold graphs associated to the strict adjacency graphs. The minimal value accounts for the highest structural similarity between the strict adjacency graphs and the $\theta$-threshold graphs.

**Table 2** Categorization performance for (gray level - spatial relation) image information. The parameters for the kernel functions are the optimal values found using the grid search. In the third column, the strict adjacency graph is used, but no spatial relation labels are added to the graph. In the fourth column, we use fuzzy adjacency labeling on the strict adjacency graph. Afterwards, we use different $\theta$-threshold fuzzy adjacency graphs.

| Region feature | Spatial relation | Strict adj. No relation | Strict adj. Fuzzy labeling | Fuzzy graphs |
|---|---|---|---|---|
| Gray level | Resemblance | 58.18% | 97.72% | $\theta$=0.00 (82.90%)<br>$\theta$=0.01 (82.92%)<br>$\theta$=0.02 (86.43%)<br>$\theta$=0.05 (84.38%)<br>$\theta_*$=0.075 (74.43%)<br>$\theta$=0.10 (76.23%)<br>$\theta$=0.20 (69.90%)<br>$\theta$=1.00 (61.73%) |
| Gray level | Satisfiability | 57.04% | 96.51% | $\theta$=0.00 (79.07%)<br>$\theta$=0.1 (70.83%)<br>$\theta$=0.25 (68.18%)<br>$\theta$=0.5 (76.75%)<br>$\theta$=0.75 (77.51%)<br>$\theta_*$=0.911 (77.29%)<br>$\theta$=1.00 (62.85%) |

regions account for satisfiability values that go beyond the threshold, as it is perceivable from the high proportion of maximum values. The second measure has a different behavior; it penalizes very fast the absence of a strong adjacency. In this case, the $R$ values associated to the strict adjacency relation are scattered on a larger interval, thus the optimal threshold is situated further from the maximum value: $\theta_*^{ress} = 0.075$.

In Table 2 we compare the categorization performances for different settings involving spatial relations. As a reference, we use the best classifier detected for a certain region feature-spatial relation pair, using grid search. This classifier relies on the strict adjacency graph extracted from the image, but the edges are labeled using the spatial relation value between the corresponding vertices. The interest of incorporating spatial relation information to the labeling is proven by the weak performance of the classifier on the adjacency graph which uses only the region feature information (the edge kernel $k_e$ being fixed set as $k_e = 1$, cf. Section 2.2).

Next, we pass to the threshold graphs $G_\theta$ in the set $\mathcal{G}$. In our setting, the spatial relations $R$ are represented using values in $[0,1]$, therefore the threshold $\theta$ is also a number in $[0,1]$. We estimate the categorization performance along the set $\mathcal{G}$; reference elements are the complete graph $G_f = G_0$, $G_{\theta_*}$ the projection of $G$ in $\mathcal{G}$, and $G_1$.

Results in Table 2 show that once we pass to a structure which is based entirely on thresholded fuzzy spatial relations, we do not improve the best performance witnessed on the strict adjacency graph structure. Within the set $\mathcal{G}$, the projection $G_{\theta_*}$ of $G$ performs well and the classifier performance may be improved by lowering slightly the threshold below the value of $\theta_*$, which accounts for adding edges with spatial information. However, $\theta$ values that are far from $\theta_*$, including the value $\theta = 0$ that corresponds to the complete graph, account for a poorer performance. This shows that in the presence of a richer information, the performance does not necessarily improve. The explanation is that the high frequency of low values for the edge labels leads to artificially high similarity estimations between graphs and masks the similarity of meaningful high label values. While the spatial information is definitely helpful in image interpretation, its generic integration into graphical models remains a difficult task and kernel functions for SVMs that cope with spatial information should be adapted specifically to different types of spatial relations.

## 6 Conclusion

In this article, we studied the benefits offered by image representations using labeled graphical models, as well as by employing fuzzy descriptors for spatial information. Graphical models allow for a flexible integration between intrinsic visual features of image parts and the spatial interactions taking place. We showed that fuzzy information is highly beneficial for the learning process when we use it to enrich the labeling of strict adjacency graphical structures, but that loose spatial interactions may screen more relevant spatial information and that generic kernel functions are not well adapted to take into account the entirety of spatial relations within images. Future work will try to adapt the graph similarity estimation to the specificity of spatial relations in order to benefit from information concerning the presence and the absence of interactions.

## References

Aldea, E., Atif, J., Bloch, I.: Image Classification using Marginalized Kernels for Graphs. In: 6th IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, GbR 2007, Alicante, Spain, pp. 103–113 (2007a)

Aldea, E., Fouquier, G., Atif, J., Bloch, I.: Kernel Fusion for Image Classification Using Fuzzy Structural Information. ISVC (2), 307–317 (2007b)

Arivazhagan, S., Ganesan, L., Priyal, S.P.: Texture classification using Gabor wavelets based rotation invariant features. Pattern Recogn. Lett. 27(16), 1976–1982 (2006), http://dx.doi.org/10.1016/j.patrec.2006.05.008

Bernardino, A., Santos Victor, J.: Fast IIR Isotropic 2-D Complex Gabor Filters With Boundary Initialization. IP 15(11), 3338–3348 (2006)

Bloch, I.: Fuzzy Spatial Relationships for Image Processing and Interpretation: A Review. Image and Vision Computing 23(2), 89–110 (2005)

Borgwardt, K.M., Kriegel, H.-P.: Graph Kernels For Disease Outcome Prediction From Protein-Protein Interaction Networks. In: Pacific Symposium on Biocomputing, pp. 4–15 (2007)

Bouchon-Meunier, B., Rifqi, M., Bothorel, S.: Towards general measures of comparison of objects. Fuzzy sets and Systems 84(2), 143–153 (1996)

Deruyver, A., Hodé, Y., Brun, L.: Image interpretation with a conceptual graph: Labeling over-segmented images and detection of unexpected objects. Artif. Intell. 173(14), 1245–1265 (2009)

Genton, M.G.: Classes of Kernels for Machine Learning: A Statistics Perspective. Journal of Machine Learning Research 2, 299–312 (2001)

Harchaoui, Z., Bach, F.: Image Classification with Segmentation Graph Kernels. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2007, pp. 1–8 (2007), http://dx.doi.org/10.1109/CVPR.2007.383049

Kashima, H.: Tsuda, K. and Inokuchi, A., Marginalized Kernels Between Labeled Graphs. In: 20st Int. Conf. on Machine Learning, pp. 321–328 (2003)

Lebrun, J., Philipp-Foliguet, S., Gosselin, P.H.: Image retrieval with graph kernel on regions. In: ICPR, pp. 1–4 (2008)

Mahé, P., Ralaivola, L., Stoven, V., Vert, J.-P.: The Pharmacophore Kernel for Virtual Screening with Support Vector Machines. J. Chem. Inf. Model. 46(5), 2003–2014 (2006), http://dx.doi.org/10.1021/ci060138m

Mahé, P., Ueda, N., Akutsu, T., Perret, J.-L., Vert, J.-P.: Extensions of marginalized graph kernels. In: ICML 2004: 21st Int. Conf. on Machine Learning (2004)

Riesen, K., Neuhaus, M., Bunke, H.: Bipartite Graph Matching for Computing the Edit Distance of Graphs. In: Escolano, F., Vento, M. (eds.) GbRPR 2007. LNCS, vol. 4538, pp. 1–12. Springer, Heidelberg (2007)

Takemura, C.M., Cesar, R.M., Bloch, I.: Fuzzy Modeling and Evaluation of the Spatial Relation "Along". In: Sanfeliu, A., Cortés, M.L. (eds.) CIARP 2005. LNCS, vol. 3773, pp. 837–848. Springer, Heidelberg (2005)

Tsuda, K., Kin, T., Asai, K.: Marginalized kernels for biological sequences. Bioinformatics 18(suppl. 1), 268–275 (2002)

Vanegas, C., Bloch, I., Maître, H., Inglada, J.: Approximate Parallelism Between Fuzzy Objects: Some Definitions. In: Di Gesù, V., Pal, S.K., Petrosino, A. (eds.) Fuzzy Logic and Applications. LNCS (LNAI), vol. 5571, pp. 12–19. Springer, Heidelberg (2009)

Vapnik, V.: Statistical Learning Theory. Wiley Interscience, Hoboken (1998)

# Part II
# Unsupervised Learning

# Multigranular Manipulations for OLAP Querying

Gilles Hubert and Olivier Teste

**Abstract.** Decisional systems are based on multidimensional databases improving OLAP analyses. This chapter describes a new OLAP operator named "BLEND" that performs multigranular analyses. This operation transforms multidimensional structures when querying in order to analyze measures according to several granularity levels like one parameter. We study valid uses of this operation in the context of strict hierarchies. Experiments within a R-OLAP implementation show the light cost of the operator.

**Keywords:** Decision Support Systems, Multidimensional Databases, OLAP Querying, Multigranular Analysis.

## 1 Introduction

Decision support systems are experiencing a great boost in development because of their capacity to effectively support analyses on available data in the organizations. These decision systems are elaborated starting from the operational system of an organization: the data identified as relevant for decision makers are extracted, transformed, then loaded (Vassiliadis *et al.*, 2002) in a centralized storage space called data warehouse. In order to improve querying and analysis of these stored data, specific techniques of data organization were developed (Kimball, 1996) based on multidimensional databases (MDB). This type of modeling considers the data to be analyzed as points in a space with several dimensions, thus forming a datacube of data (Gray *et al.*, 1996). Decision makers who use these systems visualize an excerpt of the datacubes, generally a "slice" with only two dimensions of a

Gilles Hubert · Olivier Teste
Université de Toulouse, IRIT (UMR 5505), équipe SIG, 118 Route de Narbonne,
31062 Toulouse cedex 9, France
e-mail: {Gilles.Hubert,Olivier.Teste}@irit.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 97–112.
springerlink.com 

cube. From this structure, called multidimensional table (Gyssens and Lakshmanan, 1997), the decision maker can interact with operations. The best known operations are drilling operations which consist in modifying the graduation of an analysis axis (levels of granularity) and rotations which consist in changing the cube slice. One speaks about online analysis or about OLAP ("On-line Analytical Processing") (Ravat *et al.*, 2008).

This environment offers an adapted framework to decision makers' analyses; however the imposed structure can prove to be imperfect or become obsolete. Let us consider sale amounts analyzed according to French customers and American customers. Within this framework, a decision maker may want to use the graduation according to the country for the French customers while wishing to use a different graduation simultaneously, for example the states for the customers of the USA. Indeed, for some analyses, it is necessary to compare a country like France with different geographic entities like states to compare information equivalent in size, population size, etc. The objective of this paper is to propose a solution allowing these manipulations described as *multigranular*.

## 2 Related Work

There exist mainly two approaches for MDB modeling: an approach based on the datacube metaphor whereby an MDB is represented by cubes; and an approach known as multidimensional modeling whereby a MDB is represented by a star or constellation schema (Kimball, 1996). Several field surveys (Chaudhuri and Dayal, 1997; Vassiliadis and Sellis, 1999) and comparative studies (Abelló *et al.*, 2006; Ravat *et al.*, 2008) are available.

One of the first works extends the aggregation operation in the OLAP context (Gray *et al.*, 1996). Since then, a great number of operations were defined; however due to lack of consensus on a reference model, the proposals for OLAP operations still were neither clearly identified nor defined within an algebra following the example of the relational approach. A comparative study of the many existing proposals is available in Romero and Abelló (2007).

To our knowledge no proposal can answer our problems. The closest solutions propose mechanisms aiming at personalizing an MDB by transforming its values and its structures. In Espil and Vaisman (2001) the rule-based language IRAH is introduced to allow decision makers to change value groupings between two graduations. However, this approach does not make it possible to transform the hierarchical structures of the graduations initially defined in the MDB. This approach does not allow multigranular analyzes by combining existing graduations; e.g., it allows a cisgenic organism denoted (C1, CIS) to become a transgenic organism denoted (C1, TRA). Our approach aims at generating a new graduation both composed of organisms (C1) and categories such as transgenic (TRA). More recently, Favre *et al.* (2007) introduced a mechanism based on "If-Then" rules in order to integrate users' knowledge to change the MDB schema. This mechanism allows users to add new

graduations individually. Although these solutions allow a certain adaptation of a MDB it raises two problems: firstly, the transformation process is tricky and tedious because based on the definitions of rules expressed by the decision maker, and secondly, coherence and confidence with the stored decisional data are not guaranteed any more. Introducing direct means to access values in update mode renders inoperative the usual processes of data cleaning and consolidation.

Other works in MDB evolution context proposed operations to transform the hierarchies modeled initially (Blaschka *et al.*, 1999; Hurtado *et al.*, 1999; Eder *et al.*, 2003). In Blaschka *et al.* (1999), an operation to insert a new parameter is presented. The operation "Reports Levels", defined in Hurtado *et al.* (1999), makes it possible to transform the hierarchical organization of parameters. Other transformation operations such as "Split" related to parameter values are described in Eder *et al.* (2003). This work offers a framework allowing the evolution of hierarchies, but does not really correspond to multigranular transformations. These operations can be diverted to transform an MDB. However, our goal is different as it aims to help reorganize the values between two graduations, and this, during the analysis process, without impacting the data physically stored in the MDB.

## 3 Contribution and Organization

The main contribution of this article is the proposal of a new manipulation in MDB facilitating multigranular analyses. A multigranular analysis combines the same analyzed measurements according to data resulting from several parameters: for example, we make possible the analysis of agricultural surfaces according to geographical values of different levels such as USA and European surfaces.

We extend the OLAP algebra, defined in our laboratory (Ravat *et al.*, 2008), by the multigranular analysis operator "BLEND". We carry out a study of the various possible uses of the operator in the context of strict hierarchies (Malinowski and Zimányi, 2006). We propose an operation that transforms the current hierarchy and the contours limits of the operator. Lastly, we experiment the operation in the context of a R-OLAP implementation.

An advantage of the suggested solution, is to make possible this type of analysis during analysis runtime whereas it would require complete data reorganization as well as associated ETL processes in a traditional context. The construction of an MDB is a tedious task and difficult to reproduce according to each analytical need. Applying these transformations during analysis runtime without impacting the real data organization facilitates sharing the MDB.

Section 4 presents the MDB model, i.e. the conceptual representation we adopt. We define a new operator called "BLEND" in Sect. 5. We show the various possible cases of multigranular manipulation authorized in the context of strict hierarchies (Malinowski and Zimányi, 2006). Section 6 describes the implementation of the operator in a R-OLAP context.

## 4 Multidimensional Modeling and OLAP Manipulations

This section describes our multidimensional framework based on a conceptual view displaying MDB structures as a graphical conceptual view. Our model allows users to disregard technical and storing constraints and sticks closer to decision makers' view (Golfarelli *et al.*, 2002). It allows a clear distinction between structural elements and values and offers a workable visualization for decision makers (Gyssens and Lakshmanan, 1997).

A constellation regroups several analysis subjects (facts), which are studied according to several analysis axes (dimensions) possibly shared between facts. It extends star schemas (Kimball, 1996) commonly used in the multidimensional context.

**Definition 1.** A *constellation C* is defined as $(NC, FC, DC, StarC)$ where:

- $NC$ is a constellation name,
- $FC = \{F_1, \ldots, F_m\}$ is a set of facts,
- $DC = \{D_1, \ldots, D_n\}$ is a set of dimensions,
- $StarC : FC \rightarrow 2^{DC}$ associates each fact to its linked dimensions.

A dimension models an analysis axis; i.e. it reflects information according to which analysis subjects will be analyzed. A dimension is composed of attributes (dimension properties).

**Definition 2.** A *dimension*, noted $D \in DC$, is defined as $(N^D, A^D, H^D)$ where:

- $N^D$ is a dimension name,
- $A^D = \{a_1^D, \ldots, a_u^D\} \cup \{id^D, All\}$ is a set of attributes,
- $H^D = \{H_1^D, \ldots, H_v^D\}$ is a set of hierarchies.

Dimension attributes (also called parameters or levels) are organized according to one or more hierarchies. Hierarchies represent a particular vision (perspective) of a dimension. Each attribute represents one data granularity according to which measures can be analyzed; for example, along the store dimension, a hierarchy could group individual stores into cities and cities into countries. Weak attributes (attributive properties) complete the parameter semantics, e.g. the name of an individual store.

**Definition 3.** A *hierarchy* of a dimension $D$, noted $H \in HD$, is defined as $(N^H, Param^H, Weak^H)$ where:

- $N^H$ is a hierarchy name,
- $Param^H = \langle id^D, p_1^H, \ldots, p_v^H, All \rangle$ is an ordered set of attributes, called *parameters*, which represent useful graduations along the dimension, $\forall k, p_k^H \in A^D$,
- $Weak^H : Param^H \rightarrow 2^{A^D - Param^H}$ is a function possibly associating each parameter to one or several *weak attributes*.

All hierarchies of a dimension start with a same parameter, noted $id^D$ called *root parameter* and end with a same parameter, noted *All* called *extremity parameter*.

A fact reflects information that has to be analyzed according to dimensions. This analyzed information is modeled through one or several indicators, called measures; for example, a fact data may be sale amounts occurring in shops every day. The notation $D \in StarC(F)$ represents that the dimension $D$ is linked to the fact $F$.

**Definition 4.** A *fact*, noted $F \in FC$, is defined as $(N^F, M^F)$ where:

- $N^F$ is a name of fact,
- $M^F = \{f_1(m_1^F), \ldots, f_w(m_w^F)\}$ is a set of *measures* associated with an aggregate function.

Constellation schemas depict MDB structures whereas user analyses are based on tabular representations (Gyssens and Lakshmanan, 1997) where structures and data are displayed. The visualization structure that we define is a multidimensional table (MT), which displays data from one fact and two of its linked dimensions.

**Definition 5.** A *multidimensional table* $T$ is defined as $(S, L, C, R)$ where:

- $S = (F^S, M^S)$ represents the analyzed subject through a fact $F^S \in FC$ and a set of projected measures $M^S = \{f_1(m_1), \ldots, f_x(m_x)\}$ where $\forall i \in [1..x], m_i \in M^F$,
- $L = (DL, HL, PL)$ represents the horizontal analysis axis where $PL = \langle All, p_{max}^{HL}, \ldots, p_{min}^{HL} \rangle$, $HL \in H^{DL}$ and $DL \in StarC(F^S)$, $HL$ is the current hierarchy of $DL$,
- $C = (DC, HC, PC)$ represents the vertical analysis axis where $PC = \langle All, p_{max}^{HC}, \ldots, p_{min}^{HC} \rangle, HC \in H^{DC}$ and $DC \in StarC(F^S)$, $HC$ is the current hierarchy of $DC$,
- $R = pred_1 \wedge \ldots \wedge pred_t$ is a normalized conjunction of predicates (restrictions of dimension data and fact data).

*Example 1.* We consider an MDB to analyze the surface of parcels of land with genetically modified (GM) organisms around the world. The constellation is composed of one fact and three dimensions (see Fig. 1). The graphical notations we adopt are inspired from the notations of (Golfarelli *et al.*, 1998). According to formal definitions the graphical constellation is defined as follows:

('C1', $\{F^{DISTRIBUTION}\}, \{D^{ORGANISM}, D^{DATE}, D^{GEOGRAPHY}\}, \{(F^{DISTRIBUTION}, \{D^{ORGANISM}, D^{DATE}, D^{GEOGRAPHY}\})\}$)

The fact denoted $F^{DISTRIBUTION}$ is defined as follows:

('DISTRIBUTION', {SURFACE})

The dimension denoted $D^{GEOGRAPHY}$ is defined as follows:

('GEOGRAPHY', {PARCEL, STATE, REGION, COUNTRY, CONTINENT, DENSITY}, {HGEO, HST}) where:

- HGEO = ('GEO', ⟨PARCEL, REGION, COUNTRY, CONTINENT⟩, {(COUNTRY, {DENSITY})})
- HST = ('ST', ⟨PARCEL, STATE, COUNTRY, CONTINENT⟩, {(COUNTRY, {DENSITY})})

A decision maker displays data into multidimensional tables; $T_1$ displays surfaces according to continents and organism types. This MT is transformed into $T_2$ using a combination of drill-down and roll-up operations for displaying surfaces according to country and organism varieties. $T_2$ allows the decision maker to compare parcels with GM organisms (GTS-Soya, Corn BT176 and Mon 810) as well as parcels without GM organisms.

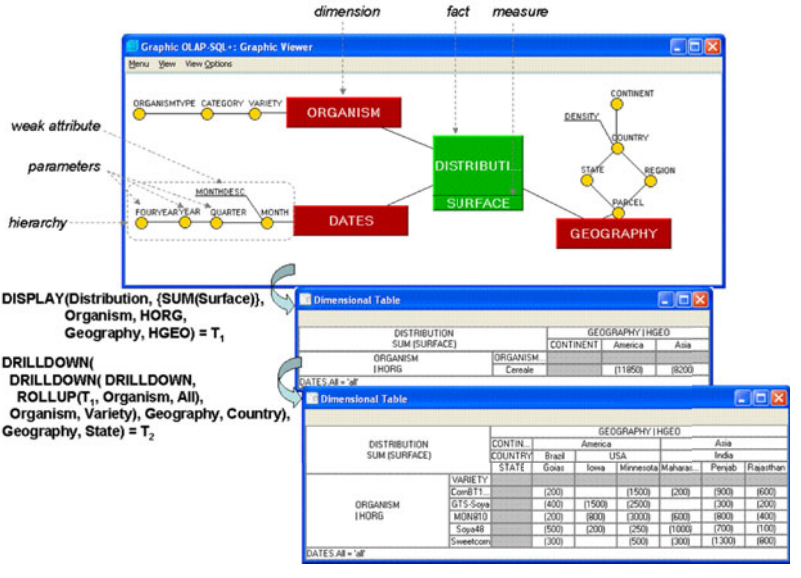



**Fig. 1** Star schema example (constellation composed of only one fact) and two multidimensional tables resulting from OLAP operations

## 5 Operator "BLEND"

In order to answer our problems of multigranular analyses, we define an operation to transform dimension parameters. This operation named "BLEND" is applied to an MT in order to modify the headings of the lines or the columns.

### *5.1 Algebraic Operator*

**Definition 6.** The operation of multigranular transformation of a MT is defined by: $BLEND(T_{SRC}, D, P_{sup}(s_{sup}), P_{inf}(s_{inf}), pred) = T_{RES}$

- $T_{SRC} = (S_{SRC}, L_{SRC}, C_{SRC}, R_{SRC})$ is the source MT to transform
- $D \in \{DL_{SRC}, DC_{SRC}\}$ is one of the dimensions of the MT $T_{SRC}$
- $P_{sup}$ and $P_{inf}$ are consecutively displayed parameters of the dimension D such that $P_{sup}$ is the parameter hierarchically higher than $P_{inf}$,

- $s_{sup} \in \{+,-\}$ and $s_{inf} \in \{+,-\}$ are tags indicating the conservation $(+)$ or not $(-)$ of the parameter associated in $T_{RES}$; the use of the tags and their various combinations are studied in an exhaustive way in the following section 5.2,
- *pred* is a selection predicate that determines the values resulting from the parameters $P_{sup}$ and $P_{inf}$ to build the definition field of the new parameter noted $P_{sup}\_P_{inf} \in T_{RES}$,
- $T_{RES}$ is the resulting MT.

The predicate *pred* is used to compute the sets $E_{sup}$ and $E_{inf}$, which gather the values resulting from the parameters $P_{sup}$ and $P_{inf}$ taking part in the construction of the new parameter field:

- $E_{sup}$ contains the values of $P_{sup}$ selected by *pred*,
- $E_{inf}$ contains the values of $P_{inf}$ selected by $\neg$ *pred*.

**Constraint 1.** *The predicate noted* pred *in the definition of operator "BLEND" is valid if and only if* $E_{sup} \cap ancestor(E_{inf}) = \varnothing$ *with:*

- $ancestor(E_{inf})$ *indicates the values of* $dom(P_{sup})$ *related to* $E_{inf}$,
- $dom(P_{sup})$ *indicates the field definition of* $P_{sup}$.

*For simplicity we will say that* pred *must define two sets of values "disjoined" in comparison with the hierarchical organization.*

**Constraint 2.** *The composition of "BLEND" operators is not commutative. The user must build his manipulations taking into account the order of the parameters* $P_{sup}$ *and* $P_{inf}$*, but also the order of the combinations of the multigranular transformations.*

## 5.2 Transformation Cases

The operator "BLEND" modifies the existing hierarchy by substituting a new parameter to one of the existing parameters (or both) or by integrating a new parameter in addition to the existing parameters. The interest of the operation is to allow the user to transform the existing hierarchy by the user replacing the initial hierarchy considered obsolete directly in the MT without reconstructing the MDB.

The integration of the new parameter can be carried out according to four scenarios:

- either the parameter replaces both existing $P_{sup}$ and $P_{inf}$ (Tab. 1-a);
- or the parameter replaces the $P_{inf}$ parameter (Tab. 1-b);
- or the parameter replaces the $P_{sup}$ parameter (Tab. 1-c);
- or the parameter is inserted between the parameters $P_{sup}$ and $P_{inf}$ (Tab. 1-d).

The tags added to the two parameters $P_{sup}$ and $P_{inf}$ indicate the selected scenario. The tag $(-)$ indicates that the parameter must not appear in the result while the tag $(+)$ indicates the opposite. In this way it is possible to transform two parameters by creating a new multigranular parameter, while maintaining whole or part of the

possibilities of initial navigations (with drilling operations). For example, in Tab. 1, the scenario (a) removes the possibilities of drilling on the countries and the states (only the multigranular parameter is available) whereas (d) maintains the two initial parameters.

It is important to note that we present here only the possibilities which maintain strict hierarchies (Malinowski and Zimányi, 2006) in which any value of the lower parameter can be dependent on only one value of the higher parameter.

## *5.3 Operator Closure Property*

The definition of the "BLEND" operator respects the closure property: it is applied to a MT and produces a new MT. This property allows chaining successive operations in order to operate complex transformations.

*Example 2.* Let us consider a complex analysis in which a decision maker wishes to compare the cereal surfaces between American states, a country such as Brazil and the Asian continent. This analysis is multigranular on three levels since it uses a continent, a country, and American states (subdivisions of a country). Starting from the MT $T_2$, we chain the two following multigranular transformations:

**BLEND(BLEND($T_2$, Geography, Country (−), State (−), Country<>'USA'), Geography, Continent (−), Country-State (−), Continent='Asia') =** $T_3$

The following figures illustrate the sequence of the two operations with the corresponding multigranular transformations.

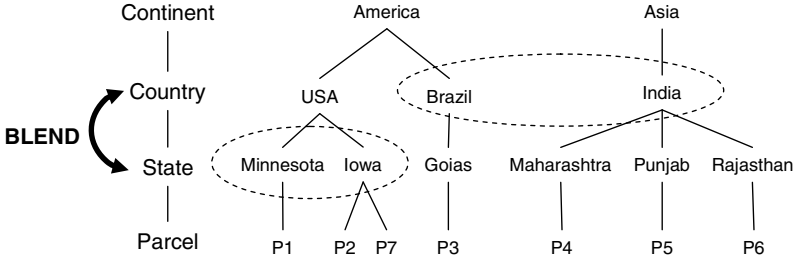



**Fig. 2** Initial structure of Geography in $T_2$

The sequence of the two "BLEND" operations induces a multigranular transformation of the data of $T_2$. The resulting table $T_3$ is presented in Fig. 5.

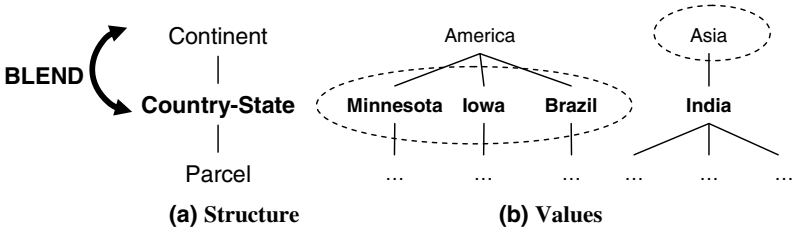



**Fig. 3** Intermediate structure of Geography after the first "BLEND" Fig. 2

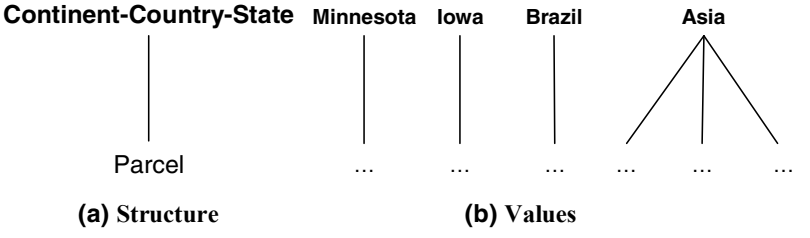



**Fig. 4** Final structure of Geography in $T_3$ after the second "BLEND" Fig. 3

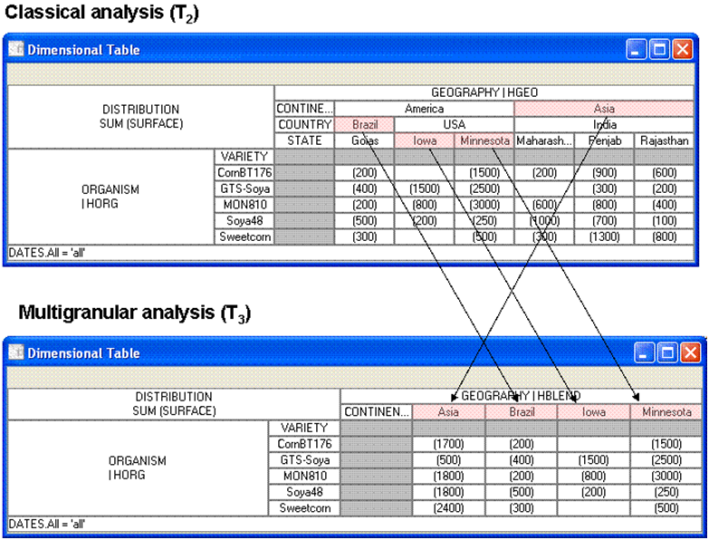
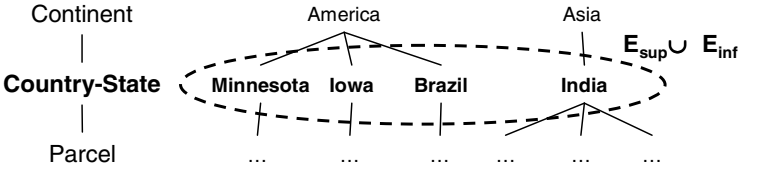
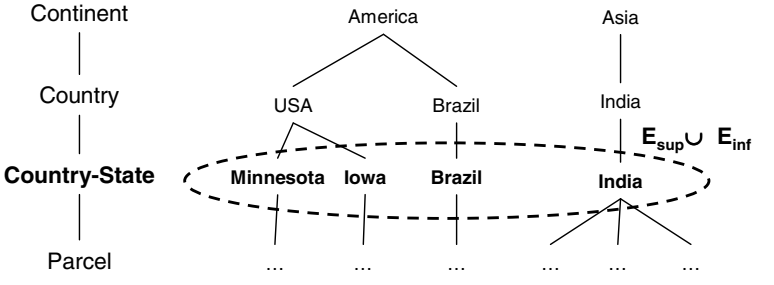



**Fig. 5** Principle of multigranular transformations

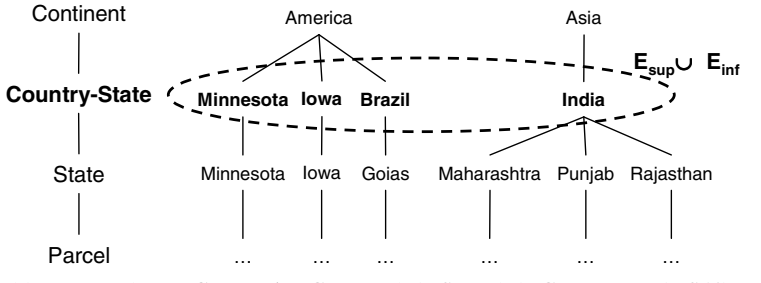
**Table 1** Four possibilities of modification of the hierarchy



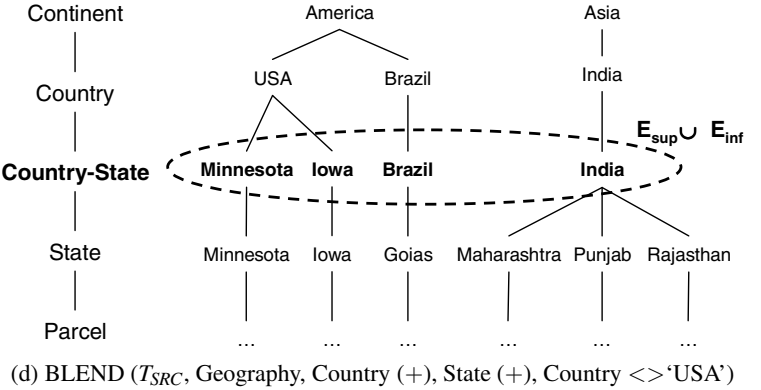

(a) BLEND ($T_{SRC}$, Geography, Country (−), State (−), Country <>'USA')




(b) BLEND ($T_{SRC}$, Geography, Country (+), State (−), Country <>'USA')




(c) BLEND ($T_{SRC}$, Geography, Country (−), State (+), Country <>'USA')




(d) BLEND ($T_{SRC}$, Geography, Country (+), State (+), Country <>'USA')

## 5.4 Special Cases of the Operator

**Empty Selection**

The predicate noted *pred* in the definition of "BLEND" could be an empty selection for the parameters $P_{sup}$ or $P_{inf}$. If $E_{sup}$, respectively $E_{inf}$, is empty, then the operation is also valid. This special case consists in deleting the parameter $P_{sup}$, respectively $P_{inf}$. Note that it is possible to obtain this result using another combination of operators from the OLAP algebra (Ravat *et al.*, 2008). Note also that due to the definition of the operator, $E_{sup}$ and $E_{inf}$ cannot be empty at the same time.

*Example 3.* Let us consider a new operation that combines countries having a strong population density (Density $>$ 20) with states having a week population density. This multigranular transformation is defining as follows:

**BLEND** ($T_3$, **Geography, Country** ($s_{sup}$), **State** ($s_{inf}$), **Density** $>$ **20**)
where $s_{sup} \in \{+, -\}$ and $s_{inf} \in \{+, -\}$.

If country densities of the USA, Brazil and India are respectively 31.15, 21.60, and 300.24 $hab/km^2$, then the predicate 'Density $>$ 20' provides the following sets: $E_{sup}$ = {'USA', 'Brazil', 'India'} and $E_{inf} = \varnothing$. This special case where $E_{inf} = \varnothing$ consists in keeping the countries and deleting their states from the current analysis.

**Root Parameter**

Using the root parameter in the "BLEND" operator implies a dimension multigranular transformation and the associated measure values have to be recalculated. More precisely, deleting the root parameter values requires the aggregation of the associated measure values; e.g., each aggregated value is linked to an upper parameter value.

*Example 4.* Let us consider a multigranular transformation using the root parameter named 'Parcel'. The decision maker wants to compare state surfaces of the USA and parcels of others countries. This multigranular transformation is defined as follows:

**BLEND** ($T_3$, **Geography, State** ($s_{sup}$), **Parcel** ($s_{inf}$), **Country = 'USA'**)

This operation calculates sets such as $E_{sup}$={'Minnesota', 'Iowa'} and $E_{inf}$= {'P3', 'P4', 'P5', 'P6'}. In the resulting multidimensional table, measure values that are linked to the USA ('P1', 'P2' and 'P7') are aggregated to be linked to the states of $E_{sup}$. In the same way of the roll-up operations, the multigranular transformation uses the aggregation function defined from the initial constructor operation noted DISPLAY (see Fig. 1); in this example the SUM function is used.

**Aggregation Functions**

In this paper, we study the operator using the aggregation SUM. This approach can be generalized with every additive function (Golfarelli *et al.*, 1998).

The operator would be applied using other aggregation functions such as average, maximum. However, note that the average is an algebraic function (Gray *et al.*,

1996; Lenz and Thalheim, 2001); i.e. the implementation of the operator is more difficult because only part of the results may be pre-calculated using views, the rest must be calculated from detailed data. For example, the surface of continent is calculated by summing surfaces of countries whereas the temperature of continent cannot be calculated by averaging temperatures of countries. The temperature of a continent is calculated by averaging temperatures of the most detailed data.

## 6 Experiments within R-OLAP Context

The "BLEND" operation is implemented within the Graphic-OLAP tool (Ravat *et al.*, 2008) we have developed in our laboratory using the Java language and the Oracle DBMS. This prototype is implemented according to a R-OLAP approach: the architecture is based on a relational storage of the data and metadata while presenting various interfaces to the user.

The constellation of facts and dimensions is implemented through tables: a set of meta-tables describes the multidimensional structure and a set of tables stores the decisional data available for the analysis. To simplify, our presentation is limited to the tables that store the detailed data; we do not approach the problems of optimization by materialized views (Zhuge *et al.*, 1998; Kotidis and Roussopoulos, 1999). Within this simplified framework, the queries specified by the user are translated into an extraction SQL query on the tables storing the decisional data. Note that the database's structure complexity increases the metadata size. We do not take into account the quantity of meta-data because it does not impact the query process compared to the detailed data.

*Example 5.* The star schema (see Fig. 1) is stored in R-OLAP as a set of relations:
DATES(**id_dates**, month, monthdesc, quarter, year, fouryear)
ORGANISMS(**id_organisms**, variety, category, organismtype)
GEOGRAPHY(**id_geography**, parcel, state, region, country, density, continent)
DISTRIBUTION(**id_repartition**, **id_dates#**, **id_organisms#**, **id_geography#**, surface)

Let us reconsider the "BLEND" operations illustrated in Figs. 1, 2 and 3. The MT $T_3$ of Fig. 5 is obtained from the result of extraction queries generated by Graphic-OLAP. Table 2 shows the SQL queries generated for each operation.

### *6.1 Experiments with Standard Relational SQL*

The experiments we made aim at estimating the operator costs. We study the cost of "BLEND" by translating this algebraic operator into its equivalent SQL query over the star schema. Two queries are compared:

- The first query (R1) uses an attribute that stores the multigranular transformation. This query simulates MDB, which would be modeled according to the user multigranular transformation needs.

**Table 2** SQL translation of "BLEND"

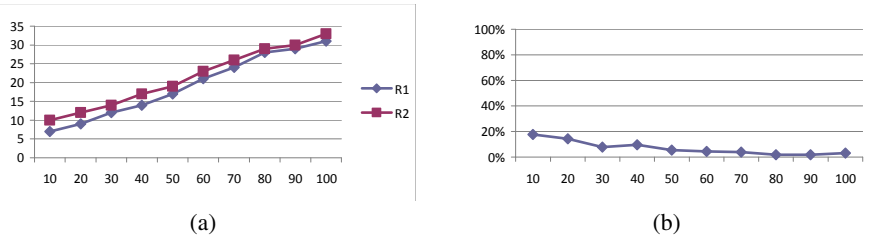
| BLEND($T_2$, Geography, Country(−), State(−), Country <> 'USA') = $T_i$ | BLEND($T_i$, Geography, Continent(−), Country-State(−), Continent = 'Asia') = $T_3$ |
|---|---|
| SQL translation: | SQL translation: |
| **SELECT SUM**(surface) **AS** superficy, continent, country_state, variety<br>**FROM**<br>(**SELECT** surface, continent, pays **AS** country_state, variety<br>**FROM** $T_2$ **WHERE** country<>'USA'<br>**UNION ALL**<br>**SELECT** surface, continent, state **AS** country_state, variety<br>**FROM** $T_2$ **WHERE NOT** (country<>'USA'))<br>**GROUP BY** continent, country_state, variety; | **SELECT SUM**(surface) **AS** surface, continent_country_state, variety<br>**FROM**<br>(**SELECT** surface, continent **AS** continent_country_state, variety<br>**FROM** $T_i$ **WHERE** continent='Asia'<br>**UNION ALL**<br>**SELECT** surface, country_state **AS** continent_country_state, variety<br>**FROM** $T_i$ **WHERE NOT** (continent='Asia'))<br>**GROUP BY** continent_country_state, variety; |

- The second query (R2) calculates the multigranular transformation from the star schema.

Tuples were generated into the ROLAP database's relations according to the following:

- |ORGANISM|= 250
- $10 \leq$ |GEOGRAPHY| $\leq 100$
- |DISTRIBUTION|= |ORGANISM| $\times$ |GEOGRAPHY|

Each relation was completed by multiple indexes on foreign keys. Values were generated using a random function but we make sure that the sizes of generated sets noted $E_{sup}$ and $E_{inf}$ are similar.




(a) (b)

**Fig. 6** Experiment results of BLEND costs

The costs are calculated from the system cost (cost provided by the explain plan of Oracle 11g Application Server). The experiments aim at showing how much the operator "BLEND" costs. The size represents the number of tuples in GEOGRAPHY (from 10 to 100) and DISTRIBUTION (from 250×10 to 250×100); the size of ORGANISM is fixed to 250 tuples. Figure 6(a) compares the queries. Naturally (R2) is more expensive than (R1) due to computation of multigranular transformation. The cost is not very important (between 18% and 2%). As Fig. 6(b) shows,

this result is interesting because the cost falls according to the relation sizes are increased.

We also investigated if results remain similar when sizes of $E_{sup}$ and $E_{inf}$ are different. We use |GEOGRAPHY|= 200 and |DISTRIBUTION|= 50000. Figure 7 shows costs of (R2) when sizes of $E_{inf}$ and $E_{sup}$ are modified: the axis $x$ represents GEOGRAPHY size whereas |DISTRIBUTION|= |ORGANISM|$\times$ |GEOGRAPHY|. We can see that cost is constant, and the size difference between $E_{inf}$ and $E_{sup}$ seems not to influence the BLEND operator cost.

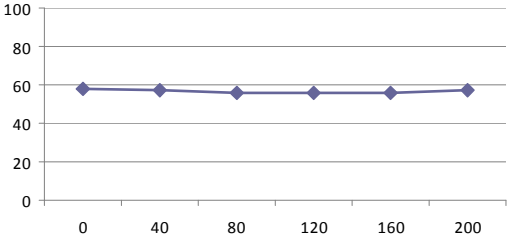



**Fig. 7** R2 cost according to the data distribution between $E_{sup}$ and $E_{inf}$

## 6.2 *Experiments with Oracle SQL3/OLAP*

We performed second experiment series in Oracle SQL3/OLAP using the GROUP BY CUBE operator. We compared (R2) with its equivalent query (R3) using the cube operator (Gray *et al.*, 1996). Figure 8 compares queries (R2) and (R3). We can note that the Oracle GROUP BY CUBE implementation is faster than the standard GROUP BY operation.

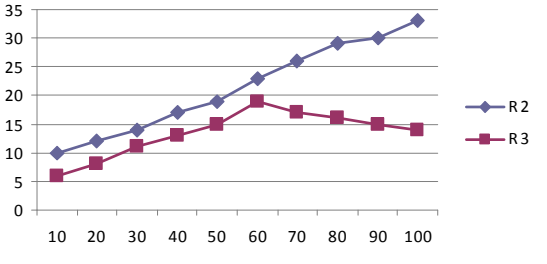



**Fig. 8** Comparison with cube operator

## 7 Conclusion

This paper deals with complex analyses consisting in combining parameters of different granularities. Such analyses known as multigranular are not easily performed with traditional systems since they require organizing the data according to each analysis. We introduce a new algebraic operator for OLAP manipulations, called "BLEND". We study the limits of its use on strict hierarchies. The approach allows transforming a hierarchy by maintaining the initial possibilities of navigation. In order to establish the feasibility of this proposal, the operator has been implemented in a R-OLAP context within the Oracle DBMS.

In the short term, a first prospect is to carry out a study on the possible techniques of operator optimization, in particular by exploiting lattices of materialized views set up within the MDB. The expression of the operation in our graphic language (Ravat *et al.*, 2007) also constitutes a direct extension of this work. We also project to study other principles of multigranular transformations in more complex contexts such as non-strict hierarchies.

## References

Abelló, A., Samos, J., Saltor, F.: YAM2: a multidimensional conceptual model extending UML. Inf. Syst. 31(6), 541–567 (2006), http://dx.doi.org/10.1016/j.is.2004.12.002

Blaschka, M., Sapia, C., Höfling, G.: On Schema Evolution in Multidimensional Databases. In: Mohania, M., Tjoa, A.M. (eds.) DaWaK 1999. LNCS, vol. 1676, pp. 153–164. Springer, Heidelberg (1999)

Chaudhuri, S., Dayal, U.: An overview of data warehousing and OLAP technology. SIGMOD Rec. 26(1), 65–74 (1997), http://doi.acm.org/10.1145/248603.248616

Eder, J., Koncilia, C., Mitsche, D.: Automatic Detection of Structural Changes in Data Warehouses. In: Kambayashi, Y., Mohania, M.K., Wöß, W. (eds.) DaWaK 2003. LNCS, vol. 2737, pp. 119–128. Springer, Heidelberg (2003)

Espil, M.M., Vaisman, A.A.: Efficient intensional redefinition of aggregation hierarchies in multidimensional databases. In: DOLAP 2001: Proceedings of the 4th ACM international workshop on Data warehousing and OLAP, pp. 1–8. ACM, New York (2001), http://doi.acm.org/10.1145/512236.512237

Favre, C., Bentayeb, F., Boussad, O.: Dimension Hierarchy Updates in Data Warehouses: a User-driven Approach. In: 9th International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Madeira, Portugal, pp. 206–211 (2007)

Golfarelli, M., Maio, D., Rizzi, S.: Conceptual Design of Data Warehouses from E/R Schema. In: HICSS 1998: Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences, vol. 7, pp. 334–343. IEEE Computer Society, Washington (1998), http://dx.doi.org/10.1109/HICSS.1998.649228

Golfarelli, M., Rizzi, S., Saltarelli, E.: WAND: A CASE Tool for Workload-Based Design of a Data Mart. In: SEBD, pp. 422–426 (2002)

Gray, J., Bosworth, A., Layman, A., Pirahesh, H.: Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Total. In: ICDE 1996: Proceedings of the Twelfth International Conference on Data Engineering, pp. 152–159. IEEE Computer Society, Washington (1996)

Gyssens, M., Lakshmanan, L.V.S.: A Foundation for Multi-dimensional Databases. In: VLDB 1997: Proceedings of the 23rd International Conference on Very Large Data Bases, pp. 106–115. Morgan Kaufmann Publishers Inc., San Francisco (1997)

Hurtado, C.A., Mendelzon, A.O., Vaisman, A.A.: Maintaining Data Cubes under Dimension Updates. In: International Conference on Data Engineering, vol. 0, pp. 346–355 (1999), http://doi.ieeecomputersociety.org/10.1109/ICDE.1999.754950

Kimball, R.: The data warehouse toolkit: practical techniques for building dimensional data warehouses. John Wiley & Sons, Inc., New York (1996)

Kotidis, Y., Roussopoulos, N.: DynaMat: a dynamic view management system for data warehouses. In: SIGMOD 1999: Proceedings of the 1999 ACM SIGMOD international conference on Management of data, pp. 371–382. ACM, New York (1999), http://doi.acm.org/10.1145/304182.304215

Lenz, H.-J., Thalheim, B.: OLAP Databases and Aggregation Functions. In: SSDBM 2001: Proceedings of the 13th International Conference on Scientific and Statistical Database Management, pp. 91–100. IEEE Computer Society, Washington (2001)

Malinowski, E., Zimányi, E.: Hierarchies in a multidimensional model: from conceptual modeling to logical representation. Data Knowl. Eng. 59(2), 348–377 (2006), http://dx.doi.org/10.1016/j.datak.2005.08.003

Ravat, F., Teste, O., Tournier, R., Zurfluh, G.: Graphical Querying of Multidimensional Databases. In: Ioannidis, Y.E., Novikov, B., Rachev, B. (eds.) ADBIS 2007. LNCS, vol. 4690, pp. 298–313. Springer, Heidelberg (2007)

Ravat, F., Teste, O., Tournier, R., Zurfluh, G.: Algebraic and Graphic Languages for OLAP Manipulations. IJDWM 4(1), 17–46 (2008)

Romero, O., Abelló, A.: On the Need of a Reference Algebra for OLAP. In: Song, I.Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 99–110. Springer, Heidelberg (2007)

Vassiliadis, P., Sellis, T.: A survey of logical models for OLAP databases. SIGMOD Record 28(4), 64–69 (1999), http://doi.acm.org/10.1145/344816.344869

Vassiliadis, P., Simitsis, A., Skiadopoulos, S.: Modeling ETL activities as graphs. In: Lakshmanan, L.V.S. (ed.) DMDW. CEUR Workshop Proceedings, vol. 58, pp. 52–61 (2002) CEUR-WS.org

Zhuge, Y., Garcia-Molina, H., Wiener, J.L.: Consistency Algorithms for Multi-Source Warehouse View Maintenance. Distrib. Parallel Databases 6(1), 7–40 (1998), http://dx.doi.org/10.1023/A:1008698814840

# A New Approach for Unsupervised Classification in Image Segmentation

Sébastien Lefèvre

**Abstract.** Image segmentation is a fundamental problem in image analysis and understanding. Among the existing approaches proposed to solve this problem, unsupervised classification or clustering is often involved in an early step to partition the space of pixel intensities (i.e. either grey levels, colours or spectral signatures). Since it completely ignores pixel neighbourhoods, a second step is then necessary to ensure spatial analysis (e.g. with a connected component labeling) in order to identify the regions built from the segmentation process. The lack of spatial information is a major drawback of the classification-based segmentation approaches, thus many solutions (where classification is used together with other techniques) have been proposed in the literature. In this paper, we propose a new formulation of the unsupervised classification which is able to perform image segmentation without requiring the need for some additional techniques. More precisely, we introduce a *k*means-like method where data to be clustered are pixels themselves (and not anymore their intensities or colours) and where distances between points and class centres are not anymore Euclidean but topographical. Segmentation is then an iterative process, where at each iteration resulting classes can be seen as influence zones in the context of mathematical morphology. This comparison provides some efficient algorithms proposed in this field (such as hierarchical queue-based solutions), while adding the iterative property of unsupervised classification methods considered here. Finally, we illustrate the potential of our approach by some segmentation results obtained on artificial and natural images.

**Keywords:** Image Segmentation, Clustering, kmeans, Mathematical Morphology.

Sébastien Lefèvre
LSIIT – CNRS / University of Strasbourg, Pôle API, Bd Brant,
BP 10413, 67412 Illkirch Cedex, France
e-mail: lefevre@unistra.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 113–131.
springerlink.com 

## 1 Introduction

Classification methods, either supervised or unsupervised, have always been very useful tools in the field of digital image analysis and processing, especially with the aim of image segmentation or image understanding. Conversely, images can be seen as semi-structured complex data which offer new perspectives and new challenges to the field of data mining and knowledge discovery.

We study here the existing link between these two fields, i.e. image processing and data mining. More precisely we focus on the role of (unsupervised) classification in solving one of the main problems of image processing, i.e. image segmentation. A segmentation, or a partition of an image into regions, can indeed be obtained from an unsupervised classification applied on all pixel values (e.g. grey levels, colours or spectral signatures) followed by a spatial analysis (e.g. connected component labeling). Despite being very commonly used, this segmentation strategy is not perfect since spatial information is taken into account only in a second step and is completely ignored during the classification phase. Thus numerous ad-hoc solutions have been proposed to solve this problem, as we will see in Sec. 2.2.

Instead of proposing yet another ad-hoc technique for classification-based image segmentation, we consider here the following question: is it possible to apply such a classification *directly* in the (spatial) space of image pixels ? If so, it would provide a straight way to use classification methods to solve the image segmentation problem. Of course, such a strategy would require some adaptations to let the classification method be able to produce directly a relevant segmentation result without any help from additional postprocessing. Main concepts within the classification paradigm (e.g. data space, similarity or distance measure) have to be rethought in the context of image segmentation. Moreover underlying algorithms may benefit from advances in the image processing field. Our contribution concerns such an approach, and we focus more precisely on one of the most famous unsupervised classification algorithms, i.e. the *k*means method. Thus we reformulate here the *k*means algorithm in the context of image segmentation, which does not require any additional step.

This chapter is organised as follows. Sec. 2 recalls how unsupervised classification (and especially partitional clustering) is used to solve the problem of image segmentation. After pointing out the major drawbacks of the classical segmentation-by-clustering paradigm, we then review the main solutions proposed in the literature. Our proposal is explained in more details in Sec. 3. Basically it consists in performing classification directly in the pixel space rather than in the intensity space. This requires avoiding the use of Euclidean distance and gravity centres classically used in the *k*means algorithm. In Sec. 4, we study the link with the field of mathematical morphology, in order to benefit from efficient algorithms. Thus the proposed scheme of unsupervised classification for image segmentation can also be seen as an iterative morphological segmentation process. Finally, we illustrate in Sec. 5 our approach by some first segmentation results obtained on artificial and natural images.

## 2 On the Limitations of Classification for Image Segmentation

In this section, we introduce the main notations and explain why image segmentation cannot be achieved with classification. We then draw up an overview of existing approaches proposed with the aim of adapting classification methods to the problem of image segmentation.

### *2.1 Segmentation versus Classification*

An image is usually defined as a function $f : \mathcal{E} \rightarrow \mathcal{T}$ which assigns to each pixel $p = (x,y)$ taken in space $\mathcal{E} \subset \mathbb{N}^2$, a value $v = f(p)$ in $\mathcal{T}$. This value may be for instance a grey level ($\mathcal{T} = \{0,\ldots,255\}$), a colour described by its tristimulus chromatic representation ($\mathcal{T} = \{0,\ldots,255\}^3$ or $\mathcal{T} = [0,1]^3$ after normalisation) or a spectral signature ($\mathcal{T} \subset \mathbb{R}^n$ for an image with $n$ spectral bands).

Segmentation aims at partitioning the pixel space $\mathcal{E}$ of an image $f$ into a set of $K$ regions $\{R_k\}_{1\leq k\leq K}$ which are homogeneous according to a given criterion (e.g. the values $v$ of pixels composing each region). Thus it is a function $\pi : \mathcal{E} \rightarrow \mathcal{C}$ which assigns to each pixel $p$ the index $k$ of component or region $R_k$ to which it belongs. Each region $R_k$ is built as a connected component (see Fig. 1(b)), i.e. a set of adjacent pixels (or neighbours two by two) with value $k$. More formally, we define a discrete path $P_{pq}$ from pixel $p$ to pixel $q$ as the set of pixels $P_{pq} = \{p_i\}_{0\leq i\leq m}$ with $p_0 = p$, $p_m = q$, and $\forall i \in \{0,\ldots,m-1\}$, $p_i$ is adjacent to $p_{i+1}$. Two pixels $p = (x_p,y_p)$ and $q = (x_q,y_q)$ are considered $\lambda$-adjacent if their $\lambda$-distance equals 1, that is $d_\lambda(p,q) = 1$. We define 4-distance, 8-distance and Euclidean distance respectively by:

$$d_4(p,q) = |x_p - x_q| + |y_p - y_q| \tag{1}$$

$$d_8(p,q) = \max(|x_p - x_q|, |y_p - y_q|) \tag{2}$$

$$d_E(p,q) = \sqrt{(x_p - x_q)^2 + (y_p - y_q)^2} \tag{3}$$

but these terms used in discrete geometry / digital image processing are equivalent to more general notions of city-block / Manhattan / boxcar / absolute value (for $d_4$) and Chebyshev / maximum value (for $d_8$) distances. The neighbourhood of a pixel $p$ is written $\mathcal{N}(p)$ and defined as the set of pixels $q$ adjacent to $p$. Each path $P_{pq}$ is associated with a cost $\omega(P_{pq})$, which may be defined as the number of pixels it contains, or relying on distances between pixels such as:

$$\omega(P_{pq}) = \sum_{i=1}^{m} d(p_{i-1}, p_i) \tag{4}$$

with $d$ a given distance measure, e.g. Eqs. (1) to (3). Finally, a region $R_k$, as a connected component, verifies $\forall p,q \in R_k$, $\exists P_{pq}$ such that $\forall p_i \in P_{pq}$, $p_i \in R_k$ or in other words $\pi(p_i) = k$.

Unsupervised classification or data clustering aims at gathering data into homogeneous sets called classes or clusters. Applied on a digital image, the classification

may be represented as a function $\pi : \mathcal{E} \rightarrow \mathcal{C}$ which assigns to each pixel $p$ the index $k$ of the class $C_k$ to which it belongs. Similarly to the case of regions $R_k$, the content of classes $C_k$ is expected to be homogeneous (e.g. pixels composing a given class share similar $v$ values). However, contrary to regions $R_k$ produced by a segmentation, classes $C_k$ do not have the property of connexity, i.e. they are not assumed to be connected components, as illustrated by Fig. 1(c).

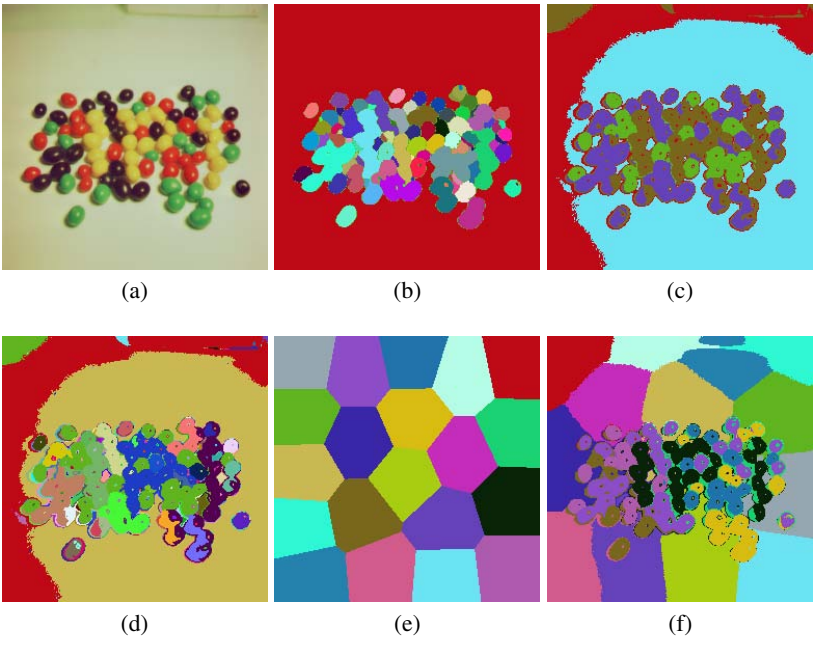



**Fig. 1** Difference between segmentation and classification: (a) input image, (b) region segmentation, (c) spectral classification, (d) spatially refined classification, (e) spatial classification, (f) spatial/spectral classification

Here we focus more precisely on the case of unsupervised classification using a *k*means-like approach (Kaufman and Rousseeuw, 1990). In this algorithm, the number $K$ of classes $\{C_k\}_{1 \leq k \leq K}$ is a priori defined (even if there exists some other partitional algorithms which overcome this condition). Each class $C_k$ is characterized by its centre noted $c_k$. Initialisation of the class centres may rely on some assumptions available on the data, or may be performed randomly if no such knowledge is available. The algorithm consists of two iterative steps performed until convergence. First, for each pixel the distance to the different class centres are computed and the pixel is assigned to the class with the closest centre. Then class centres are updated by using the new data partition.

When involved to solve the problem of image segmentation, the *k*means algorithm is applied in the pixel value space $\mathcal{T}$ rather than the pixel space $\mathcal{E}$, i.e. data to

be classified are the pixel values $v$ rather than directly the pixels $p$. Alg. 1 describes this process, which tends to minimise a global cost function $J$, e.g. defined by:

$$J = \sum_{k=1}^{K} \sum_{p \in C_k} \big(d(f(p), c_k)\big)^2 \tag{5}$$

**Algorithm 1.** original *k*means algorithm for image segmentation

**Input**: Image $f : \mathcal{E} \to \mathcal{T} : p \mapsto f(p)$
**Input**: Number $K$ of classes
**Output**: Set of classes $\{C\}_K$ or classification map $\pi : \mathcal{E} \to \mathcal{C} : p \mapsto \pi(p)$

*/* initialisation of classification map and class centres */*
**foreach** *pixel* $p$ **do** $\pi^0(p) \leftarrow \emptyset$
**foreach** *class centre* $c_k$ **do** $c_k^0 \leftarrow$ RANDOM $(\mathcal{T})$
*/* iterative assignment-update process */*
**repeat**
  $l \leftarrow l + 1$
  **foreach** *pixel* $p$ **do** */* pixel-to-class assignement */*
    computation of distances to the different class centres, i.e. $d(f(p), c_k^{l-1})$
    assignment to the class with closest centre, i.e. $\pi^l(p) = \arg\min_k d(f(p), c_k^{l-1})$
  **foreach** *class centre* $c_k$ **do** */* update of class centres */*
    $c_k^l = \text{avg}\{f(p_i) \mid \pi^l(p_i) = k\}$ with avg the average function
**until** $\pi^l = \pi^{l-1}$ */* stability partition as convergence criterion */*

The distance $d$ involved in the algorithm and in Eq. (5) is the Euclidean distance computed in the $n$-dimensional space $\mathcal{T}$, that is $d_E(v,w) = \sqrt{\sum_{i=1}^{n}(v_i - w_i)^2}$ with $v = (v_1, \ldots, v_n)$ and $w = (w_1, \ldots, w_n)$. The convergence criterion (partition stability) may be somewhat relaxed and replaced by a convergence of the cost function, or even a finite number of iterations.

In the previous algorithm (see also Fig. 1(c)), it is clear that the location of a pixel $p$ (i.e. its coordinates $(x,y)$) has been completely ignored, contrary to its value $f(p)$. Two pixels $p$ and $q$ with close values $f(p)$ and $f(q)$ will then be most probably assigned to the same class $C_k$ even if they form two disjoint connected components in the classification map $\pi$. Thus some additional processings are required to obtain a segmentation (i.e. a partition of the image into homogeneous, connected classes).

## *2.2 Classification Relying on Spatial Information*

In order to obtain a segmentation from a classification, it is necessary to involve spatial information (Haralick and Shapiro, 1985). We give here a brief survey of main approaches addressing this problem in the literature. In this survey, we do not consider approaches such as Markov Random Fields which combine classification

and spatial regularisation, even if these approaches can be seen as generalisations of classification methods such as *k*means (Pappas, 1992). We focus instead on approaches such as *k*means or fuzzy *c*means.

The most commonly adopted solution consists of a postprocessing, where a connected component labeling is applied on the classification result (see Fig. 1(d)). It aims at gathering into a single region $R_k$ all adjacent pixels $p$ which have been assigned to the same class $C_j$. A region $R_k$ is then a connected component verifying $\forall p \in R_k,\ \pi(p) = j$ in the classification map. A class $C_j$ is thus split into several regions $R_k$ (unless if it already ensures the connexity property).

However, applying a classification at the pixel level and then gathering pixels of similar class into connected components often results in a segmentation map where many regions contain only a single isolated pixel (instead of a pixel aggregate). So a filtering step can be involved in order to remove these isolated pixels and to reassign them to neighbouring regions. In the case of a fuzzy classification, the postprocessing to be applied can be even a segmentation algorithm such as region growing (Eum *et al.*, 1996), region merging (Chen and Lu, 2002), or marker-based watershed (Lezoray and Cardot, 2002).

Involving spatial information can also be done through a preprocessing step. This can be a segmentation into regions, which result (i.e. regions) will be further processed with a subsequent classification (but in this case the result is rather a region-based classification than a segmentation). Moreover, one can describe each pixel $p$ by its value $v$ but also by its coordinates $(x,y)$ (Krishnapuram and Freg, 1992) as illustrated in Figure 1(f). It is also possible to perform an image interpolation, where the estimated value $\hat{f}(p)$ in each pixel $p$ is computed from its 4 or 8 neighbours (e.g. through a mean or median). Classification is then applied on pixels using values $\hat{f}$ alone or in complement with initial image $f$ (Turi, 2001; Chen and Zhang, 2004).

Another way is to consider spatial information directly within the classification algorithm. In Ilea and Whelan (2008), textural information can be associated with colour information. While the former is measured by a gradient, the latter is obtained by applying a *k*means on a smoothed image (resulting from an anisotropic filtering) and represented in two different colour spaces. The dissimilarity measure between a pixel and a class relies on these information, which are related to pixels but also to classes, and updated at each iteration in order to track the evolution of class content.

In Liew *et al.* (2000), the dissimilarity between a pixel and a class centre within the fuzzy *c*means algorithm is defined using the neighbourhood of the pixels: contribution of each neighbour (computed as a distance to the corresponding class centre) is proportional to the similarity between this neighbour and the considered pixel (the similarity being measured in the feature space). This approach is inspired from Tolias and Panas (1998) where similarity between the pixel to be classified and its neighbour modifies the class membership probability of this pixel (by adding or removing of a constant). This similarity between neighbours can be directly computed in the class membership space (Noordam *et al.*, 2000; Xia *et al.*, 2007). A similar approach using a probabilistic classification based on neighbour kriging indicators is proposed in Pham (2001).

Applying a classifier in a multiscale framework, through a hierarchical method, offers another mean to take into account spatial context. In Luo *et al.* (2003), the authors propose a new algorithm where the following steps are applied successively: image subsampling, computation of features related to colour and texture, spatially constrained classification, and region merging. These two last steps are performed iteratively at each level of the hierarchy, thus allowing to isolate the main connected components (and to aggregate remaining pixels to these components), and then to apply once again the *k*means algorithm independently on each connected component. This principle of successive classifications is also presented in Cheng and Sun (2000) where clusters are built only from pixels with homogeneous neighbourhood. Finally, the *k*means algorithm can be applied on a sliding window within the image, where class centres are propagated during image scan (Leydier *et al.*, 2004).

To summarize, various solutions for spatialising classification methods have been proposed, in order to make them adequate for image segmentation: preprocessing, postprocessing, new attributes, multiscale approach, classification correction of one pixel depending on its neighbours, etc. However, to the best of the author's knowledge, applying a classification directly in the spatial domain (i.e. pixel space) has not been studied yet, despite the fact that it seems a very intuitive solution to consider the spatial behaviour of the segmentation. So we propose in next section such a solution, which can be seen as a spatial classification using the *k*means algorithm.

## 3 Another Usage of Classification in Image Segmentation

To be applied successfully on the problem of image segmentation, a classification method has to rely on spatial information. Instead of using additional steps in the classification process, here we rather reformulate the classification in a spatial context. Thus we propose to spatialise the classification method in order to make it usable to perform image segmentation.

### *3.1 Spatial Classification*

We have pointed out in Sec. 2.1 that the only difference between a segmentation and a classification was the lack of spatial connexity constraint in the result of the latter. Modifying the behaviour of a classification algorithm in order to ensure spatial connexity of the produced classes (or regions in this context) is a way to elaborate a classification method intrinsically able to solve the problem of image segmentation. Since various classification algorithms exist, we have decided here to focus on one of its most representative, the *k*means algorithm, and to study how the spatial connexity property can be provided to it. In other words, our contribution is related to the spatialisation of the *k*means algorithm.

Spatial information is directly present in images through notions of connexity and adjacency between neighbouring pixels. Thus we propose to apply the classification in the pixel space $\mathcal{E}$ and not anymore in a feature space (e.g. the pixel value space $\mathcal{T}$). The different operations (distance $d$ between points and class centres during

assignment step, update of class centres with mean function avg during update step) are also performed in the space $\mathcal{E}$, considering $p$ rather than $f(p)$ in the different formula. Similarly, class centres $c_k$ will be defined in $\mathcal{E}$ (i.e. they are image pixels) and not anymore in $\mathcal{T}$. Thus the criterion to be optimized in the clustering process is slightly modified to become

$$J = \sum_{k=1}^{K} \sum_{p \in C_k} d(p, c_k) \tag{6}$$

while the original criterion given in Eq. (5) was using $f(p)$.

However, applying directly the *k*means algorithm in the space $\mathcal{E}$ will not have the expected result for segmentation: since image pixels are regularly located on a square grid of finite size, ignoring pixel values during pixel-to-class assignment step will lead to a result independent of the image content (Fig. 1(e))! An intermediary solution has to be elaborated in order to take into account both pixels $p$ and their values $f(p)$ during the computation of distance $d(p, c_k)$ between pixels and class centres. This will be studied in next section.

Another problem has also to be solved to ensure an adequate behaviour of the *k*means algorithm in the context of segmentation. It is related to the inherent inability of the algorithm to deal with non convex sets. During a segmentation step, it is not feasible to assume that all regions would be convex. In the case of a concave region, updating the class (or region) centre based on the computation of the gravity centre can result in locating the centre outside the class. As we will see in the next section, we rely on the assumption that the centre belongs to its region or class to ensure the spatial connexity of produced classes. Thus we need to replace the gravity centre by another measure and we propose to use here a kind of median (thus leading to a *k*median-like algorithm rather than a *k*means-like one). Moreover, we propose to keep class centres in $\mathcal{E}$ (i.e. choosing the class centres among the pixels present in the image), like a *k*prototypes approach (Han *et al.*, 2001) (but we still keep the overall process of the *k*means algorithm). In order to obtain such a median measure in $\mathcal{E}$, we propose to define the class centre as its central or inner most pixel, that can be obtained by means of techniques from mathematical morphology (Soille, 2003). Implementation details will be given in Sec. 4.

Let us notice that we have chosen to consider here independently the pixel space $\mathcal{E}$ and the pixel value space $\mathcal{T}$. We could have formulate directly a solution in the space $\mathcal{E} \times \mathcal{T}$, e.g., by setting some constraints on the classes to be obtained or by defining some appropriate representation and allocation functions. But applying a classification in such a space $\mathcal{E} \times \mathcal{T}$ would have brought a much higher computational complexity. Nevertheless, since our algorithm shares some common properties with dynamic clouds (Diday, 1971) or *k*medoids (Kaufman and Rousseeuw, 1990) for instance, the link with such methods should be studied more deeply.

### 3.2 Topographic Distance and Connexity Property

When assigning a pixel to a given class, we have to take into account both pixel coordinates and value. Thus we propose to combine $p$ and $f(p)$ through a topographical distance for which we give some definitions below. Let us notice that updating class centres can also rely on this distance and on values $p$ and $f(p)$, as we will see further.

Let us represent an image as a topographical surface, where each pixel $p$ is associated to its elevation, i.e. its value $f(p)$. In this context, the cost of a path $P_{pq}$ can be reformulated by taking into account its topography or the observed difference in height between $p$ and $q$, using both the difference between values $f(p)$ and $f(q)$ and the (spatial) distance $d(p,q)$ between pixels $p$ and $q$ (or more precisely, between their respective locations, involving a distance measure among those already defined in Eqs. (1)-(3)). Thus, several definitions for the topographical cost of a path have been proposed (Prêteux, 1992; Meyer, 1994; Philipp-Foliguet, 2000). The cost term introduced by Prêteux (1992) can be simply formulated in the case of a finite discrete image as the highest pixel on the path:

$$\omega(P_{pq}) = \max_{p_i \in P_{pq}} f(p_i) \tag{7}$$

or in other words the path ridge, with the convention $\omega(P_{pp}) = -\infty$.

Meyer (1994) considers the steepest slope defined for a pixel $p$ as

$$\delta(p) = \max_{\substack{q \in \mathcal{N}(p) \\ f(q)<f(p)}} \left( \frac{f(p)-f(q)}{d(p,q)} \right) \tag{8}$$

which can be simply computed with a morphological erosion. The topographical cost between two neighbouring pixels is defined as

$$q \in \mathcal{N}(p), \quad \omega(p,q) = \begin{cases} \delta(p) & \text{if } f(p) > f(q) \\ (\delta(p)+\delta(q))/2 & \text{if } f(p) = f(q) \\ \delta(q) & \text{if } f(p) < f(q) \end{cases} \tag{9}$$

The topographical cost of a whole path is then computed as

$$\omega(P_{pq}) = \sum_{i=1}^{m} \left( d(p_{i-1},p_i) \cdot \omega(p_{i-1},p_i) \right), \quad P_{pq} = \{p_i\}_{0 \le i \le m} \tag{10}$$

Philipp-Foliguet (2000) proposes a simpler definition which avoids to seek for ridges or steepest slopes, using a weighting coefficient $\kappa$ (usually $\kappa = 1$) :

$$\omega(P_{pq}) = \sum_{i=1}^{m} \left( \kappa \cdot d(p_{i-1},p_i) + |f(p_{i-1}) - f(p_i)| \right), \quad P_{pq} = \{p_i\}_{0 \le i \le m} \tag{11}$$

Finally, whatever the definition of the topographical cost in use, the topographical distance $d_T(p,q)$ is defined as the minimum cost of a path from $p$ to $q$:

$$d_T(p,q) = \min_{P_{pq}}(\omega(P_{pq})) \tag{12}$$

Let us notice that $d_T$ is a *true* distance function only with the definition of Philipp-Foliguet. In the other cases, we have $\forall p,q,\ \ d_T(p,q) = 0 \nLeftrightarrow p = q$. Prêteux proposes to use $\exp(d_T(p,q))$ while Meyer presents different techniques for modifying the topographical surface in order to deal with the problematic case of plateaus (where we have $d_T(p,q) = 0$ with $p \neq q$). Let us state that another pseudo-distance measure (i.e. a distance function that does not satisfy the identity of indiscernibles) could be obtained by simply defining the cost as

$$\omega(P_{pq}) = \sum_{i=1}^{m} |f(p_i) - f(p_{i-1})|, \quad P_{pq} = \{p_i\}_{0 \leq i \leq m} \tag{13}$$

In the following, we will use a simplified version of the Philipp-Foliguet measure:

$$\omega(P_{pq}) = \sum_{i=1}^{m} \omega(p_{i-1}, p_i), \quad P_{pq} = \{p_i\}_{0 \leq i \leq m} \tag{14}$$

and consider a 8-distance where the topographical cost $\omega(p,q)$ between two neighbouring pixels $p$ and $q$ is given by

$$\omega(p,q) = |f(p) - f(q)| + \varepsilon, \ \ q \in \mathcal{N}_8(p) \tag{15}$$

with the convention $\omega(p,p) = 0$, and $\varepsilon \ll \nabla f$ being a very small term (i.e. lowest than the minimal absolute difference between values of two neighbouring pixels) ensuring the identity of indiscernibles. An illustration of this distance is given in Fig. 2 through the computation of a distance transform from a given point in a grey level image.

We recall that our goal is to ensure the spatial connexity of classes $C_k$ produced by the classification algorithm. Let us check that this property is verified by the pixel-to-class assignment step. Using the definition introduced in Sec. 2.1, a class $C_k$ is connected if $\forall p,q \in C_k,\ \ \exists P_{pq}$ such that $\forall p_i \in P_{pq},\ \ p_i \in C_k$. To simplify, we will use in this case the notation $P_{pq} \in C_k$. The assignment of a pixel $p$ to $C_k$ is performed if $d_T(p,c_k) < d_T(p,c_j),\ \ \forall j \neq k$. Obviously the centre $c_k$ of the class $C_k$ will stay assigned to $C_k$ since $d_T(c_k,c_j) > d_T(c_k,c_k) = 0,\ \ \forall j \neq k$. The connexity notion can then be written as $\forall p,q \in C_k,\ \ \exists (P_{pc_k}, P_{qc_k}) \in C_k$ or simply by $\forall p \in C_k,\ \ \exists P_{c_k p} \in C_k$. Let us denote by $q$ the pixel preceding $p$ in this path $P_{c_k p}$ of minimal cost, we have then $q = \arg\min_{p_i \in \mathcal{N}(p)} d_T(c_k,p_i)$ and $d_T(c_k,p) = d_T(c_k,q) + d_T(q,p)$. We can prove the connexity of $C_k$ by showing that the assumption $q \notin C_k$ is not valid. Let us assume $q \in C_j,\ \ j \neq k$, we have then $d_T(c_k,q) > d_T(c_j,q)$. Combining the previous formula, we obtain $d_T(c_k,p) > d_T(c_j,q) + d_T(q,p) \geq d_T(c_j,q)$. This inegality is not valid since it would mean the assignment of $p$ to $C_j$. So the proposed scheme for


**Fig. 2** Illustration of the topographical distance computed from the top-left pixel (from left to right): input image, topographical distance transform, look-up table representing the distance scale

pixel-to-class assignment ensures to produce connected classes during the classification process.

## 4 Implementation Issues

The *k*means algorithm[1] that we proposed here differs from its original version only on the few following aspects, as underlined in Algorithm 2: (1) the space where pixels are represented and centres are selected ($\mathcal{E}$ instead of $\mathcal{T}$); (2) the distance measure $d_T(p, c_k)$ which is not Euclidean anymore but rather topographical; (3) the class centre computation method which does not rely on the mean but rather on the median. These changes ensure our algorithm an adequate behaviour for image segmentation, since the resulting classification is also intrinsically a segmentation. Moreover, they provide a link with the field of mathematical morphology, which brings some efficient implementations.

Indeed, the topographical distance has been used in the context of mathematical morphology to define segmentation methods such as watershed or skeleton by influence zone (Prêteux, 1992; Meyer, 1994). The pixel-to-class assignment step of our *k*means algorithm can then be seen as a morphological segmentation with markers corresponding to class centres. Thus it is possible to benefit from algorithmic developments in mathematical morphology, and especially hierarchical queue-based algorithms for marker-based segmentation for which we can find implementation details in the book of Soille (2003).

[1] Let us observe that the term *kmeans* is not perfectly adequate here, since our algorithm is more of *kmeans-like* type.

**Algorithm 2.** Proposed *k*means algorithm for image segmentation

**Input**: Image $f : \mathcal{E} \rightarrow \mathcal{T} : p \mapsto f(p)$
**Input**: Number $K$ of classes (i.e. regions)
**Output**: Set of classes $\{C\}_K$ or classification (i.e. segmentation) map
$\pi : \mathcal{E} \rightarrow \mathcal{C} : p \mapsto \pi(p)$

*/* initialisation of classification map and class centres */*
**foreach** *pixel* $p$ **do** $\pi^0(p) \leftarrow \emptyset$
**foreach** *class centre* $c_k$ **do** $c_k^0 \leftarrow$ RANDOM $(\mathcal{E})$
*/* iterative assignment-update process */*
**repeat**
  $l \leftarrow l+1$
  **foreach** *pixel* $p$ **do** */* pixel-to-class assignment */*
    computation of topographical distances to the different centres, i.e. $d_T(p, c_k^{l-1})$
    assignment to the class with closest centre, i.e. $\pi^l(p) = \arg\min_k d_T(p, c_k^{l-1})$
  **foreach** *class centre* $c_k$ **do** */* update of class centres */*
    $c_k^l = \text{med}_T\{p_i \mid \pi^l(p_i) = k\}$ with $\text{med}_T$ the topographical median function
**until** $\pi^l = \pi^{l-1}$ */* stability partition as convergence criterion */*

As already stated, updating class centres can also be done using morphological methods. We propose to define the centre $c_k$ of class $C_k$ as the topographical median $\text{med}_T$ of $C_k$. More precisely, we follow some definitions of multidimensional medians given in the literature (Small, 1990), and consider the convex hull peeling paradigm to determine the inner most element among data (or cluster elements). Adapted to the topographical context, the topographical median $\text{med}_T$ of a class $C_k$ is identified as the furthest pixel from the cluster borders. It can be obtained using the maximum of the distance transform between pixels of $C_k$ to the background defined by $\mathcal{E} \setminus C_k$ (the background may be cleverly limited to exterior borders of $C_k$). This centre is defined by

$$c_k = \arg\max_{p \in C_k} d(p, \mathcal{E} \setminus C_k) \tag{16}$$

with

$$d(p,X) = \min\{d(p,x) \mid x \in X\} \tag{17}$$

the distance of $p$ to the set $X$. The distance $d(p,x)$ can be the Euclidean distance $d_E$ or preferably the topographical distance $d_T$, thus ensuring a better stability to the algorithm since it is the same distance measure which is used in the assignment step. Coherence between these two steps reinforces algorithm convergence through the optimisation of the quality criterion given in Eq. (6) and considering topographical distances. However this convergence cannot be completely ensured since the proposed centre computation scheme can result in outliers (i.e. the class centre maximizes the distance to other classes but may not minimize the distance within its class) and subsequently in oscillations within the iterative process. Thus,

in order to guarantee the algorithm convergence, another criterion may be involved: maximal number of iterations, minimal displacement of class centres between two iterations, convergence of the global cost function, etc. Let us remark that we do not observe any lack of convergence with results given in Sec. 5. Nevertheless, this problem of convergence will be the topic of further studies.

A naive implementation of the proposed algorithm consists in seeking for the shortest path in a graph to measure the distances $d_T(p, c_k)$ between pixels and class centres. This requires $O(P^2)$ operations per pixel, or $O(P)$ if optimal data structures are used, with $P$ the number of pixels. Considering that this computation has to be performed for all pixels and all classes, this results in a global complexity of $O(KP^3)$ (or $O(KP^2)$ with appropriate data structures) per iteration, with $K$ the number of classes.

Since our method can be assimilated to an iterative morphological segmentation, using efficient morphological algorithms based on hierarchical queues helps to limit the algorithm computational complexity. Indeed this complexity can be independent of the number of classes $K$, roughly $O(P)$ by iteration. More precisely, during the assignement step, we do not compute exhaustively the $K$ distance transforms since it is sufficient to know in each pixel which is the closest centre. By propagating distances from class centres, we can process and classify iteratively the pixels of increasing distances inserted in the queue (where only the first occurence of each pixel is considered). Updating class centres can be obtained using the same principle and requires to apply $K$ incomplete distance transforms (each one being limited to its related class), which is equivalent to process only once each pixel. Complexities of these two steps are thus linear in $O(P)$. Let us also observe that since the final distances between pixels and classes centres are only sums of local topographical distances between neighbouring pixels, these local distances may be precomputed in order to further limit computation time (this requires a buffer of size 4 or 8 times the image size, depending on the adjacency in use, i.e. 4- or 8-adjacency). To compare, we recall that *k*means algorithm has a cost equal to $O(KP)$ per iteration, $K$ being possibly high (but still lower than $P$) since it corresponds here to the number of objects present in the image and not anymore to the number of classes.

Algorithm 3 describes the proposed efficient implementation of the topographical *k*means algorithm for image segmentation. It relies on the $\texttt{TOPO}(f, \{s_k, S_k\})$ function defined in Algorithm 4 which returns both a classification map $\pi$ and a distance transform function $\phi$ from a set of initial points $s_k$ which classes $S_k$ are assumed to be known. As we have already noticed, this function is used twice in order to compute both distances to the class centres in the assignment step and distances to the class borders in the updating step. Thus we distinguish both results and note them respectively $(\pi, \phi)$ and $(\bar{\pi}, \bar{\phi})$. As for the `TOPO` function, it assumes the following operations being available to manipulate the hierarchical queue: `NOTEMPTY` checks if the queue contains at least one element, $\texttt{INSERT}(a, b, c)$ insert the element $a$ of class $b$ at priority $c$, and `REMOVE` returns and removes the current triplet $(a, b, c)$ from the queue. A hierarchical queue may be implemented as an array of FIFO queues (each queue being indexed by its priority), thus this last operation seeks for

**Algorithm 3.** Efficient proposed *k*means algorithm for image segmentation

**Input**: Image $f : \mathcal{E} \to \mathcal{T} : p \mapsto f(p)$
**Input**: Number $K$ of class (i.e. regions)
**Output**: Set of classes $\{C\}_K$ or classification (i.e. segmentation) map
$\pi : \mathcal{E} \to \mathcal{C} : p \mapsto \pi(p)$

*/* initialisation of classification map and class centres */*
**foreach** *pixel* $p$ **do** $\pi^0(p) \leftarrow \emptyset$
**foreach** *class centre* $c_k$ **do** $c_k^0 \leftarrow$ `RANDOM` $(\mathcal{E})$
*/* iterative assignment-update process */*
**repeat**
  $l \leftarrow l+1$
  */* pixel-to-class assignement */*
  $(\pi^l, \phi^l) \leftarrow$ `TOPO` $(f, \{c_k^{l-1}, \pi(c_k^{l-1})\})$
  */* update of class centres */*
  $\Omega^l \leftarrow \{p_i \mid \exists q \in \mathcal{N}(p_i),\ \pi^l(q) \neq \pi^l(p_i)\}$ */* identify border pixels */*
  $(\bar{\pi}^l, \bar{\phi}^l) \leftarrow$ `TOPO` $(f, \{\Omega^l, \pi^l(\Omega^l)\})$ */* distance transform from borders */*
  **foreach** *class* $C_k$ **do**
    $c_k^l = \max\{\bar{\phi}^l(p_i) \mid \bar{\pi}^l(p_i) = k\}$ */* use the furthest points from the borders */*
**until** $\pi^l = \pi^{l-1}$ */* stability partition as convergence criterion */*

**Function** `TOPO` $(f, \{s_k, S_k\})$.

**Input**: Image $f : \mathcal{E} \to \mathcal{T} : p \mapsto f(p)$
**Input**: Set $\{s_k, S_k\}$ of initial pixels $s_k$ with related classes $S_k$
**Output**: Classification (i.e. segmentation) map $\pi : \mathcal{E} \to \mathcal{C} : p \mapsto \pi(p)$
**Output**: Topographic distance transform $\phi : \mathcal{E} \to \mathbb{R} : p \mapsto \phi(p)$

*/* initialisation of* $\pi$, $\phi$, *and the hierarchical queue */*
**foreach** $p \in \mathcal{E}$ **do**
  $\pi(p) \leftarrow \emptyset$
  $\phi(p) \leftarrow 0$
**foreach** $s_k$ **do** `INSERT` $(s_k, S_k, 0)$
*/* scan all pixels with increasing* $d_T$ **/*
**while** `NOTEMPTY` **do**
  $(p, C_p, d_p) \leftarrow$ `REMOVE`
  **if** $\pi(p) \neq \emptyset$ **then continue** */* ignore already labeled pixels */*
  $\pi(p) \leftarrow C_p$ */* assign the pixel to the class with closest centre */*
  $\phi(p) \leftarrow d_p$ */* set the distance between the pixel and the closest class centre */*
  **foreach** $q \in \mathcal{N}(p),\ \pi(q) = \emptyset$ **do** */* process all unlabeled neighbouring pixels */*
    $d_q \leftarrow d_p + d_T(p,q)$ */* propagate topographical distance */*
    `INSERT` $(q, C_p, d_q)$ */* add the pixel to the queue */*
**return** $\pi, \phi$

the non-empty queue of lowest priority and returns and removes the oldest element within this queue.

## 5 Experiments

In order to underline the potential interest of our proposal, we have performed some experiments both on artificial and natural images. Fig. 3 shows the behaviour of our algorithm compared to a standard *k*means applied on grey levels, facing an increasing level of noise. We consider 3 classes (i.e. regions) in our algorithm while 2 classes (expected to be black and white) are used for the standard *k*means. As we can observe in this figure, noise robustness achieved by the proposed algorithm seems satisfactory, contrary to the original *k*means method.

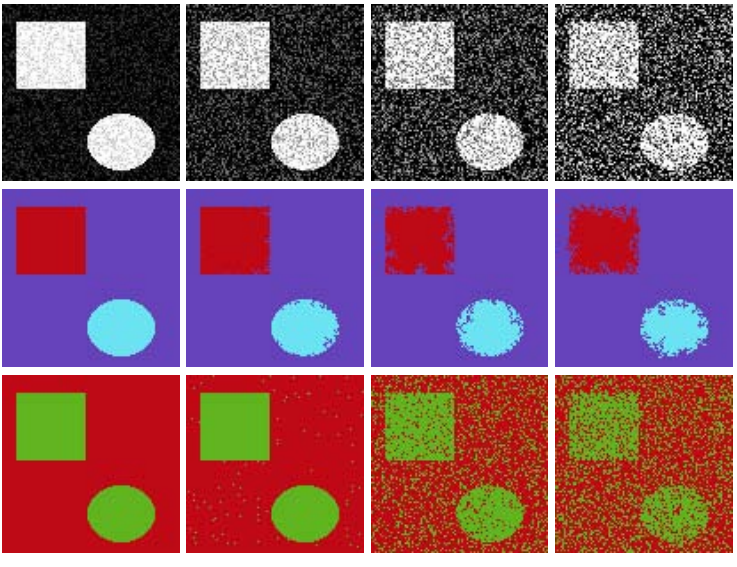


**Fig. 3** Classification results on an artificial image with an increasing level of Gaussian noise (from left to right): noisy input image (top), proposed classification with 3 regions (middle), *k*means performed on intensities with 2 classes (bottom)

The iterative process of the *k*means algorithm is illustrated in Fig. 4. Even if the centres are randomly located on the image, the proposed algorithm manages to correctly identify the objects present in the image as clusters in the classification paradigm. Borders between classes (i.e. regions) are displayed in white while the colours represent distances from the closest class centre, using the same colour scale as in Fig. 2, i.e. increasing from magenta to red.

Similarly to the standard use of the *k*means procedure in unsupervised classification, the number $k$ of classes should be chosen very carefully. As observed in

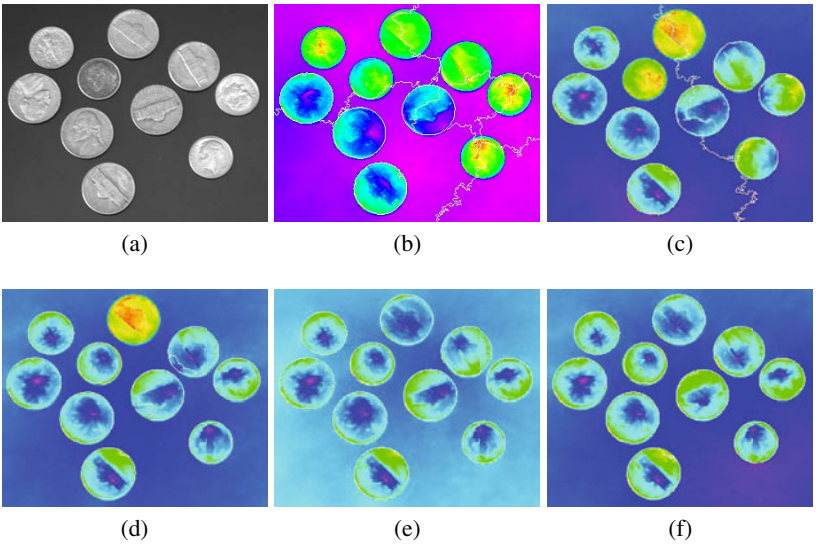



**Fig. 4** Illustration of the iterative topographical *k*means process: (a) input image, topographical distances from centres $\phi^l$ and resulting classification borders $\Omega^l$ after iteration $l = 1$ (b), $l = 2$ (c), $l = 3$ (d), $l = 4$ (e), and $l = 5$ (f) where convergence has been observed

Fig. 5, an appropriate number of classes (i.e. equal to the number of objects to be segmented) will help the algorithm to produce the expected result. On the contrary a too low or too high value of $k$ may result in undersegmentation and oversegmentation, respectively.

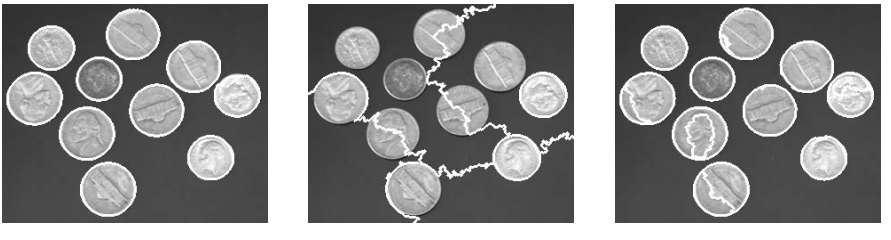


**Fig. 5** Influence of the number of classes: (left) correct segmentation with 11 classes, (centre) undersegmentation with 6 classes, (right) oversegmentation with 16 classes

Finally, let us reuse the introductory image of Fig. 1. Here we have compared our algorithm with a standard *k*means applied on the pixel colours (represented as RGB-valued vectors) to produce 6 colour clusters, followed by a connected component labeling to identify all connected groups of pixels sharing the same colour class. The 1800 regions produced by this approach are shown in Fig. 6 (centre). On the

**Fig. 6** Relevance of the proposed classification scheme: colour input image (left), spectral k-means followed by connected component labeling (centre), topographical k-means (right)

contrary, we can define the expected number of regions as classes in the *k*means algorithm with our approach. Thus we have segmented the marbles using 60 classes (i.e. 60 regions), as illustrated by Fig. 6 (right). The result, despite not being perfect, is rather satisfactory and shows the relevance of our proposal.

## 6 Conclusion

Classification is one of the most common solutions to solve the problem of image segmentation, i.e. transforming an image from the space of pixels to a set of objects which can be subsequently analysed in order to understand the image content. However, classification methods are often applied on the pixel value space, thus ignoring spatial information despite its primordial importance in the context of segmentation. In order to solve this problem and consider the spatial context of each pixel within a classification procedure, the numerous solutions proposed in the literature share the need for additional steps in the classification-based image segmentation process.

In this chapter, we propose a completely different approach which consists in applying the classification in the spatial domain (of the pixels) rather than in the spectral domain (of the pixel values). Thus we can keep the classification method as is (with some adaptations related to the spatial application of the method) and avoid to rely on additional ad-hoc steps. We illustrate our proposal with the spatialisation of the *k*means algorithm, by replacing the Euclidean distance by a topographical distance and by modifying the class centre computation scheme. Besides, a link with the field of mathematical morphology provides efficient algorithms for segmentation, with a complexity in $O(P)$ (i.e. linear and not depending on the number of classes $k$).

This new approach for unsupervised classification in image segmentation brings several perspectives. Among the major future works, we can mention the need to overcome the problems related to the *k*means algorithm (e.g. choice of initial class centres, prior knowledge of the number of classes). Another research direction is to consider a fuzzy paradigm since this has already shown some interest over hard classification (see for instance the Fuzzy C-Means algorithm), and to perform a

fuzzy assignment of pixels to classes (Philipp-Foliguet, 2000). Moreover, it could be relevant to involve spatial constraints within the classification (Han *et al.*, 2001) which in the context of segmentation may be adequate to deal with high gradient areas observed on region borders. A study on robustness and efficiency of the algorithm and experiments on larger datasets are worth being made to understand the pros and cons of our proposal. In particular, we will further study the problem of algorithm convergence. Finally, we also consider to compare our proposal with other recent methods for classification and segmentation, such as mean shift (Comaniciu and Meer, 2002), path-based clustering (Fischer and Buhmann, 2003) or its extension called robust path-based spectral clustering (Chang and Yeung, 2008), since we believe our contribution is not limited to the *k*means algorithm.

**Acknowledgements.** The author wishes to thank his colleague Dr. Alexandre Blansché from LSIIT – CNRS / University of Strasbourg for pointing out the possible lack of convergence of the proposed algorithm, which will be the topic of further studies.

## References

Chang, H., Yeung, D.: Robust path-based spectral clustering. Pattern Recognition 41, 191–203 (2008)

Chen, S., Zhang, D.: Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure. IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics 34(4), 1907–1916 (2004)

Chen, T., Lu, Y.: Color image segmentation – an innovative approach. Pattern Recognition 35(2), 395–405 (2002)

Cheng, H., Sun, Y.: A hierarchical approach to color image segmentation using homogeneity. IEEE Transactions on Image Processing 9(12), 2071–2082 (2000)

Comaniciu, D., Meer, P.: Mean shift: a robust approach toward feature space analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 603–619 (2002)

Diday, E.: Une nouvelle méthode en classification automatique et reconnaissance des formes: la méthode des nuées dynamiques. Revue de Statistique Appliquée 19(2), 19–33 (1971)

Eum, K., Lee, J., Venetsanopoulos, A.: Color image segmentation using a possibilistic approach. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 1150–1155 (1996)

Fischer, B., Buhmann, J.: Path-based clustering for grouping of smooth curves and texture segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(4), 513–518 (2003)

Han, J., Kamber, M., Tung, A.: Spatial clustering methods in data mining: a survey. In: Miller, H., Han, J. (eds.) Geographic Data Mining and Knowledge Discovery. Taylor & Francis, Abington (2001)

Haralick, R., Shapiro, L.: Image segmentation techniques. Computer Vision, Graphics and Image Processing 29, 100–132 (1985)

Ilea, D., Whelan, P.: CTex – An Adaptive Unsupervised Segmentation Algorithm Based on Color-Texture Coherence. IEEE Transactions on Image Processing 17(10), 1926–1939 (2008)

Kaufman, L., Rousseeuw, P.: Finding groups in data: an introduction to cluster analysis. Wiley, New York (1990)

Krishnapuram, R., Freg, C.: Fitting an unknown number of lines and planes to image data through compatible cluster merging. Pattern Recognition (25), 385–400 (1992)

Leydier, Y., Bourgeois, F.L., Emptoz, H.: Sérialisation du k-means pour la segmentation des images en couleur: Application aux images de documents et autres. In: Colloque International Francophone sur l'Ecrit et le Document (CIFED) (2004)

Lezoray, O., Cardot, H.: Cooperation of color pixel classification schemes and color watershed: a study for microscopical images. IEEE Transactions on Image Processing 11(7), 783–789 (2002)

Liew, A., Leung, S., Lau, W.: Fuzzy image clustering incorporating spatial continuity. IEE Proceedings on Vision, Image and Signal Processing 147(2), 185–192 (2000)

Luo, M., Ma, Y., Zhang, H.: A spatial constrained k-means approach to image segmentation. In: Pacific Rim Conference on Multimedia, pp. 738–742 (2003)

Meyer, F.: Topographic distance and watershed lines. Signal Processing 38, 113–125 (1994)

Noordam, J., van den Broek, W., Buydens, L.: Geometrically guided fuzzy c-means clustering for multivariate image segmentation. In: IAPR International Conference on Pattern Recognition (ICPR), pp. 462–465 (2000)

Pappas, T.: An adaptive clustering algorithm for image segmentation. IEEE Transactions on Signal Processing 40, 901–914 (1992)

Pham, T.: Image segmentation using probabilistic fuzzy c-means clustering. In: IEEE International Conference on Image Processing (ICIP), pp. 722–725 (2001)

Philipp-Foliguet, S.: Segmentation d'images en régions floues. In: Rencontres Francophones sur la Logique Floue et ses Applications, La Rochelle, France, pp. 186–196 (2000)

Prêteux, F.: Watershed and skeleton by influence zones: a distance-based approach. Journal of Mathematical Imaging and Vision 1, 239–255 (1992)

Small, C.: A survey of multidimensional medians. International Statistical Review / Revue Internationale de Statistique 58(3), 263–277 (1990)

Soille, P.: Morphological Image Analysis: Principles and Applications. Springer, Berlin (2003)

Tolias, Y., Panas, S.: Image segmentation by a fuzzy clustering algorithm using adaptive spatially constrained membership functions. IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans 28(3), 359–369 (1998)

Turi, R.: Clustering-Based Colour Image Segmentation. PhD dissertation, Monash University. 373 pages (2001)

Xia, Y., Feng, D., Wang, T., Zhao, R., Zhang, Y.: Image segmentation by clustering of spatial patterns. Pattern Recognition Letters 28, 1548–1555 (2007)

# Cluster-Dependent Feature Selection through a Weighted Learning Paradigm

Nistor Grozavu, Younès Bennani, and Mustapha Lebbah

**Abstract.** This paper addresses the problem of selecting a subset of the most relevant features from a dataset through a weighted learning paradigm. We propose two automated feature selection algorithms for unlabeled data. In contrast to supervised learning, the problem of automated feature selection and feature weighting in the context of unsupervised learning is challenging, because label information is not available or not used to guide the feature selection. These algorithms involve both the introduction of unsupervised local feature weights, identifying certain relevant features of the data, and the suppression of the irrelevant features using unsupervised selection. The algorithms described in this paper provide topographic clustering, each cluster being associated to a prototype and a weight vector, reflecting the relevance of the feature. The proposed methods require simple computational techniques and are based on the self-organizing map (SOM) model. Empirical results based on both synthetic and real datasets from the UCI repository, are given and discussed.

**Keywords:** Topographic Clustering, Self-organizing Map, Unsupervised Features Selection, Cluster Characterization, Weighted Learning.

## 1 Introduction

Data mining, or knowledge discovery in databases (KDD), an evolving area in information technology, has received much interest in recent studies. The aim of data mining is to extract knowledge from data (Wang and Huang, 2009). The data size can be measured in two dimensions, the size of features and the size of observations. Both dimensions can take very high values, which can cause problems during the exploration and analysis of the dataset. Models and tools are therefore required to

Nistor Grozavu · Younès Bennani · Mustapha Lebbah
LIPN-UMR 7030, Université Paris 13, 99, av. J-B Clément, 93430 Villetaneuse, France
e-mail: Nistor.Grozavu@lipn.univ-paris13.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 133–147.
springerlink.com 

process data for an improved understanding. Indeed, datasets with a large dimension (size of features) display small differences between the most similar and the least similar data. In such cases it is thus very difficult for a learning algorithm to detect similarity variables that define the clusters. This is the so-called “curse of dimensionality".

Feature selection is commonly used in machine learning, wherein a subset of the features available from the data are selected for application of a learning algorithm. The best subset contains the features that give the highest accuracy score. This is an important stage of preprocessing and is one of two ways of avoiding the curse of dimensionality. The main objectives of dimensionality reduction are thus:

- to facilitate the visualization and data understanding;
- to reduce the required storage space;
- to reduce the learning time;
- to identify the relevant features.

The number of observations can be reduced through unsupervised learning and feature selection. The importance of each feature depends on the size of the learning dataset - for a small sample size, eliminating a relevant feature can reduce the error. Note also that irrelevant features can be very informative when used together. Several methods can be used to reduce the size of features.

- Selection: a subset of features is chosen from the initial data space;
- Transformation: new features are built in a transformed space - an output space.

We aimed to reduce the described space size using unsupervised learning by selection through feature weighting. To select for relevant features, we combined feature weighting with feature selection. In feature selection, the task is reduced to simply eliminating the features that are completely irrelevant. Feature selection is commonly used in supervised learning (Yacoub and Bennani, 2000; Bennani., 1999; Fukunaga, 1990; Almuallim and Dietterich, 1991). This method is based on maximizing certain functions of predictive accuracy.

Feature weighting is an extension of the selection process whereby the features are assigned continuous weights, which can be regarded as degrees of relevance. Continuous weighting provides more information about the relevance of various features. Clustering and feature weighting are thus clearly linked. Applying these tasks in sequence can reduce the performance of the learning system. Therefore, a new algorithm for clustering and for feature weighting is needed. Feature weighting for unsupervised learning has received interest only recently; this paper may therefore serve as a useful guideline to future researchers.

We have focused on models that are based on both dimensionality reduction and clustering, using self-organizing maps (SOM Kohonen, 2001) in order to characterize clusters. SOM models are often used for visualization and unsupervised topological clustering, this technique allowing projection in low dimensional spaces that are generally two dimensional. A number of previous studies have already described extensions and reformulations of the SOM model (Bishop *et al.*, 1998; Lebbah *et al.*, 2007; Verbeek *et al.*, 2005).

Several important research topics in cluster analysis and feature weighting have been previously discussed (Frigui and Nasraoui, 2004; Wang *et al.*, 2008; Tsai and Chiu, 2008; Huang *et al.*, 2005; Blansche *et al.*, 2006; Dy and Brodley, 2004). In particular Dy et al. propose a probabilistic model for feature selection in unsupervised learning using an expectation-maximization (EM) method (Dy and Brodley, 2004). Frigui and Nasraoui (2004) proposed two local weighting unsupervised clustering algorithms based on fuzzy c-means algorithms (SCAD1 and SCAD2) to categorize the unlabeled data and determine the best feature weights within each cluster. Two further studies (Wang *et al.*, 2008; Huang *et al.*, 2005) describe an approach that minimizes the same objective function as used by Frigui et al., but which additionally estimates the global feature weighting. The proposed mechanism for feature weighting has also been extended for a fuzzy *k*-means algorithm (Li and Yu, 2006) and subspace clustering (Jing *et al.*, 2007). Similar techniques, based on *k*-means and weighting have been developed by other researchers (Tsai and Chiu, 2008; Huh and Lim, 2009).

In contrast to the global weighting approach based on the SOM method, which considers a single weight vector for the map (Guérif and Bennani, 2007; Benabdeslem and Lebbah, 2007), our local weighting algorithms characterize each cell of the map by a prototype and weight vector, with each component reflecting the corresponding feature relevance. These weight vectors are thus used for local feature selection, characterizing clusters with the best subset of features. For the feature selection task, we used a method inspired from Cattell (1966)'s scree test, which was initially developed for the selection of principal components.

In the following sections, we introduce the classical self-organizing maps (SOM) (section 2) and then discuss both the *lwo*-SOM (local weighting observation) and *lwd*-SOM (local weighting distance) methods (section 3). We present the feature selection algorithm and the principle of Cattell's algorithm in section 4. In section 5, we show experimental results obtained for several datasets. These datasets allow us to demonstrate the use of this algorithm for topological clustering and feature weighting. Some conclusions are discussed at the end of the paper, as are and future perspectives for research in this area.

## 2 Classical Self-organizing Map (SOM)

Self-organizing maps are increasingly used as tools for the visualization of data, as they allow projection in low, typically bi-dimensional spaces. The basic model proposed by Kohonen consists of a discrete set $\mathcal{C}$ of cells called "map". This map has a specific topology defined by an undirected graph, which is usually a regular, two-dimensional grid. For each pair of cells $(j,k)$ on the map, the distance $\delta(j,k)$ is defined as the length of the shortest chain linking cells $j$ and $k$ on the grid. For each cell $j$ this distance defines a neighboring cell; a kernel positive function $\mathcal{K}$ ($\mathcal{K} \geq 0$ and $\lim_{|y|\to\infty} \mathcal{K}(y) = 0$) is introduced to determine the neighboring area.

We define the mutual influence of two cells $j$ and $k$ by $\mathcal{K}_{j,k}$. In practice, as for classical topological maps, we use a smooth function to determine the size of the neighboring area: $\mathcal{K}_{j,k} = \exp(\frac{-\delta(j,k)}{T})$. Using this kernel function, $T$ becomes a parameter of the model. As in the Kohonen algorithm, we decrease $T$ from an initial value $T_{max}$ to a final value $T_{min}$.

Let $\Re^d$ be the Euclidean data space and $E = \{\mathbf{x}_i; i = 1,\ldots,N\}$ a set of observations, where each observation $\mathbf{x}_i = (x_i^1, x_i^2, ..., x_i^d)$ is a vector in $\Re^d$. For each cell $j$ of the grid (map), we associate a referent vector (prototype) $\mathbf{w}_i = (w_i^1, w_i^2, ..., w_i^d)$ which characterizes one cluster associated to cell $i$. The set of referent vectors is denoted by $\mathcal{W} = \{\mathbf{w}_j, \mathbf{w}_j \in \Re^d\}_{j=1}^{|\mathcal{W}|}$. The set of parameter $\mathcal{W}$ has to be estimated iteratively by minimizing the classical objective function defined as follows:

$$R(\chi, \mathcal{W}) = \sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{W}|} \mathcal{K}_{j,\chi(\mathbf{x}_i)} \|\mathbf{x}_i - \mathbf{w}_j\|^2 \tag{1}$$

where $\chi$ assigns each observation $\mathbf{x}_i$ to a single cell in the map $\mathcal{C}$. This cost function can be minimized using both stochastic and batch techniques (Kohonen, 2001).

## 3 Local Weighting Learning Paradigm and SOM

One disadvantage of the classical SOM algorithms is that they treat all features equally. This is not desirable for many applications of clustering, in which observations are defined by a large number of features. A cluster provided by SOM is often characterized by only a subset of features rather than by the entire features set. The presence of other features may therefore prevent the discovery of the specific cluster structure associated to each cell. The relevance of each feature changes from one cluster to another. Thus, the question remains *how to compute feature relevance (weights) automatically during SOM learning process.* Feature weighting is an extension of the feature selection procedure, whereby features are assigned continuous weights, which can be considered as degrees of relevance (Blansche *et al.*, 2006).

The proposed approach for SOM clustering and feature weighting aims to select both the optimal prototypes and feature weights at the same time. Each prototype $\mathbf{w}_j = (w_j^1, w_j^2, ..., w_j^d)$ corresponding to cell $j$ is allowed to have its own set of local features weights $\pi_j = (\pi_j^1, \pi_j^2, ..., \pi_j^d)$. We denote the set of weight vectors ($|\Pi| = |W|$) by $\Pi = \{\pi_j, \pi_j \in \Re^d\}_{j=1}^{|\Pi|}$.

In the following section, we present two versions of local feature weighting using SOM: observation weighting and distance weighting.

### *3.1 Local Weighting Observations : lwo-SOM*

We based our method on initial work describing the supervised model $w$-LVQ2 (Yacoub and Bennani, 2000). This approach adapts weights to filter the observation

during the learning process. Using this model, we weighted observations $\mathbf{x}$ using weight vectors $\pi$ before computing the distance. The objective function was rewritten as follows:

$$R_{lwo}(\chi, \mathcal{W}, \Pi) = \sum_{i=1}^{|E|} \sum_{j=1}^{|\mathcal{W}|} \mathcal{K}_{j,\chi(\mathbf{x}_i)} \|\pi_j \mathbf{x}_i - \mathbf{w}_j\|^2 \tag{2}$$

Minimization of $R_{lwo}(\chi, \mathcal{W}, \Pi)$ was performed by iterative repetition of the following three steps until stabilization. The initialization step determines the prototype set $\mathcal{W}$ and the set of associated weights $\Pi$, at each training step $(t+1)$. An observation $\mathbf{x}_i$ is then randomly chosen from the input dataset and the following operations are repeated:

- Minimize $R_{lwo}(\chi, \hat{\mathcal{W}}, \hat{\Pi})$ with respect to $\chi$ by fixing $\mathcal{W}$ and $\Pi$. Each weighted observation $(\pi_j \mathbf{x}_i)$ is assigned to the closest prototype $\mathbf{w}_j$ using the assignment function, defined as follows:

$$\chi(\mathbf{x}_i) = \arg\min_j \left( \|\pi_j \mathbf{x}_i - \mathbf{w}_j\|^2 \right)$$

- Minimize $R_{lwo}(\hat{\chi}, \mathcal{W}, \hat{\Pi})$ with respect to $\mathcal{W}$ by fixing $\chi$ and $\Pi$. The prototype vectors are updated using the gradient stochastic expression:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \varepsilon(t) \mathcal{K}_{j,\chi(\mathbf{x}_i)} \left( \pi_j \mathbf{x}_i - \mathbf{w}_j(t) \right)$$

- Minimize $R_{lwo}(\hat{\chi}, \hat{\mathcal{W}}, \Pi)$ with respect to $\Pi$ by fixing $\chi$ and $\mathcal{W}$. The update rule for the feature weight vector $\pi_j(t+1)$ is:

$$\pi_j(t+1) = \pi_j(t) + \varepsilon(t) \mathcal{K}_{j,\chi(\mathbf{x}_i)} \mathbf{x}_i \left( \pi_j(t) \mathbf{x}_i - \mathbf{w}_j(t) \right)$$

As in the traditional stochastic learning algorithm of Kohonen, we denote the learning rate at time $t$ by $\varepsilon(t)$. The training is usually performed in two phases. In the first phase, a high initial learning rate $\varepsilon(0)$ and a large neighborhood radius $T_{max}$ are used . In the second phase, a low learning rate and small neighborhood radius are used from the beginning.

### 3.2 *Local Weighting Distance: lwd-SOM*

Unlike *lwo*-SOM, the local distance weighting involves weighting the distance between observations and prototypes. We propose to minimize the following objective function:

$$R_{lwd}(\chi, \mathcal{W}, \Pi) = \sum_{i=1}^{N} \sum_{j=1}^{|\mathcal{W}|} \mathcal{K}_{j,\chi(\mathbf{x}_i)} (\pi_j)^{\beta} \|\mathbf{x}_i - \mathbf{w}_j\|^2 \tag{3}$$

where $\beta$ is the discrimination coefficient.

The *lwd*-SOM cost function is minimized in three steps:

1. Minimize $R_{lwd}(\chi, \hat{\mathcal{W}}, \hat{\Pi})$ with respect to $\chi$ by fixing $\mathcal{W}$ and $\Pi$. The equation is defined as follows:

$$\chi(\mathbf{x}_i) = \arg\min_j \left( (\pi_j)^\beta \|\mathbf{x}_i - \mathbf{w}_j\|^2 \right)$$

2. Minimize $R_{lwd}(\hat{\chi}, \mathcal{W}, \hat{\Pi})$ with respect to $\mathcal{W}$ by fixing $\chi$ and $\Pi$. The prototype's vectors are updated using the following equation:

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \varepsilon(t)\mathcal{K}_{j,\chi(\mathbf{x}_i)}(\pi_j)^\beta (\mathbf{x}_i - \mathbf{w}_j(t))$$

3. Minimize $R_{lwd}(\hat{\chi}, \hat{\mathcal{W}}, \Pi)$ with respect to $\Pi$ by fixing $\chi$ and $\mathcal{W}$. A weighting vector $\pi_j(t+1)$ is updated according to the following equation:

$$\pi_j(t+1) = \pi_j(t) + \varepsilon(t)\mathcal{K}_{j,\chi(\mathbf{x}_i)}\beta(\pi_j(t))^{\beta-1}\|\mathbf{x}_i - \mathbf{w}_j(t)\|^2$$

In addition, the parameter $\beta$ needs to be provisionally fixed. As with the *lwo*-SOM algorithm, we start with a large initial value for the learning radius which decreases during the learning process allowing quantization of the prototypes. At the end of the learning phase, the *lwd*-SOM model represents a *k*-means model with simultaneously weighted features (SCAD, Frigui and Nasraoui 2004, or *w-k*-means, Huang *et al.* 2005).

## 4 Automatic Characterization of Clusters through Feature Selection

Feature selection for clustering or unsupervised feature selection is used to identify the feature subsets that accurately describe the clusters. This improves the interpretability of the induced model, as only relevant features are involved in it, without degrading its descriptive accuracy. Additionally, the identification of relevant and irrelevant features with SOM learning provides valuable insight into the nature of the cluster-structure.

Feature selection for clustering analysis is difficult because, unlike supervised learning, there are no class labels for the dataset and no obvious criteria to guide the search Wiratunga *et al.* (2006). Feature selection in clustering must provide features that describe the "best" homogeneous cluster. Here, we used the weight set $\Pi$ and prototype set $\mathcal{W}$ provided by *lwo*-SOM and *lwd*-SOM. We then clustered the map and used selection to characterize the resulting clusters associated with cells and group of cells. For map clustering we used traditional hierarchical clustering combined with the Davies-Bouldin index to choose the optimal partition (Vesanto and Alhoniemi, 2000). We then used a *scree-test*-like method to select the most important features. The subjective scree test is a graphical method first proposed by Cattell (1966) for principal components analysis (PCA).

The basic idea of the scree test is to generate a curve associated with eigenvalues, allowing random behavior to be identified (a simple line plot). Cattell suggests to find the place where the smooth decrease of eigenvalues appears to level off to the right of the plot. To the right of this point, presumably, one finds only "factorial scree". Non graphical solutions to the Cattell scree test are also proposed (Raîche *et al.*, 2006): an acceleration factor and the optimal coordinates index. The acceleration factor indicates where the elbow of the scree plot appears. It corresponds to the acceleration of the curve, i.e. the second derivative. Frequently this scree is appearing where the slope of the hill changes drastically to generate the scree. It is why many studies choose the criterion eigenvalue where the slope changes quickly to determine the number of components for a PCA. It is what Cattell named the elbow. So, they look for the place where the positive acceleration of the curve is at his maximum. Cattell's scree test and Bartlett's chi-square test for the number of factors to be retained from a factor analysis are shown to be based on the same rationale, with the former reflecting subject sampling variability, and the latter reflecting variable sampling variability (Horn and Engstrom, 1979). Eigenvalues considered in Cattell's scree method can be interpreted as the degree of relevance of each factor axis. Hence, in our case, we use this method by analogy to choose the variables represented by their relevance vector $\Pi$. The purpose is to detect, the 'scree' where the slope of the relevance graph changes radically which corresponds to the position of the variable from which the pertinence $\pi$ becomes not significant. The number of variables retained is equal to the number of values preceding this 'scree'. We therefore needed to identify the point of maximum deceleration in the curve.

Figure 1 shows an example of a curve generated using a weight vector. We observed the scree on the 19th feature which means that the irrelevant features have

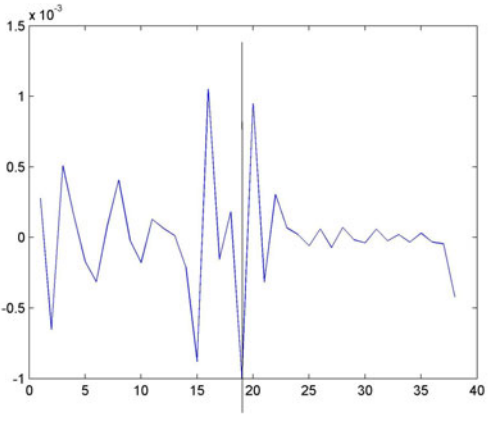



**Fig. 1** An example of the automatic scree method using a particular weight vector. The axes *X* and *Y* correspond to features and weights, respectively. The scree is indicated by the vertical bar.

index values lying in the range $[20-40]$. We used an automated process to apply this technique to each weight vector $\pi_j = (\pi_j^1, \pi_j^2, ..., \pi_j^d)$. We thus executed the following steps:

**Scree Test Acceleration Factor**

1. Sort the weights in descending order $\pi_j$. Thus we obtain a new order $\pi_j = (\pi_{j,1}, \pi_{j,2}, ..., \pi_{j,i}, ..., \pi_{j,d})$; where $i$ indicates the index order.
2. Compute the first difference $df_i = \pi_{j,i} - \pi_{j,i+1}$;
3. Compute the second difference (acceleration) $acc_i = df_i - df_{i+1}$;
4. Find the scree: $\max_i (abs(acc_i) + abs(acc_{i+1}))$;
5. Retain all the features displayed before the scree (we used the initial index values of features before sorting).

## 5 Experimental Protocol

We performed several experiments on five known problems from the UCI Repository of machine learning databases (Asuncion and Newman, 2007).

| Dataset | nb. observations | nb. features | nb. classes |
|---|---|---|---|
| Waveform | 5000 | 40 | 3 |
| WDBC | 569 | 32 | 2 |
| Isolet | 1559 | 617 | 2 |
| Madelon | 2000 | 500 | 32 |
| SpamBase | 4601 | 57 | 2 |

- Waveform dataset: This dataset consists of 5000 observations divided into three classes (Figure 2). The original dataset included 40 features, 19 of which were attributed to noise, with mean 0 and variance 1. Each class was generated from a combination of 2 of 3 "base" waves.
- Wisconsin Diagnostic Breast Cancer (WDBC): This dataset includes 569 observations with 32 features (ID, diagnosis, 30 real-valued input features). Each observation is labeled as benign (357) or malignant (212). Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.
- Isolet dataset: This dataset was generated as follows. 150 subjects spoke the name of each letter of the alphabet twice, giving 52 training examples from each subject. Subjects were grouped into sets of 30 speakers each, referred to as isolet1, isolet2, isolet3, isolet4, and isolet5. The data consisted of 1559 observations and 617 features. All features were continuous, real-valued features within the range -1.0 to 1.0.
- Madelon dataset: MADELON is an artificial dataset, with continuous input features. It formed part of the NIPS 2003 feature selection challenge. This dataset is a two-class classification problem which contains data points grouped into 32 clusters placed on the vertices of a five dimensional hypercube and randomly labeled +1 or -1. The five dimensions constitute five informative features. Fifteen

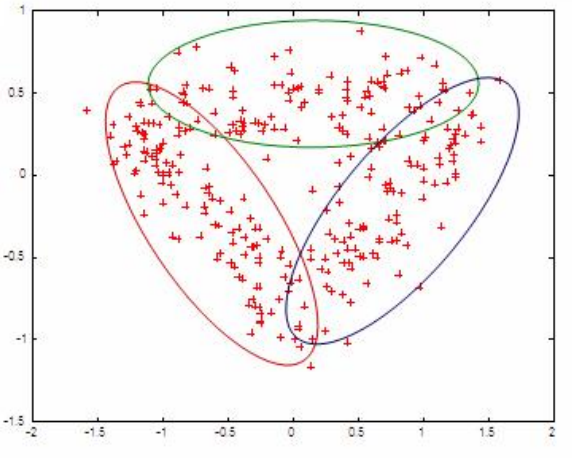



**Fig. 2** Waveform dataset. 3 classes of waves

linear combinations of these features were added to form a set of 20 (redundant) informative features. Based on these 20 features, the examples can be separated into the two classes (corresponding to the +/-1 labels).

- SpamBase dataset: The SpamBase dataset is composed of 4601 observations described by 57 features. Every feature describes an e-mail and its category: spam or not-spam. Most of the attributes indicate whether a particular word or character occurs frequently in the e-mail. The run-length attributes (55-57) measure the length of sequences of consecutive capital letters.

To evaluate the quality of clustering, we compared results to a "ground truth". We used the clustering accuracy for measuring the clustering results. In general, the results of clustering are usually assessed on the basis of some external knowledge about how clusters should be structured. The only way to assess the usefulness of a clustering result is indirect validation, whereby clusters are applied to the solution of a problem and the correctness is evaluated against objective external knowledge. This procedure is defined by (Jain and Dubes, 1988) as "validating clustering by extrinsic classification", and has been used in many other studies. To use this approach we therefore need labeled datasets, where the external (extrinsic) knowledge is the class information provided by labels. Thus, the identification of significant clusters in the data, by *lwo*-SOM or *lwd*-SOM will be reflected by the distribution of classes. A purity score can thus be expressed as the percentage of elements in a cluster that have been assigned a particular class.

We also validated our approaches in supervised case learning paradigms. We used the $K$-fold cross validation technique, repeated $s$ times for $s = 5$ and $K = 3$, to estimate the performance of *lwo*-SOM or *lwd*-SOM. For each run, the dataset was split into three disjoint subsets of equal size (15 runs for each dataset). We used two subsets for training and then tested the model on the remaining subset using

all features and selected features (selected on the cells or on clusters). The labels generated were compared to the real labels of the test set for each run.

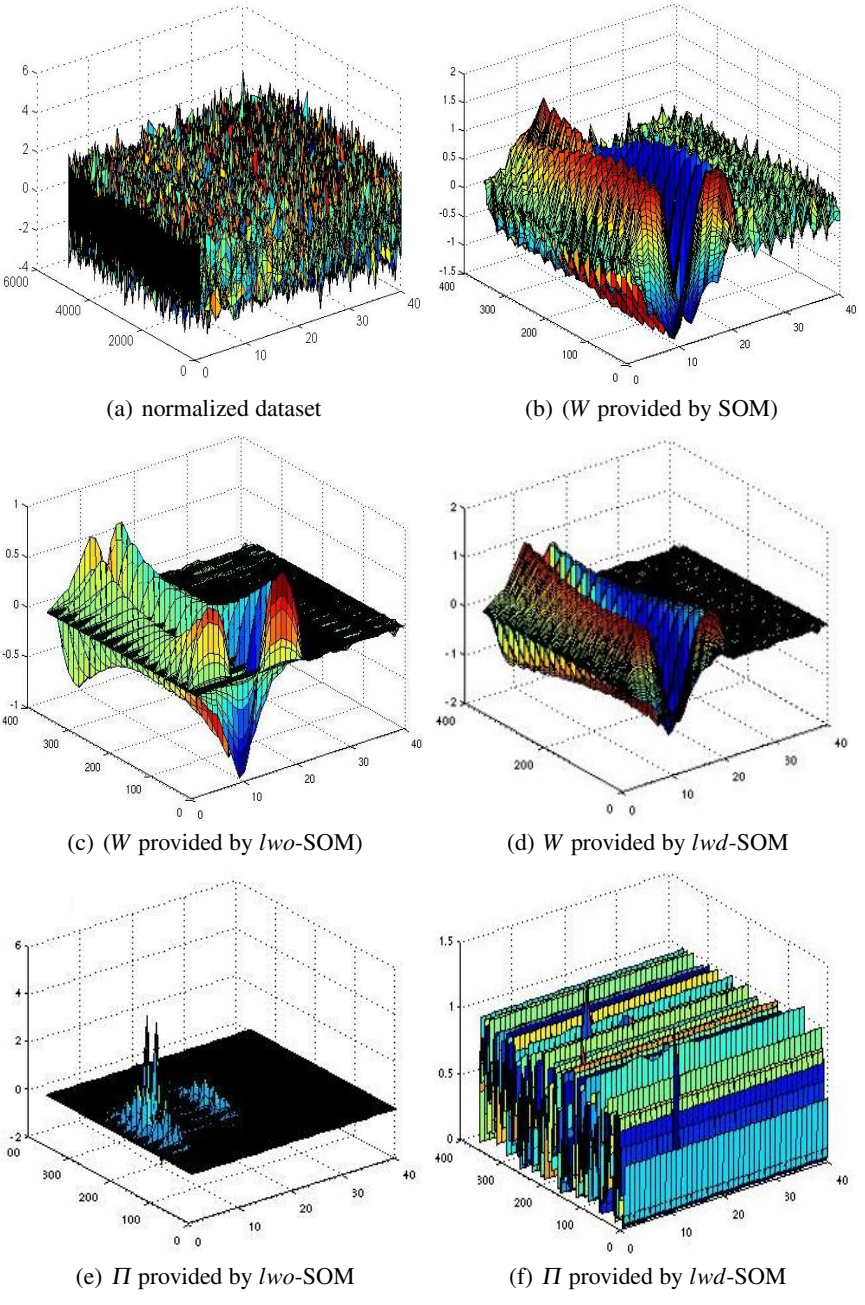
We used the purity index to evaluate the quality of map segmentation. This index shows the correspondence between the class of data and cluster label, which is computed using the majority vote rule . A high value for this measure indicates a high level of homogeneous clustering. A purity index value close to 0 is indicative of poor clustering, whereas an index value close to 1 is indicative of a good clustering result.

### *5.1 Results on Waveform*

We used this dataset to show a good level of performance for both algorithms (*lwd*-SOM and *lwo*-SOM) for simultaneous clustering and feature weighting. All observations were used to generate a map with $26 \times 14$ cells dimension. Both learning algorithms provided two vectors for each cell: the referent vector $\mathbf{w}_j = (w_j^1, w_j^2, ..., w_j^d)$ and weight vector $\pi_j = (\pi_j^1, \pi_j^2, ..., \pi_j^d)$, where $d = 40$. Preparing data for clustering requires some preprocessing, such as normalization or standardization. In the first experimentation step, we normalized the initial dataset to obtain more homogeneous data (Figure 3(a)). We used variance normalization, representing a linear transformation that scales the values such that their variance is equal to 1.

We created 3D representations of the referent vector and weight vector provided by classical SOM and by our methods (*lwd*-SOM and *lwo*-SOM). The axes $X$ and $Y$ indicate the features and the referent indexes, respectively. The amplitude indicates the mean value of each component. Examination of the three graphs $(3(b), 3(c), 3(d))$ shows that the noise represented by features 19 to 40 may be clearly detected with low amplitudes. This visual analysis of the results clearly shows that the new algorithm *lwo*-SOM provides the best results. Both graphs of weights $\Pi$ and prototypes $\mathcal{W}$ show that features associated to noise is irrelevant with low amplitude. Visual analysis of both weight vectors (figure 3(e) and figure 3(f)) showed the weight vectors obtained with *lwo*-SOM to give a more accurate representation of the data structure (features relevance) than the weight vectors obtained with *lwd*-SOM. The *lwo*-SOM algorithm provides good results because the weight vectors work as a filter for observations and estimates the referents that result from this filtering. We applied the selection task to all parameters of the map before and after map clustering to check that it was possible to automatically select the features using our algorithms. This task involves detecting major changes for each input vector represented as a signal graph. We used hierarchical classification (Vesanto and Alhoniemi, 2000) for clustering the map.

After *lwo*-SOM map clustering, we obtained three clusters with a purity index equal to 0.7076. Using *lwd*-SOM resulted in six clusters. However, in *lwd*-SOM map, clustering with the product $\Pi\mathcal{W}$ led to the generation of three clusters (purity index equal to 0.6803), which were significant in our example. This demonstrates that when there is no cluster (labels) information, feature weighting can be used to find and characterize homogeneous clusters.



(a) normalized dataset (b) ($W$ provided by SOM)

(c) ($W$ provided by $lwo$-SOM) (d) $W$ provided by $lwd$-SOM

(e) $\Pi$ provided by $lwo$-SOM (f) $\Pi$ provided by $lwd$-SOM

**Fig. 3** 3D visualization of the referent vector and weight vector. The axes $X$ and $Y$ indicate features and the referent index values, respectively. The amplitude indicates the mean value of each component of map $26 \times 14$ (364 cells).

**Table 1** Comparison of the selected features for each cluster using classical methods and our new methods (*lwo*-SOM, *lwd*-SOM). $[i-j]$ indicates the set of selected features

| Db | # real cluster | SOM $\mathcal{W}$ | *lwd*-SOM $\Pi\mathcal{W}$ | *lwo*-SOM $\mathcal{W}$ |
|---|---|---|---|---|
| wave-form | 3 | $cl_1$: [12-20]<br>$cl_2$: [10-18]<br>$cl_3$: [1-8; 14-20]<br>$cl_4$: [9-13]<br>$cl_5$: [2-10] | $cl_1$: [6-15]<br>$cl_2$: [4-10]<br>$cl_3$: [7-19] | $cl_1$: [3-8; 11-16]<br>$cl_2$: [8-11; 14-19]<br>$cl_3$: [3-20] |
| Purity | | 0,6620 | 0,6803 | 0,7076 |

The characterization of clusters with the "Scree Test" algorithm is provided in Table 1. For each algorithm, we present the features selected for each cluster. Both techniques (*lwo*-SOM, *lwd*-SOM) provided three clusters characterized by different features. By contrast, segmentation of the map using classical SOM provided six clusters with a purity index value of 0.662. Map segmentation was performed using hierarchical clustering with all the features. For clusters $cl_1$, $cl_2$ and $cl_3$, the features selected using *lwd*-SOM were also selected using *lwo*-SOM. We found that both algorithms *lwo*-SOM and *lwd*-SOM identified relevant and informative features, giving more accurate results than classical SOM. The new and classical methods were compared after segmentation of the map. We investigated the effect of selected features before and after, or without segmentation by testing this selection process in the supervised paradigm and computing the accuracy index for each method.

## 6 Results for Other Datasets

We tested our algorithms on additional datasets with different characteristics. To demonstrate the potential benefits of simultaneous clustering and feature weighting, we used the referent and weight vector for map clustering. For both algorithms proposed, we showed the feature selection results obtained for the Isolet, Madelon, WDBC and SpamBase datasets.

Table 2 shows the comparison between the results obtained from the four datasets: we compared the characteristics of all clusters for each dataset and found that our two methods *lwo*-SOM and *lwd*-SOM provided similar results. We recall that *lwo*-SOM and *lwd*-SOM characterize clusters in an automated unsupervised manner. Validation of the results obtained was difficult because these methods used unsupervised learning. Therefore, we compared our methods using supervised validation techniques.

Table 3 provides a comparison of the accuracy of classification for various datasets after running a 3-fold cross-validation five times. We compared different situations in which the features were selected using our methods (*lwo*-SOM, *lwd*-SOM) or classical SOM. We found that our methods performed better than SOM

**Table 2** Comparison of selected features for each cluster using our methods (*lwo*-SOM, *lwd*-SOM). $[i-j]$ indicates the set of selected features

| Db. | # clu-. ster | *lwd*-SOM using $\Pi\mathcal{W}$ | *lwo*-SOM using $\mathcal{W}$ |
|---|---|---|---|
| wdbc | 2 | $cl_1$-$cl_9$:[4;24] | $cl_1$-$cl_9$:[4;24] |
| Made-lon | 2 | $cl_1$:[1]<br>$cl_2$: [91, 281, 403-424] | $cl_1$:[1]<br>$cl_2$: [242, 417-452] |
| Isolet | 26 | $cl_1$-$cl_{13}$:<br>[1-330, 450-617] | $cl_1$-$cl_{13}$:<br>[5-302, 434-488]<br>[545-551, 586-593] |
| SpamB | 2 | $cl_1$: [56]; $cl_2$:[57] | $cl_1$:[56]; $cl_2$: [57] |

**Table 3** Comparison of purity scores with ±SD after running a 3-fold cross-validation five times (15 runs for each). b/a - before and after segmentation; Sel f. - selected features by cell; Sel cl. - selected features by cluster

| Db. | b/a sel/cl | method | | |
|---|---|---|---|---|
| | | SOM | *lwo*-SOM | *lwd*-SOM |
| Waveform | b. | 0.7488±0.0117 | 0.8178±0.0104 | 0.7833±0.0201 |
| | Sel f. | 0.7158±0.0085 | 0.8310±0.0096 | 0.8281±0.0108 |
| | Sel cl. | 0.7479±0.0293 | 0.8289±0.0118 | 0.8279±0.0131 |
| Isolet | b. | 0.7786±0.05 | 0.7975±0.04 | 0.7792±0.047 |
| | Sel f. | 0.7409±0.052 | 0.7863±0.043 | 0.7608±0.041 |
| | Sel cl. | 0.6786±0.061 | 0.7821±0.047 | 0.7796±0.048 |
| wdbc | b. | 0.8941±0.042 | 0.9203±0.037 | 0.9052±0.041 |
| | Sel f. | 0.8923±0.047 | 0.9152±0.04 | 0.9023±0.043 |
| | Sel cl. | 0.891±0.046 | 0.9145±0.041 | 0.9014±0.042 |
| Spam | b. | 0.8958±0.041 | 0.8669±0.041 | 0.8568±0.043 |
| | Sel f. | 0.8579±0.039 | 0.8754±0.04 | 0.8727±0.043 |
| | Sel cl. | 0.6184±0.044 | 0.8564±0.041 | 0.8534±0.042 |
| made-lon | b. | 0.6541±0.041 | 0.6803±0.04 | 0.6752±0.039 |
| | Sel f. | 0.6608±0.038 | 0.7017±0.041 | 0.6914±0.04 |
| | Sel cl. | 0.6524±0.052 | 0.7163±0.042 | 0.7089±0.047 |

for the various situations (using all features, features selected by cell and features selected by cluster). In all cases local weighting observation *lwo*-SOM gave a significantly higher classification accuracy than other algorithms. Means and standard deviation (SD) for the accuracy index values were computed for 15 independent runs. We found that the proposed methods *lwo*-SOM and *lwd*-SOM performed significantly better and were more consistent than the traditional SOM for the various tested cases.

## 7 Conclusions and Future Work

We have described a process for selecting relevant features in unsupervised learning paradigms using two new approaches. These new methods are based on the SOM model and feature weighting. Both learning algorithms *lwo*-SOM and *lwd*-SOM provide cluster characterization by determining the feature weights within each cluster. We described extensive testing using a novel statistical method for unsupervised feature selection. Our approaches demonstrated the efficiency and effectiveness of this method in dealing with high dimensional data for simultaneous clustering and weighting. The models proposed in this paper were tested on a wide variety of datasets (Table 1), showing a better performance for the *lwo*-SOM algorithms than for the *lwd*-SOM or classical SOM algorithm. We also showed that through different means of visualization, *lwo*-SOM and *lwd*-SOM algorithms provide various pieces of information that could be used in practical applications. This paper offers several perspectives for future work. We can extend both models to take into account possible correlations between features and the robustness to noise. We can also, extend the algorithms to treat other types of features (categorical, mixed features) using an appropriate measure or distance.

**Acknowledgements.** We thank the EGC'09 committee for inviting us to participate in this book. This work was supported by Cap Digital under Infom@gic Project.

## References

Almuallim, H., Dietterich, T.: Learning with many irrelevant features. In: Proceedings of the Ninth National Conference on Artificial Intelligence, pp. 547–552. AAAI Press, Anaheim (1991)

Asuncion, A., Newman, D.: UCI Machine Learning Repository (2007), `http://www.ics.uci.edu/~mlearn/{MLR}epository.html`

Benabdeslem, K., Lebbah, M.: Feature selection for Self Organizing Map. In: International Conference on Information Technology Interface-ITI 2007, Cavtat-Dubrovnik,Croatia, June 25-28, pp. 45–50 (2007)

Bennani., Y.: Adaptive weighting of pattern features during learning. In: IJCNN 1999, Piscataway, NJ, vol. 5, pp. 3008–3013 (1999)

Bishop, C.M., Svensén, M., Williams, C.K.I.: GTM: The generative topographic mapping. Neural Comput. 10(1), 215–234 (1998)

Blansche, A., Gancarski, P., Korczak, J.: MACLAW: A modular approach for clustering with local attribute weighting. Pattern Recognition Letters 27(11), 1299–1306 (2006)

Cattell, R.: The scree test for the number of factors. Multivariate Behavioral Research 1, 245–276 (1966)

Dy, J.G., Brodley, C.E.: Feature Selection for Unsupervised Learning. JMLR 5, 845–889 (2004)

Frigui, H., Nasraoui, O.: Unsupervised learning of prototypes and attribute weights. Pattern Recognition 37(3), 567–581 (2004)

Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Computer Science and Scientific Computing Series. Academic Press, London (1990)

Guérif, S., Bennani, Y.: Dimensionality reduction trough unsupervised features selection. In: International Conference on Engineering Applications of Neural Networks (2007)

Horn, J.L., Engstrom, R.: Cattell's Scree Test in Relation to Bartlett's Chi-Square Test and Other Observations on the Number of Factors Problem. Multivariate Behavioral Research 14(3), 283–300 (1979)

Huang, J.Z., Ng, M.K., Rong, H., Li, Z.: Automated Variable Weighting in k-Means Type Clustering. IEEE Trans. Pattern Anal. Mach. Intell. 27(5), 657–668 (2005), http://dx.doi.org/10.1109/TPAMI.2005.95

Huh, M.-H., Lim, Y.B.: Weighting variables in K-means clustering. Journal of Applied Statistics 36(1), 67–78 (2009)

Jain, A.K., Dubes, R.C.: Algorithms for clustering data. Prentice-Hall, Inc., Upper Saddle River (1988)

Jing, L., Ng, M.K., Huang, J.Z.: An Entropy Weighting k-Means Algorithm for Subspace Clustering of High-Dimensional Sparse Data. IEEE Trans. on Knowl. and Data Eng. 19(8), 1026–1041 (2007), http://dx.doi.org/10.1109/TKDE.2007.1048

Kohonen, T.: Self-organizing Maps. Springer, Berlin (2001)

Lebbah, M., Rogovschi, N., Bennani, Y.: BeSOM: Bernoulli on Self Organizing Map. In: IJCNN 2007, Orlando, Florida (2007)

Li, C.-X., Yu, J.: A novel fuzzy C-means clustering algorithm. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) RSKT 2006. LNCS (LNAI), vol. 4062, pp. 510–515. Springer, Heidelberg (2006)

Raîche, G., Riopel, M., Blais, J.-G.: Non Graphical Solutions for the Cattell's Scree Test. In: International Meeting of the Psychometric Society, IMPS 2006, HEC, Montréal (2006)

Tsai, C.-Y., Chiu, C.-C.: Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm. Comput. Stat. Data Anal. 52(10), 4658–4672 (2008), http://dx.doi.org/10.1016/j.csda.2008.03.002

Verbeek, J., Vlassis, N., Krose, B.: Self-organizing mixture models. Neurocomputing 63, 99–123 (2005)

Vesanto, J., Alhoniemi, E.: Clustering of the Self-Organizing Map. IEEE Transactions on Neural Networks 11(3), 586–600 (2000)

Wang, C.-M., Huang, Y.-F.: Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data. Expert Systems with Applications 36(3, Part 2), 5900–5908 (2009)

Wang, Q., Ye, Y., Huang, J.Z.: Fuzzy K-Means with Variable Weighting in High Dimensional Data Analysis. In: International Conference on Web-Age Information Management, vol. 0, pp. 365–372 (2008), http://doi.ieeecomputersociety.org/10.1109/WAIM.2008.50

Wiratunga, N., Lothian, R., Massie, S.: Unsupervised Feature Selection for Text Data. In: Roth-Berghofer, T.R., Göker, M.H., Güvenir, H.A. (eds.) ECCBR 2006. LNCS (LNAI), vol. 4106, pp. 340–354. Springer, Heidelberg (2006)

Yacoub, M., Bennani, Y.: Features Selection and Architecture Optimization in Connectionist Systems. IJNS 10(5) (2000)

# Two Variants of the OKM for Overlapping Clustering

Guillaume Cleuziou

**Abstract.** This paper deals with overlapping clustering and presents two extensions of the approach OKM denoted as OKMED and WOKM. OKMED generalizes the well known *k*-medoid method to overlapping clustering and help in organizing data with any proximity matrix as input. WOKM (Weighted-OKM) proposes a model with local weighting of the clusters; this variant is suitable for overlapping clustering since a single data can matches with multiple classes according to different features. On text clustering, we show that OKMED has a behavior similar to OKM but offers to use metrics other than euclidean distance. Then we observe significant improvement using the weighted extension of OKM.

**Keywords:** Overlapping clustering, medoid-based clustering, local weighting.

## 1 Introduction

Overlapping clustering is a specific task in Pattern Recognition, it consists in organizing a dataset into clusters that contain similar data and such that data belong to *at least* one cluster. This type of clustering is a natural way to organise data for a large number of real world applications. Information Retrieval requires to cluster documents by domain and each document is potentially multi-domains. In Bioinformatics the gene to cluster can reach into several metabolic pathways. In Natural Language Processing a verb can satisfy to multiple sub-categorization framework, etc.

As for usual clustering, there are no more trivial solutions to obtain absolute overlapping clusters. Furthermore, the search space (set of coverages) is much more big in case of overlaps than in case of crisp clustering.

During the four last decades some solutions have been proposed specifically for overlapping clustering. Dattola (1968) used a reallocating approach with multiple

Guillaume Cleuziou
LIFO, University of Orléans
e-mail: guillaume.cleuziou@univ-orleans.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 149–166.
springerlink.com 

assignments of the data based on a predefined threshold. Jardine and Sibson (1971) introduced the *k*-ultrametrics that lead to fundamental studies on overlapping hierarchies: pyramids (Diday, 1987) or weak hierarchies (Bertrand and Janowitz, 2003). More recently, under the pressure of applications in Information Retrieval or Bioinformatics, new investigations have been led in order to extend the partitioning models (*k*-means or CEM) for overlapping considerations. In such a way Banerjee *et al.* (2005a) proposed the Model-based Overlapping Clustering (MOC) that generalizes CEM (Celleux and Govaert, 1992) and Cleuziou (2008) extended the well-known *k*-means approach (MacQueen, 1967) with OKM (Overlapping *k*-means). The two last solutions are very closed and the main differences concern (1) the way to define intersections between clusters and (2) the algorithm associated (initialization and assignments). A more theoretical and experimental comparison is presented in Cleuziou and Sublemontier (2008).

The underlying model common to OKM and MOC provides a general framework allowing the exploration of many tracks. For example, many extensions of *k*-means have been proposed: to determine a suitable number of classes *k* (D.Pelleg and Moore, 2000), to limit the risk to obtain a locally optimal solutions (Likas *et al.*, 2003) or to initialize the algorithm intelligently (Peña *et al.*, 1999). In the present study we chose to explore two specific variants for OKM in order to provide a solution to practical problems in the domain: (1) the necessity to diversify the metrics used and (2) the possibility for a data to be a assigned to different clusters on the basis of different sets of features.

We then propose first the extension OKMED that uses the medoid-based clustering framework and allows to organize a dataset into overlapping clusters given any proximity matrix as input. OKMED requires to define judiciously the notion of *overlap representative* and begs a theoretical complexity problem that can be easily get round with practical heuristics.

The second contribution is a weighted extension WOKM that generalizes OKM by introducing local weighting for each cluster. WOKM takes a leaf out of the weighted *k*-means algorithm proposed by Chan *et al.* (2004) and refers more fundamentally to the "adaptative distances" introduced by Diday and Govaert (1977) ; it seems to be particularly suitable for overlapping clustering: by attaching different weights for the features in each cluster, a data is seen differently from one cluster to another, then a same data can naturally belongs to different clusters for different reasons (features). We will show that the translation from the initial weighted partitioning model to the overlapping one is not trivial and we will propose algorithmic solution allowing to ensure the convergence of the method. The efficiency of the proposed solutions will be assessed by experiments on real datasets.

The paper is organized in four main sections: Section 2 gives the general formal framework of the overlapping models OKM and MOC in order to better understand the two following sections that concern the variants OKMED and WOKM respectively. Before to conclude, Section 5 is dedicated to experiments performed on real text clustering datasets and various multi-labelled benchmarks.

## 2 MOC and OKM : Formal Framework

The model MOC proposed by Banerjee *et al.* (2005a) and the model OKM proposed by Cleuziou (2008) are some (overlapping) extensions of the methods based on reallocation around mobile centers. MOC is initially formalized in term of overlapping mixture models. However, the optimization of the objective criterion (log-likelihood) requires :

- a restriction in the generative model: constant and equal variances,
- a simplification in the algorithm: CEM rather than EM.

In such a way, MOC can be seen as an optimization method based on an inertia criterion that is formalized as a least square criterion.

Let $\mathcal{X} = \{x_i\}_{i=1}^n$ be a dataset in $\mathbb{R}^p$, the objective function used in the models MOC and OKM can be expressed in the following common formalism:

$$\mathcal{J}(\{\pi_c\}_{c=1}^k) = \sum_{x_i \in \mathcal{X}} \|x_i - \phi(x_i)\|^2 \tag{1}$$

In criterion (1) the $\{\pi_c\}_{c=1}^k$ denote the $k$ overlapping classes and $\phi(x_i)$ denotes the representative of $x_i$ into the clustering scheme, called "image" of $x_i$ by Cleuziou (2007). The image is obtained by a combination of class centers $\{m_c\}_{c=1}^k$ for the classes where $x_i$ appears: a sum in the model MOC and an arithmetic average in the model OKM:

$$\phi_{MOC}(x_i) = \sum_{m_c \in A_i} m_c \quad ; \quad \phi_{OKM}(x_i) = \frac{\sum_{m_c \in A_i} m_c}{|A_i|} \tag{2}$$

with $A_i = \{m_c | x_i \in \pi_c\}$ be the set of the class centers where $x_i$ appears.

The previous objective criterion suggests two remarks:

- the objective criterion (1) is an inertia criterion as for the least square criterion used in $k$-means; indeed it expresses the inertia of the data $\{x_i\}_{i=1}^n$ with respect to their respective image $\{\phi(x_i)\}_{i=1}^n$ in the clustering.
- in case of partitioning (non-overlapping clusters), each data belongs to only one cluster ($\forall i,\ |A_i| = 1$); for both models the image $\phi(x_i)$ of $x_i$ matches with the center $m_c$ of the cluster where $x_i$ appears; the objective criterion is then exactly the least square criterion (sum of the distances to the center), in this way MOC and OKM generalize $k$-means.

The optimization[1] of the objective criterion (minimization) is performed by the iteration of the two traditional steps: the computation of the class parameters (the centers $\{m_c\}_{c=1}^k$) and the assignment of each data to the clusters (single or multi-assignment in our case). The algorithms MOC and OKM use different heuristics

[1] The problem is not convex and the optimization process allows to provide a locally optimal solution as for analogous partitioning methods.

for the initialization of the parameters and for the multi-assignment step that is a combinatorial problem.

## 3 OKMED as a Generalization of $k$-Medoids

### 3.1 Motivation and Medoid-Based Methods

The medoid-based methods consists in aggregating the data around representatives or prototypes of the clusters, the prototypes - denoted as *medoids* being chosen among the data themselves. In this way it differs from centroid-based methods where the cluster prototypes do not necessary belong to $\mathcal{X}$.

The algorithm PAM (*Partitioning Around Medoids*) proposed by Kaufman and Rousseeuw (1987) is considered as a reference in this research field. PAM builds a partition of the data by iterating two steps: assignment of each data to its nearest medoid and updating of the medoid for each cluster.

During the second step, medoids updating consist in searching among the set of data belonging to the cluster, the one that minimizes the sum of the distances with any other data into the cluster.

The two main advantages of these methods are: firstly their robustness as regards to the outliers and secondly the possibility they offer to use any metric since they only require a proximity matrix over the dataset; the second point specifically motivates the present study. Indeed, the current overlapping models MOC and OKM are limited for the moment to a restricted family of metrics: the Bregman divergences, and the extension to other measures is not trivial. Roughtly speeking, a Bregman divergence $d_f$ is defined as

$$d_f(x,y) = f(x) - f(y) - \langle x-y, \nabla f(y) \rangle$$

with $f$ a strictly convex function; the squared euclidean distance and the Kullback-Leibler divergence are two of the widely used Bregman divergences (see Banerjee *et al.* (2005b) for more details on Bregman divergence).

### 3.2 The Model OKMED

We propose, for the model OKMED, to generalize the objective criterion (1) of the original model OKM, to any distance or dissimilarity between the data. Let $\mathcal{X} = \{x_i\}_{i=1}^n$ be a dataset and $d$ be a dissimilarity from $\mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$, the objective criterion for OKMED is given by:

$$\mathcal{J}(\{\pi_c\}_{c=1}^k) = \sum_{x_i \in \mathcal{X}} d^2(x_i, \phi(x_i)) \tag{3}$$

Again, the objective aims at minimizing the inertia of the data with respect to their image. The notion of image has to be redefined using cluster medoids rather than centroids: the image $\phi_{OKMED}(x_i)$ of the data $x_i$ in the clustering $\{\pi_c\}_{c=1}^k$ is then

defined as the data from $\mathcal{X}$ that minimizes the sum of the dissimilarities with all the medoids of the clusters where $x_i$ appears:

$$\phi_{OKMED}(x_i) = \arg\min_{x_j \in \mathcal{X}} \sum_{m_c \in A_i} d(x_j, m_c) \tag{4}$$

Let us notice that, with this new definition, the computation of an image requires to test all the data in $\mathcal{X}$. In practice, each image computation can be performed only once per combination[2] of assignments $A_i$ observed.

We can finally mention that in case of single assignments (crisp partitioning), each data $x_i$ belonging to only one cluster $\pi_c$, the image $\phi(x_i)$ is exactly the medoid $m_c$. Thus $k$-medoids must be considered has a special case of the model OKMED, *via* the objective criterion (3) previously defined.

### 3.3 *The Algorithm* OKMED

In the same way that aggregating methods, we propose an algorithm that aims at minimizing the criterion (3) by iterating two steps: assignments of the data and updating of the parameters (medoids). Figure 1 gives the description of the algorithm.

OKMED($D$,$k$,$t_{max}$,$\varepsilon$)

**Input:** $D$ a dissimilarity matrix from $(n \times n)$ over the dataset $\mathcal{X}$, $k$ the number of expected clusters, $t_{max}$: the maximum number of iterations (optional), $\varepsilon$: a convergence parameter (optional).

**Output:** $\{\pi_c\}_{c=1}^{k}$: a set of overlapping clusters that covers $\mathcal{X}$ (coverage).

1. Pick randomly $k$ medoids $\{m_c^{(0)}\}_{c=1}^{k}$ in $\mathcal{X}$,
   t=0.
2. For each data $x_i \in \mathcal{X}$ compute the assignments

$$A_i^{(t+1)} = \text{ASSIGN}(x_i, \{m_c^{(t)}\}_{c=1}^{k})$$

   derive a clustering $\{\pi_c^{(t+1)}\}_{c=1}^{k}$ such that $\pi_c^{(t+1)} = \{x_i | m_c^{(t)} \in A_i^{(t+1)}\}$
3. For each cluster $\pi_c^{(t+1)}$ successively, compute the new medoid

$$m_c^{(t+1)} = \text{MEDOID}(\pi_c^{(t+1)}, k)$$

4. If $\{\pi_c^{(t+1)}\}$ is different from $\{\pi_c^{(t)}\}$ or $t < t_{max}$ or $\mathcal{J}(\{\pi_c^{(t)}\}) - \mathcal{J}(\{\pi_c^{(t+1)}\}) > \varepsilon$, then $t = t+1$ and GoTo step 2; Else return the final clustering $\{\pi_c^{(t+1)}\}_{c=1}^{k}$.

**Fig. 1** Algorithm OKMED

[2] The number of possible such combinations is theoretically high, however only few combinations are observed in real situations.

The assignment of a data to one or several clusters is performed by the function ASSIGN that uses an heuristic proposed by Cleuziou (2008). Its adaptation for OKMED consists for each data $x_i$, in considering the medoids with a specific order (from the nearest to the farthest from $x_i$ according to $D$) and to assign $x_i$ to the associated cluster while the inertia $d(x_i, \phi(x_i))$ decreases. The new assignment $A_i^{(t+1)}$ is stored only if it improves the previous assignment $A_i^{(t)}$ as regards to the objective criterion ; by the way, the criterion is ensured to decrease at this step of the algorithm.

The updating of the parameters concerns the search of new representatives or medoids for each cluster that improve the objective criterion. The heuristic we propose for the search is formalized by the function MEDOID (cf. Figure 2); it searches *a* relevant medoid rather than *the* best (or optimal) medoid, in the sense of the objective criterion, for two main reasons:

- firstly because it is preferable to limit the medoid evaluations which are very costly in our overlapping context since they require, for each data belonging to the cluster, the computation of its image using the possible new medoid.
- then, in order to avoid as possible to choose a data that belongs to many other clusters as medoid for a cluster ; a data belonging only to the considered cluster would be preferred if it allows to improve the objective criterion.

MEDOID($\pi_c$,k)

**Input:** $\pi_c$ a cluster over the dataset $\mathcal{X}$, $k$ the number of expected clusters.
**Output:** $m_c$ the medoid for cluster $\pi_c$.

1. Compute the inertia of the data from $\pi_c$:

$$\mathcal{J}(\pi_c) = \sum_{x_i \in \pi_c} d^2(x_i - \phi(x_i))$$

2. **For** $b$ **from** 1 **to** $k$ **do:**

**For each** $x_j \in \pi_c$ **such that** $|A_j| = b$ **do:**

compute the images $\phi(x_i)'$ with $m_c = x_j$ for each $x_i \in \pi_c$
compute the new inertia for $\pi_c$

$$\mathcal{J}'(\pi_c) = \sum_{x_i \in \pi_c} d^2(x_i - \phi'(x_i))$$

if $\mathcal{J}'(\pi_c) < \mathcal{J}(\pi_c)$ return $x_j$ (new medoid for $\pi_c$)

**Fig. 2** Updating of the cluster medoids

Each one of the two steps - assignment and medoid computation - allows to improve the objective criterion (3) ; thus, by noticing that the set of solutions is finite[3]

[3] Set of overlapping clustering with $n$ data and $k$ clusters.

we can conclude on the convergence of the algorithm OKMED. The final clustering refers to a local optimum of the objective criterion depending of the initialization performed.

Finally, if the non-overlapping algorithm PAM has a quadratic complexity, the computation of the images in OKMED is costly and induces a complexity in $O(tn^3k)$, where $t$, $n$ and $k$ denote the number of iterations, the size of the dataset and the number of clusters respectively.

## 4 Local Weighting and Overlapping Clustering with WOKM

### *4.1 Motivation and Initial Model*

Let us consider as example the problem of text clustering where each text is described by a vector of word frequencies, given a fixed vocabulary. If the aim is to organize texts based on the domain (or thematic), we can logically think that some texts deal with only one domain (specific sub-vocabulary) and some other texts deal with several domains (mixed sub-vocabularies). By the way, overlapping clustering is clearly a better organizational structure compared to a crisp partitioning. However, with the models mentioned previously (MOC and OKM), even if a multi-domain data has a strong opportunity to be assigned to several clusters, the presence of a sub-vocabulary $S_1$ tends to penalize the assignment to a cluster impacted by another sub-vocabulary $S_2$.

Clustering models with local weighting of the clusters aim precisely at avoiding this phenomenon by allowing a data to be assigned to a cluster as regards to a subset of features that are important for the cluster concerned. By the way, the presence of a sub-vocabulary $S_1$ (subset of features) would be ignored during the process of assignment to a cluster that is impacted only by a sub-vocabulary $S_2$ . Intuitively, local weighting models are particularly suitable in the overlapping clustering framework.

In this section, we extend the weighting-$k$-means model (WKM) proposed by Chan *et al.* (2004) to the overlapping context. WKM generalizes the least square criterion used in $k$-means by mean of a feature weighting that is different for each cluster. Let $\mathcal{X} = \{x_i\}_{i=1}^n$ a dataset in $\mathbb{R}^p$, the objective criterion used in WKM is as follows:

$$\mathcal{J}(\{\pi_c\}_{c=1}^k) = \sum_{c=1}^{k} \sum_{x_i \in \pi_c} \sum_{v=1}^{p} \lambda_{c,v}^{\beta} |x_{i,v} - m_{c,v}|^2 \text{ with } \forall c, \sum_{v=1}^{p} \lambda_{c,v} = 1 \qquad (5)$$

The term $\lambda_{c,v}$ used in (5) denotes the weight associated to feature $v$ in cluster $c$ and $\beta$ is a parameter ($> 1$) that regulates the influence of the local weighting in the model.

With this framework, we propose in the next section the model WOKM that generalizes both OKM and WKM models.

### 4.2 *The Model* WOKM

The integration of the local weights to the clusters into the objective criterion used in overlapping clustering (1) is not trivial. Indeed, the inertia measures the scattering of the data with respect to their image rather than their cluster representative. We then have first to define the image of a data into the framework with local weights. We propose to define the image of $x_i$ by a weighted average of the cluster centroids for $x_i$:

$$\phi_{WOKM}(x_i) = (\phi_1(x_i), \ldots, \phi_p(x_i)) \text{ with } \phi_v(x_i) = \frac{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta} m_{c,v}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} \tag{6}$$

The previous definition ensures: on the one hand the model to be general and on the other hand an intuitive construction for the data images in the weighted overlapping clustering. In addition, let us notice that the image of a data $x_i$ characterizes a point in $\mathbb{R}^p$ that is representative of the intersection of the clusters from $A_i$. Because a vector of weights $\lambda_c$ is associated to each cluster $\pi_c$, we must propose a weighting for the overlaps, in other words we must propose a vector of weights $\gamma_i$ for the images $\phi(x_i)$. This vector is defined as follows:

$$\gamma_{i,v} = \frac{\sum_{m_c \in A_i} \lambda_{c,v}}{|A_i|} \tag{7}$$

From this definition it results the following objective criterion for the model WOKM:

$$\mathcal{J}(\{\pi_c\}_{c=1}^{k}) = \sum_{x_i \in \mathcal{X}} \sum_{v=1}^{p} \gamma_{i,v}^{\beta} |x_{i,v} - \phi_v(x_i)|^2 \tag{8}$$

The criterion (8) is subjects to the following constraint $\forall c,\ \sum_{v=1}^{p} \lambda_{c,v} = 1$ on the local weights of the clusters, this weights being encapsulated into the definition of the image weights $\{\gamma_i\}$. Let us notice that the model we propose generalizes previous models:

- in case of single assignments (crisp partitioning), if $x_i \in \pi_c$ then $\phi_v(x_i) = m_{c,v}$ and $\gamma_{i,v} = \lambda_{c,v}$; the objective criterion (8) is then equivalent to the one used in WKM (5).
- in case of uniform weighting ($\forall c,\ \forall v,\ \lambda_{c,v} = 1/p$), $\gamma_{i,v} = 1/p$ and $\phi_{WOKM}(x_i) = \phi_{OKM}(x_i)$; the objective criterion (8) is then equal to the one used in OKM (1).

### 4.3 *The Algorithm* WOKM

The optimization of (8) is performed with an algorithm that iterates three steps: assignment, cluster centers updating and weights updating (cf. Figure 3).

WOKM($\mathcal{X}$,$k$,$t_{max}$,$\varepsilon$)

**Input:** $\mathcal{X}$ a dataset over $\mathbb{R}^p$, $k$ the number of expected clusters, $t_{max}$ the maximum number of iterations (optional), $\varepsilon$ a convergence parameter (optional).

**Output:** $\{\pi_c\}_{c=1}^k$: a set of overlapping clusters that covers $\mathcal{X}$ (coverage).

1. Pick randomly $k$ centres $\{m_c^{(0)}\}_{c=1}^k$ from $\mathbb{R}^p$ or from $\mathcal{X}$,
   Initialize the weights $\{\lambda_{c,v}^{(0)}\}$ in an uniform manner ($\lambda_{c,v}^{(0)} = 1/p$),$t = 0$.
2. For each data $x_i \in \mathcal{X}$ compute the assignments

$$A_i^{(t+1)} = \text{ASSIGN}(x_i, \{m_c^{(t)}\}_{c=1}^k)$$

   derive a clustering $\{\pi_c^{(t+1)}\}_{c=1}^k$ such that $\pi_c^{(t+1)} = \{x_i | m_c^{(t)} \in A_i^{(t+1)}\}$
3. For each cluster $\pi_c^{(t+1)}$ successively, compute the new centroid

$$m_c^{(t+1)} = \text{CENTROID}(\pi_c^{(t+1)})$$

4. For each cluster $\pi_c^{(t+1)}$ successively, compute the new weights

$$\lambda_{c,.} = \text{WEIGHTING}(\pi_c^{t+1})$$

5. If $\{\pi_c^{(t+1)}\}$ differs from $\{\pi_c^{(t)}\}$ or $t < t_{max}$ or $\mathcal{J}(\{\pi_c^{(t)}\}) - \mathcal{J}(\{\pi_c^{(t+1)}\}) > \varepsilon$, then $t = t+1$ and GoTo step 2; Else, return the clustering $\{\pi_c^{(t+1)}\}_{c=1}^k$.

**Fig. 3** Algorithm WOKM

The assignment step (ASSIGN) is similar with the corresponding step in algorithms OKM and OKMED: the data is assigned to its nearest clusters while $\sum_{v=1}^p \gamma_{i,v}^\beta |x_{i,v} - \phi_v(x_i)|^2$ decreases.

The second step (CENTROID) that updates the cluster centers can be performed on each cluster successively by considering the other centroids fixed; the associated convex optimization problem is solved by defining the new optimal center $m_c^*$ for cluster $\pi_c$ as the center of gravity of the dataset $\{(\hat{x}_i^c, w_i) | x_i \in \pi_c\}$; $\hat{x}_i^c$ denoting the cluster center $\pi_c$ that would allow the image $\phi(x_i)$ to match exactly with the data $x_i$ itself ($\forall v,\ |x_{i,v} - \phi_v(x_i)| = 0$) and $w_i$ denotes the associated vector of weights defined as follows: $w_{i,v} = \dfrac{\gamma_{i,v}^\beta}{\left(\sum_{m_l \in A_i} \lambda_{l,v}^\beta\right)^2}$ (see the Appendix for more details on the problem solving).

The third step (WEIGHTING) updates the vectors of local weights $\{\lambda_c\}_{c=1}^k$; the optimization problem with constraint ($\sum_{v=1}^p \lambda_{c,v} = 1$) cannot be directly solved because the vectors $\lambda_c$ are mutually dependant, contrary to the non-overlapping model that refers to the theorem from Bezdek (1981) to determine optimal weights. We

then propose a new heuristic based on the Bezdeck theorem; the heuristic consists for each class in:

1. computing a new weighting $\lambda_{c,v}$ for the cluster $\pi_c$ by estimating on each feature the variance of the data that belong only to $\pi_c$:

$$\lambda_{c,v} = \frac{\left(\sum_{\{x_i \in \pi_c |\ |A_i|=1\}} (x_{i,v} - m_{c,v})^2\right)^{1/(1-\beta)}}{\sum_{u=1}^{p} \left(\sum_{x_i \in \pi_c |\ |A_i|=1} (x_{i,u} - m_{c,u})^2\right)^{1/(1-\beta)}}$$

2. storing the computed weighting only if it improves the objective criterion (8) associated to the model of WOKM.

Let us notice that the heuristics used for the assignments and the weights updating are both performed in such a way that the objective criterion decreases in order to ensure the WOKM algorithm to converge.

As for the non-overlapping approach, the algorithm WOKM has a complexity linear on the size of the dataset ($n$). The order of complexity is $O(tnpk\ log\ k)$ where $p$ denote the size of the feature set.

## 5 Experiments

In this section we present experiments that aim at observing the behavior of the two variants OKMED and WOKM. The first dataset (Iris) is commonly used in categorization or clustering, it helps in making a first impression about the efficiency of a new classification method. The second dataset Reuters-21578[4] (Apté *et al.*, 1994) concerns the text clustering task that matches exactly with the target application domains since the texts are precisely multi-labelled. Finally, the overlapping clustering approaches are tested and compared on three other multi-labelled benchmarks with different properties (different numbers of data, features, clusters and different sizes of overlaps).

For each experiment, the proposed evaluation compares the obtained clustering with respect to a referent clustering given by the labels[5]. The comparison is quantified with the F-measure that combines precision and recall over the set of data associations retrieved or expected. Let $\Pi_r$ and $\Pi_o$ be the referent and obtained clusterings respectively, let $N_r$ and $N_o$ be the set of associations (pairs of points associated in a same cluster) in $\Pi_r$ and $\Pi_o$ :

$$\text{precision} = \frac{|N_o \cap N_r|}{|N_o|}\ ;\ \text{recall} = \frac{|N_o \cap N_r|}{|N_r|}\ ;\ F - \text{measure} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

The feature *assignment* also reported in the following experiments quantifies the importance of the overlaps in an overlapping clustering, it is defined by the average number of clusters each data belongs to.

[4] http://www.research.att.com/∼lewis/reuters21578.html

[5] The labels associated to each data are not used during the clustering process.

## 5.1 Preliminary Experiments on the Iris Dataset

The Iris dataset (from the UCI repository) (D.J. Newman and Merz, 1998) contains 150 data defined over $\mathbb{R}^4$ and equitably distributed over three classes ; one of these classes (*setosa*) is known to be clearly separated from the two others.

The values reported in Table 1 result from the average on 500 runs with $k = 3$. The methods being sensible to the initialization step, the initial cluster centers differs from one run to another but for each run the different algorithms have the same initialization.

**Table 1** Comparison of the models on Iris

| | Precision | Recall | F-measure | Assignment |
|---|---|---|---|---|
| $k$-means | 0.75 | 0.82 | 0.78 | 1.00 |
| $k$-medoids | 0.75 | 0.84 | 0.79 | 1.00 |
| Weighted $k$-means | 0.85 | 0.89 | 0.86 | 1.00 |
| OKM | 0.57 | 0.98 | 0.72 | 1.40 |
| OKMED | 0.61 | 0.88 | 0.71 | 1.16 |
| WOKM | 0.62 | 0.98 | 0.76 | 1.32 |

The results on the non-overlapping methods are given as a rough guide; since the dataset is not multi-labelled the overlapping methods are logically penalized by the evaluation process.

However a first result to notice is the F-measure obtained with OKMED that is almost equal to the F-measure observed with OKM; since this phenomenon is also observed on their correspondent non-overlapping models ($k$-medoids and $k$-means respectively) we shown experimentally that the model and the algorithm associated to OKMED generalizes $k$-medoids. We also notice, through the Assignment feature, that OKMED induces smaller overlaps than OKM; this is explained by the fact that in OKMED the set of possible images $\phi(x_i)$ for the data $x_i$ is finite (and limited to $\mathcal{X}$) contrary to the model OKM with images computed in $\mathbb{R}^p$.

About the weighted clustering models, we observe that the weighted models outperform unweighted correspondent models. With this experiment we thus confirm that WOKM must be considered as a generalization of WKM and above all that our intuition about the contribution of local weights into the overlapping framework seems to be verified.

## 5.2 Experiments on the Reuters Dataset

The second series of experiments is performed on the Reuters dataset that is commonly used as benchmark for Information Retrieval tasks. Because the number of runs per test is high and to allow the multiplication of the tests (different methods, different parameters $k$, etc.) we consider only a subset of 300 texts described

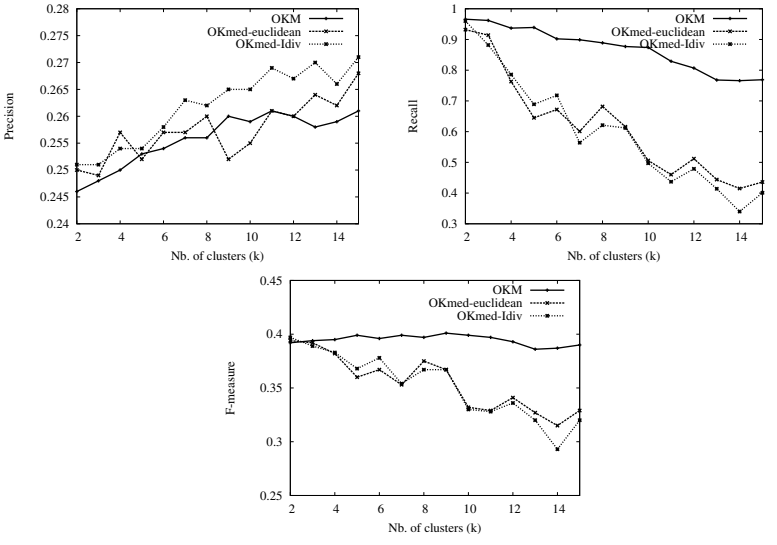



**Fig. 4** OKMED with different metrics

by word frequencies; the vocabulary being composed of 500 words with highest tf×idf.

In order to show the contribution of OKMED *via* the unrestricted set of metrics it allows to employ, a comparison between (1) OKM, (2) OKMED with euclidean distance and (3) OKMED with Kullback-Leibler divergence[6] (or I-Divergence) is reported in Figure 4 with different values for parameter $k$.

We observe that OKMED has a behaviour stable for the different metrics and above all we notice that the use of the I-Divergence allows to outperform other solutions as regards to the precision. The seeming superiority of OKM on the F-measure is actually due to excessive overlaps inducing a recall artificially high.

Finally, the curves reported in Figure 5 detail the contribution of the weighted clustering models, especially on the overlapping framework.

Local weighting seems not to significantly contribute in non-overlapping models ($k$-means w.r.t. WKM), the contribution is noticeable in case of overlapping clustering and it results:

1. a restriction on the size of the overlaps (lower average number of assignment per data);
2. a limited repercussion (of the diminution of the overlaps) on the recall;
3. a significant improvement of the precision.

Generally speaking, the local weighting introduced with WOKM seems allowing to adjust the model OKM by a limitation of the parasitic (or excessive) multi-assignment.

[6] Frequently used for text analysis.

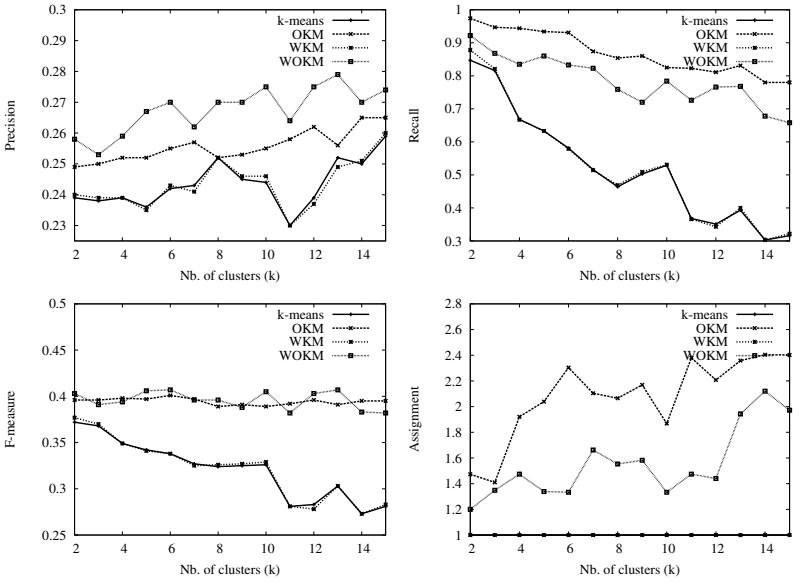



**Fig. 5** Influence of the local weighting of the clusters

## *5.3 Comparative Study on Three Multi-labelled Datasets*

We complete the preliminary experiments with a comparative study on three multi-labelled datasets with numerical features. It concerns different domains, the datasets have different number of data (instances), features and clusters (labels) and their overlaps (cardinality) are more or less important (cf. table 2).

**Table 2** Quantified description of the multi-labelled datasets

| name | domain | instances | features | labels | cardinality |
|---|---|---|---|---|---|
| Emotions | music | 593 | 72 | 6 | 1.87 |
| Scene | multimedia | 2407 | 294 | 6 | 1.07 |
| Yeast | biology | 2417 | 103 | 14 | 4.24 |

The dataset *emotions* (Tsoumakas *et al.*, 2008) contains 593 songs with a duration of 30 seconds, described with 72 rhythmic or timber features and manually labelled by experts through 6 emotional labels (happy, sad, calm, surprised, quiet, angry).

The *scene* dataset (Boutell *et al.*, 2004) is made up of color images described with color and space features (spatial color moments). Originally, one label was associated to each image (or scene) among the set of labels: *beach, sunset, fall foliage, field, mountain, urban*. After a human re-labeling, approximately 7.4% of the images belonged to multiple classes.

Finally, *yeast* (Elisseeff and Weston, 2001) is formed by micro-array expression data and phylogenetic profiles. The input dimension is 103. Each gene is associated with a set of functional classes.

The values in Tables 3 report precision, recall, F-measure, average assignments (or cardinality) and CPU time (in seconds) obtained with different overlapping clustering algorithm. They result from the average on 5 runs using the euclidean distance and a parameter $k$ that corresponds to the true number of labels.

**Table 3** Quantitative comparison of OKM, WOKM, OKMED and MOC on multi-labelled datasets

| **Precis.** | Emotions | Scene | Yeast |
|---|---|---|---|
| OKM | $0.49_{\pm 0.01}$ | $0.23_{\pm 0.00}$ | $0.78_{\pm 0.00}$ |
| WOKM | $0.49_{\pm 0.01}$ | $0.21_{\pm 0.00}$ | $0.78_{\pm 0.00}$ |
| OKMED | $0.49_{\pm 0.01}$ | $0.24_{\pm 0.01}$ | $0.79_{\pm 0.00}$ |
| MOC | $0.48_{\pm 0.01}$ | $0.42_{\pm 0.02}$ | $0.80_{\pm 0.00}$ |

| **Recall.** | Emotions | Scene | Yeast |
|---|---|---|---|
| OKM | $0.65_{\pm 0.07}$ | $0.94_{\pm 0.02}$ | $0.86_{\pm 0.03}$ |
| WOKM | $0.65_{\pm 0.07}$ | $0.59_{\pm 0.07}$ | $0.86_{\pm 0.03}$ |
| OKMED | $0.53_{\pm 0.06}$ | $0.74_{\pm 0.08}$ | $0.29_{\pm 0.02}$ |
| MOC | $0.21_{\pm 0.01}$ | $0.40_{\pm 0.05}$ | $0.94_{\pm 0.01}$ |

| **F-meas.** | Emotions | Scene | Yeast |
|---|---|---|---|
| OKM | $0.56_{\pm 0.03}$ | $0.36_{\pm 0.00}$ | $0.82_{\pm 0.01}$ |
| WOKM | $0.56_{\pm 0.03}$ | $0.31_{\pm 0.01}$ | $0.82_{\pm 0.01}$ |
| OKMED | $0.50_{\pm 0.03}$ | $0.36_{\pm 0.01}$ | $0.42_{\pm 0.03}$ |
| MOC | $0.30_{\pm 0.01}$ | $0.41_{\pm 0.03}$ | $0.86_{\pm 0.00}$ |

| **Assign.** | Emotions | Scene | Yeast |
|---|---|---|---|
| OKM | $1.98_{\pm 0.2}$ | $2.43_{\pm 0.06}$ | $4.69_{\pm 0.10}$ |
| WOKM | $1.98_{\pm 0.2}$ | $1.20_{\pm 0.24}$ | $4.69_{\pm 0.10}$ |
| OKMED | $1.91_{\pm 0.2}$ | $2.17_{\pm 0.08}$ | $2.19_{\pm 0.11}$ |
| MOC | $1.00_{\pm 0.0}$ | $1.00_{\pm 0.00}$ | $6.05_{\pm 0.05}$ |

| **CPU time** | Emotions | Scene | Yeast |
|---|---|---|---|
| OKM | $3_{\pm 1.2}$ | $38_{\pm 12.8}$ | $55_{\pm 32}$ |
| WOKM | $23_{\pm 8.0}$ | $106_{\pm 110}$ | $559_{\pm 279}$ |
| OKMED | $20_{\pm 7.5}$ | $4259_{\pm 1089}$ | $1419_{\pm 217}$ |
| MOC | $1_{\pm 0.5}$ | $50_{\pm 0.02}$ | $2048_{\pm 359}$ |

We first notice that OKMED performs as well as OKM on *scene* and obtain lower results on the two other datasets. This result is mainly due to the smaller overlaps allowed by OKMED, and particularly on the *yeast* dataset since the overlaps produced by OKM are twice greater than for OKMED. The higher complexity of the medoid-based algorithm is clearly observed experimentally with the time costs reported on the last table: few seconds are sufficient to deal with the 593 instances from *emotions* and more than 20 minutes are required to deal with the 2,400 instances from *scene* and *yeast*.

The motivations for the weighted variant of OKM is confirmed by this experiment since we observe that WOKM produces overlaps more realistic (smaller) as regard to the true cardinality of the datasets (e.g. on *scene*). Conversely, OKM producing excessive multi-assignments, it obtains logically higher F-measure due to the (imperfect) evaluation process.

At last, but not at least, the MOC approach fails in discovering a suitable overlapping structure. It results in either no overlaps (for *emotions* and *scene*) or too much overlaps (*yeast*). The other evaluation scores are thus difficult to compare between very different structuring.

## 6 Conclusion and Perspectives

We proposed in this paper two contributions in the domain of overlapping clustering. The first contribution is the model OKMED that draw one's inspiration from medoid-based partitioning methods; OKMED allows to consider any proximity measure as input for the overlapping clustering task contrary to the original model OKM which is - at the moment - restricted to the euclidean distance. The second contribution aims at introducing local weighting framework into overlapping clustering models, by means of the algorithm WOKM.

As illustrated in Figure 6, the models OKMED and WOKM are presented as generalizations of both:

- crisp-partitioning models: *k*-means, *k*-medoids and weighted-*k*-means,
- overlapping models: OKM and MOC.

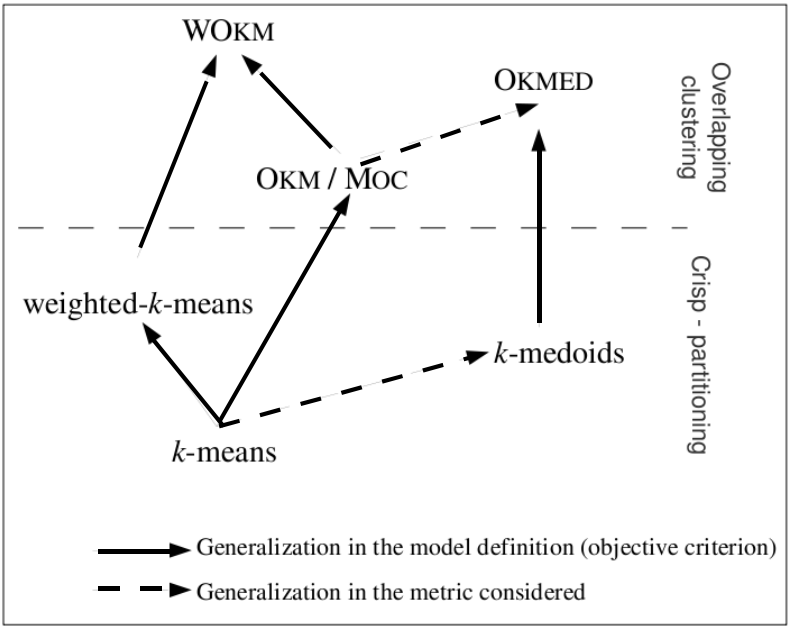



**Fig. 6** Theoretical organization of the clustering models

We proposed for each model an algorithm that leads to an overlapping clustering with strategies driven by the associated objective criterion. The two models have been tested, compared and validated with experiments on suitable multi-labelled datasets from very different domains.

We plan to progress in the extension of the overlapping clustering family of methods by investigating other relevant variants such as the self-organizing maps

(Kohonen, 1984) or kernelized clustering (Dhillon, 2004) in the overlapping framework. In addition, the two models proposed in the present study could be used as a basis framework for the development of a new approach that would combine the benefits of both models into a medoid-based overlapping clustering capturing cluster shapes.

## References

Apté, C., Damerau, F., Weiss, S.M.: Automated learning of decision rules for text categorization. ACM Trans. Inf. Syst. 12(3), 233–251 (1994), http://doi.acm.org/10.1145/183422.183423

Banerjee, A., Krumpelman, C., Ghosh, J., Basu, S., Mooney, R.J.: Model-based overlapping clustering. In: KDD 2005: Proceeding of the eleventh ACM SIGKDD, pp. 532–537. ACM Press, New York (2005a), http://doi.acm.org/10.1145/1081870.1081932

Banerjee, A., Merugu, S., Dhillon, I., Ghosh, J.: Clustering with Bregman Divergences. J. Mach. Learn. Res. 6, 1705–1749 (2005b)

Bertrand, P., Janowitz, M.F.: The k-weak Hierarchical Representations: An Extension of the Indexed Closed Weak Hierarchies. Discrete Applied Mathematics 127(2), 199–220 (2003)

Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)

Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recognition 37(9), 1757–1771 (2004), http://dx.doi.org/10.1016/j.patcog.2004.03.009

Celleux, G., Govaert, G.: A Classification EM Algorithm for Clustering and Two Stochastic Versions. Computational Statistics and Data Analysis 14(3), 315–332 (1992)

Chan, E.Y., Ching, W.-K., Ng, M.K., Huang, J.Z.: An optimization algorithm for clustering using weighted dissimilarity measures. Pattern Recognition 37(5), 943–952 (2004)

Cleuziou, G.: OKM: une extension des k-moyennes pour la recherche de classes recouvrantes. In: EGC 2007, Cépaduès edn., Namur, Belgique. Revue des Nouvelles Technologies de l'Information, vol. 2 (2007)

Cleuziou, G.: An Extended Version of the k-Means Method for Overlapping Clustering. In: 19th ICPR Conference, Tampa, Florida, USA, pp. 1–4 (2008)

Cleuziou, G., Sublemontier, J.-H.: Etude comparative de deux approches de classification recouvrante: Moc vs. Okm. In: 8èmes Journées Francophones d'Extraction et de Gestion des Connaissances, Cépaduès edn. Revue des Nouvelles Technologies de l'Information, vol. 2 (2008)

Dattola, R.: A fast algorithm for automatic classification. Technical report, Report ISR-14 to the National Science Foundation, Section V, Cornell University, Department of Computer Science (1968)

Dhillon, I.S.: Kernel k-means, spectral clustering and normalized cuts, pp. 551–556. ACM Press, New York (2004)

Diday, E.: Orders and overlapping clusters by pyramids. Technical report, INRIA num.730, Rocquencourt 78150, France (1987)

Diday, E., Govaert, G.: Classification avec distances adaptatives. RAIRO 11(4), 329–349 (1977)

Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI Repository of machine learning databases. University of California, Irvine, Dept. of Information and Computer Sciences (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html

Pelleg, D., Moore, A.: X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 727–734. Morgan Kaufmann, San Francisco (2000)

Elisseeff, A., Weston, J.: A Kernel Method for Multi-Labelled Classification. In: Advances in Neural Information Processing Systems, vol. 14, pp. 681–687. MIT Press, Cambridge (2001)

Jardine, N., Sibson, R.: Mathematical Taxonomy. John Wiley and Sons Ltd., London (1971)

Kaufman, L., Rousseeuw, P.J.: Clustering by means of medoids. In: Dodge, Y. (ed.) Statistical Data Analysis based on the L1 Norm, pp. 405–416 (1987)

Kohonen, T.: Self-Organization and Associative Memory. Springer, Heidelberg (1984)

Likas, A., Vlassis, N., Verbeek, J.: The Global K-means Clustering Algorithm. Pattern Recognition 36, 451–461 (2003)

MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical statistics and probability, vol. 1, pp. 281–297. University of California Press, Berkeley (1967)

Peña, J., Lozano, J., Larrañaga, P.: An empirical comparison of four initialization methods for the k-means algorithm. Pattern Recognition Letters 20(50), 1027–1040 (1999)

Tsoumakas, G., Katakis, I., Vlahavas, I.: Effective and Efficient Multilabel Classification in Domains with Large Number of Labels. In: Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data, MMD 2008 (2008)

## Appendix

The updating of the centroids aims at finding the new cluster centers that make the WOKM objective criterion (8) to decrease. Each cluster center is updated successively in such a way that for each component $v$ and each cluster $\pi_{c^*}$ the computation of $m_{c^*,v}$ is a convex optimization problem:

$$\mathcal{J}_{c^*,v} = \sum_{x_i \in \pi_{c^*}} \gamma_{i,v}^{\beta} (x_{i,v} - \phi_v(x_i))^2 = \sum_{x_i \in \pi_{c^*}} \gamma_{i,v}^{\beta} \left( x_{i,v} - \frac{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta} m_{c,v}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} \right)^2 \quad (9)$$

Given the parameters $\{\lambda\}$, $\{m_{c,v}\}_{c \neq c^*}$ and the assignments $\{A_i\}$ fixed, the minimization of the objective criterion (8) is performed by the minimization of $\mathcal{J}_{c^*,v}$ that is reached for $\frac{\partial \mathcal{J}_{c^*,v}}{\partial m_{c^*,v}} = 0$.

$$\frac{\partial \mathcal{J}_{c^*,v}}{\partial m_{c^*,v}} = \sum_{x_i \in \pi_{c^*}} 2.\gamma_{i,v}^{\beta} \left( \frac{\lambda_{c^*,v}^{\beta}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} \right) \left( \frac{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta} m_{c,v}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} - x_{i,v} \right)$$

The problem is then to find $m_{c^*,v}$ such that

$$\sum_{x_i \in \pi_{c^*}} \gamma_{i,v}^{\beta} \left( \frac{\lambda_{c^*,v}^{\beta}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} \right) \left( \frac{\lambda_{c^*,v}^{\beta} m_{c^*,v}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} + \frac{\sum_{m_c \neq m_{c^*} \in A_i} \lambda_{c,v}^{\beta} m_{c,v}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} - x_{i,v} \right) = 0$$

$$\Leftrightarrow \sum_{x_i \in \pi_{c^*}} \gamma_{i,v}^{\beta} \left( \frac{\lambda_{c^*,v}^{\beta}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} \right)^2 \left( m_{c^*,v} - \hat{x_i}^{c^*} \right) = 0 \tag{10}$$

Where $\hat{x_i}^{c^*}$ in (10) denotes the cluster center $m_{c^*}$ that would allow the image $\phi(x_i)$ to match exactly with the data $x_i$ itself ($\forall v,\ |x_{i,v} - \phi_v(x_i)| = 0$) :

$$\hat{x_i}^{c^*} = \left( x_{i,v} - \frac{\sum_{m_c \neq m_{c^*} \in A_i} \lambda_{c,v}^{\beta} m_{c,v}}{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}} \right) . \frac{\sum_{m_c \in A_i} \lambda_{c,v}^{\beta}}{\lambda_{c^*,v}^{\beta}}$$

Finally, the solution of (10) is given by

$$m_{c^*,v} = \left( \frac{\sum_{x_i \in \pi_{c^*}} \gamma_{i,v}^{\beta} . \hat{x_i}^{c^*}}{\left( \sum_{m_c \in A_i} \lambda_{c,v}^{\beta} \right)^2} \right) / \left( \frac{\sum_{x_i \in \pi_{c^*}} \gamma_{i,v}^{\beta}}{\left( \sum_{m_c \in A_i} \lambda_{c,v}^{\beta} \right)^2} \right) \tag{11}$$

In other words, the solution $m_{c^*,v}$ is the center of gravity of the dataset $\{(\hat{x}_i^c, w_i) | x_i \in \pi_c\}$ where $w_i$ denotes the associated vector of weights defined as follows: $w_{i,v} = \frac{\gamma_{i,v}^{\beta}}{\left( \sum_{m_c \in A_i} \lambda_{c,v}^{\beta} \right)^2}$.

# A Stable Decomposition Algorithm for Dynamic Social Network Analysis

Romain Bourqui, Paolo Simonetto, and Fabien Jourdan

**Abstract.** Dynamic networks raise new challenges for knowledge discovery. To efficiently handle this kind of data, analysis methods have to decompose the network, modelled by a graph, into similar sets of nodes. In this article, we present a graph decomposition algorithm that generates overlapping clusters. The complexity of this algorithm is $O(|E| \cdot deg_{max}^2 + |V| \cdot log(|V|)))$. This algorithm is particularly efficient because it can detect major changes in the data as it evolves over time.

**Keywords:** Overlapping Clustering, Clusters Evolution, Social Network Analysis.

## 1 Introduction

A graph is a data structure used to organise large scale relational data; it is used in many application fields such as biology, micro-electronics and social sciences. Graph are particularly well suited for knowledge discovery, since there exists many algorithms to mine their structure and understand their underlying properties (e.g. Newman and Girvan, 2004; Palla *et al.*, 2007; Suderman and Hallett, 2007). Much attention has been given to the problem of identifying clusters in these networks. In the social sciences, clusters may represent groups of individuals sharing the same interests (communities), while, in biology, they may represent proteins involved in

Romain Bourqui
Eindhoven University of Technology, P.O. Box 513; 5600 MB Eindhoven, Netherlands
e-mail: R.Bourqui@tue.nl

Paolo Simonetto
LaBRI, Université Bordeaux 1, 351, cours de la Libération F-33405 Talence cedex
e-mail: simonett@labri.fr

Fabien Jourdan
INRA, UMR1089, Xénobiotiques, F-31000 Toulouse, France
e-mail: Fabien.Jourdan@toulouse.inra.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 167–178.
springerlink.com 

the same biological processes (e.g. Newman and Girvan, 2004; Palla *et al.*, 2007; Bader and Hogue, 2003). Clustering methods visual abstraction of the network to be built by aggregating clusters into single nodes. These abstractions are particularly interesting with dealing with large networks, as they reduce the number of elements displayed (Auber *et al.*, 2003).

Finding communities in a network is generally related to structural decomposition. Decomposition algorithms compute sets of elements (clusters) sharing one or more properties. To evaluate the quality of a decomposition one usually compares the interconnections within and between clusters. More precisely, a good decomposition will have a high intra-cluster density and a low inter-cluster density[1].

Dynamic data, in our case dynamic networks, are increasingly present, requiring knowledge discovery methods. Automatic data extractions are continuously improved and databases are populated quickly in biology (e.g. quantitative data on a organism evaluating according to environmental changes) and in the social sciences (e.g. co-citation networks, movie actor networks). Consequently, it is not only a question of identifying communities at a single time instant but it is also understanding the evolution of communities over time. In other words, structural changes, such for instance merges, expansions, splits of communities (Palla *et al.*, 2007).

In this article, we will approach the problem of dynamic network analysis by using static graph decomposition. This step can be used to detect topological changes in the graph as part of a larger framework, where dynamic graphs are turned into sequences of static graphs, decomposed in communities, and compared within consecutive time stamps.

This article is organised as follows. In section 2, we present the overall approach. In section 3, the decomposition algorithm is described. We evaluate the "stability" of the decomposition on a social network in section 4. Finally, we draw some conclusions in section 5.

## 2 Methodology

Our decomposition algorithm is part of the framework depicted in figure 1. The first step turns the dynamic network into a set of static graphs. If we consider a dynamic graph $G$, defined on a time interval $[0..T]$, this transformation consists in building a set of static graphs $\{G_{[0,\tau[}, ..., G_{[T-\tau,T]}\}$, where $\tau$ is the discretisation factor and $G_{[t,t+\tau[}$ is the static graph corresponding to the time period $[t,t+\tau[$ (i.e. this graph contains all the nodes and edges of the dynamic graph present during the period $[t,t+\tau[$).

The main idea behind our approach is that if the graph changes little (assuming that the discretisation factor is relevant) then two static graphs describing two consecutive periods have similar topological structures. Therefore, if our algorithm is stable enough, we will obtain a "similar" decomposition. To compare these decompositions, we introduce a similarity measure.

[1] Density is defined as the number of edges divided by the maximal number of possible edges.

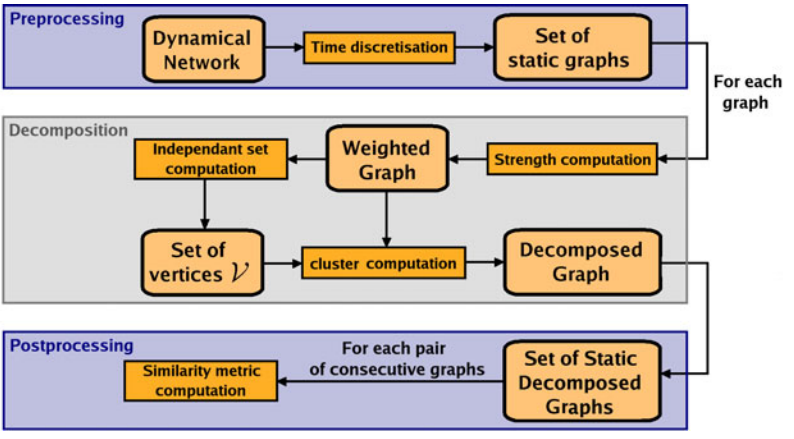



**Fig. 1** Three main steps of our method

In this article, we present a new algorithm that is fairly stable when few topological changes occour between two networks. This algorithm is divided into three steps. Firstly, the strength metric (Auber *et al.*, 2003) is computed on edges and nodes. Secondly, we look for a maximal set of independent nodes (i.e. nodes at least at distance two from each other). Finally, we build sets "around" these independent nodes.

## 3 Algorithm

### *3.1* Strength *Metric*

Auber *et al.* (2003) introduced a new metric called Strength to quantify the contribution of an edge (or a node) to the cohesion of its neighbourhood. More precisely, this metric counts for each edge, the number of cycles of length 3 and 4 going through that edge and then normalises the value by the maximum number of such cycles. As result, if a node or an edge forms part of a community its Strength value will be high.

To formally define Strength (Auber *et al.*, 2003; Chiricota *et al.*, 2003), we define some notation. Let $u$ and $v$ be two nodes of graph $G = (V,E)$. We denote $M_u(v) = N_G(v) \setminus (N_G(u) \cup \{u\})$ the neighbours of $v$ (excluding $u$) that are not in the neighbourhood of $u$. We denote $W_{uv} = N_G(u) \cap N_G(v)$ as the set of nodes in both the neighbourhoods of $v$ and $u$. Let $A$ and $B$ be two set of nodes, we note $E(A,B)$ the set of edges linking a node in $A$ to a node in $B$. Finally, $s(A,B) = |E(A,B)|/(|A| \cdot |B|)$ is the ratio between the number of edges linking $A$ and $B$ to the maximum number of edges that could link these two sets[2]. Strength metric value of an edge $e = (u,v)$ is:

[2] When $A = B$, then $s(A,A) = s(A) = 2 \cdot |E(A)|/(|A| \cdot (|A| - 1))$.

$$w_s(e) = \frac{\gamma_{3,4}(e)}{\gamma_{max}(e)} \tag{1}$$

Where:

$$\begin{aligned}\gamma_{3,4}(e) = {} & |W_{uv}| + |E(M_v(u), M_u(v))| + |E(M_v(u), W_{uv})| \\ & + |E(W_{uv}, M_u(v))| + |E(W_{uv})|\end{aligned} \tag{2}$$

$$\begin{aligned}\gamma_{max}(e) = {} & |M_v(u)| + |W(u,v)| + |M_u(v)| + |M_v(u)||M_u(v)| \\ & + |M_v(u)||W_{uv}| + |W_{uv}||M_u(v)| + |W_{uv}|(|W_{uv}| - 1)/2\end{aligned} \tag{3}$$

The authors subsequently define the Strength $w_s(u)$ of node $u$ as follows:

$$w_s(u) = \frac{\sum_{e \in inc(u)} w_s(e)}{deg(u)}$$

Where $inc(u)$ is the set of edges incident to $u$ and $deg(u)$ the degree of $u$.

The time complexity to compute Strength is $O(|E| \cdot deg_{max}^2)$ where $deg_{max}$ is the maximum degree of a node in the graph.

### 3.2 Extracting a Maximal Independent Set

In this step, the algorithm finds community centres used to identify clusters. To do so, we develop a method inspired by *MISF* (Maximal Independent Set Filtering) of Gajer and Kobourov (2000). Our approach extracts a maximal set $\mathcal{V} \in V$ such that $\forall u, v \in \mathcal{V}$, $dist_G(u,v) \geq 2$, where $dist_G(u,v)$ is the theoretical graph distance between $u$ and $v$. Selecting nodes at distance 2 in the graph allows to obtain a representative sampling of the nodes in the network. Moreover, the number of selected nodes defines the number of communities in the network. Finally, this technique guaranties the uniqueness of each group found by our algorithm (two sets can not contain exactly the same set of nodes).

Given that nodes in $\mathcal{V}$ will be centres of the communities, these nodes should not be pivots in the network. In fact, if we consider a pivot as a centre of a cluster, this cluster may contain several communities. For instance, in figure 2.(b), the "central" node has been chosen as a community centre, resulting in the generation of a single group.

Pivots of the network can be identified using Strength metric, as they have a relatively low Strength value. Thus, nodes with a high Strength value are preferentially added to the set $\mathcal{V}$ (see figure 2.(c)). To select this set of nodes, we developed the algorithm 1. In this algorithm, nodes are first sorted by increasing Strength value. Then, the first node of the list still in the graph is added to $\mathcal{V}$, and it and its neighbours are removed from graph. This last step is repeated until the graph contains no nodes, allowing to guarantee that the nodes in $\mathcal{V}$ are at least at distance two in the graph.

The time to sort the list of nodes is $O(|V| \cdot log(|V|))$ and in space $O(|V|)$. We can easily prove that the **for** loop time and space complexity is $O(|V| + |E|)$. The overall cost of set $\mathcal{V}$ computation is in time $O(|V| \cdot log(|V|) + |E|)$ and in space $O(|V| + |E|)$.

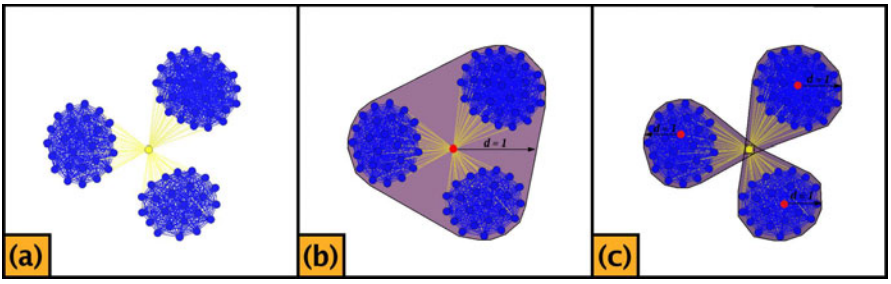



**Fig. 2** (a) Subgraph of the "Hollywood graph" or actors graph where vertices represent actors and two vertices are linked if the corresponding actors were involved in a common movie. Color of each vertex corresponds to its strength value, from the lowest in yellow to the highest in blue. (b) If the red vertex is chosen as a community "centre", we then could obtain one unique cluster containing the whole network. (c) If the community "centres" have high strength values, then we obtain 3 clusters corresponding to the 3 movies of that network

**Algorithm 1.** Extraction of the set $\mathcal{V}$. The sortNodeWithStrength($G$, *sorted_nodes*) method sorts the vertices of $G$ by decreasing Stength values and store the result in *sorted_nodes*.

```
Input: A graph G = (V,E)
Output: A maximal set V of vertices at distance at least 2
vector<node> sorted_nodes;
sortNodeWithStrength(G, sorted_nodes);
for unsigned int i from 0 to (number of vertices in G) do
    node u = sorted_nodes[i];
    if u in G then
        append(V,u);
        foreach node v in neighbourhood of u do
            remove(G, v);
        end
        remove(G, u);
    end
end
```

## 3.3 Group Extraction

Algorithm 2 allows to extract clusters based on set $\mathcal{V}$. In this algorithm, we build "spheres of radius 1" in the graph around nodes of $\mathcal{V}$. For each node $u$ in $\mathcal{V}$, every edge $(u,v)$ with a Strength value greater than a fixed threshold $\varepsilon$ is added to the community of $u$. To compute the threshold, we assume that the sparser the network is, the sparser communities are. Thus, the threshold $\varepsilon$ is computed according to the number of edges of graph $G = (V,E)$ and to the maximum number of edges of the complete graph $K_{|V|}$ with $|V|$ nodes. In algorithm 2, the threshold $\varepsilon$ was

determinated empirically and can be modified in order to build clusters that are more or less tolerant to noise.

**Algorithm 2.** Building groups of vertices

```
Input: A graph G = (V,E), the Strength of each edge, a maximal set V
Output: A set D of groups of vertices
double ε = 2·|E|/(|V|·(|V|−1));
foreach node u in V do
    Group curGroup = createNewGroup();
    append(curGroup, u);
    foreach edge e = (u,v) adjacent to u do
        if Strength(e) > ε then
            append(curGroup, v);
        end
    end
    append(D, curGroup);
end
```

For each node $u$ in $V$, algorithm 2 goes through edges incident to $u$ and runs in $O(deg(u))$. The overall time complexity is $O(\sum_{u \in V} deg(u))$. Given that $\sum_{u \in V} deg(u) = 2 \cdot |E|$, algorithm 2 has a time complexity of $O(|E|)$ and a space complexity of $O(|V| + |E|)$.

Finally, we can conclude that the overall complexity of the decomposition algorithm is $O(|E| \cdot deg^2_{max} + |V| \cdot log(|V|))$ and in space $O(|V| + |E|)$.

## 4 Algorithm Application

### *4.1 Material*

We chose as a case study a subset of the dataset used in InfoVis 2007 Constest (InfoVis 2007 Contest, 2007). This dataset comes from the well known IMDb (Internet Movie Database). We extracted a set of 432 movies involving 4025 actors and built a graph as follows: a node in the graph is an actor and two nodes are connected by an edge if the corresponding actors were involved in at least one common movie. The resulting network contains 4025 nodes and 41216 edges (see figure 3). This benchmark is particularly well suited for our study since it contains many communities. In fact, every movie is a clique connecting all the actors who played in this movie, and cliques share nodes when an actor played in several movies.

We evaluate the quality of our decomposition algorithm in two different ways. First, we use a generalisation of the *MQ* measure introduced by Mancoridis *et al.* (1998). This generalisation takes into account the cases where nodes can belong to several clusters. Secondly, we measure the sensitivity of the decomposition to structural changes of the network during the dynamic process. To evaluate this level

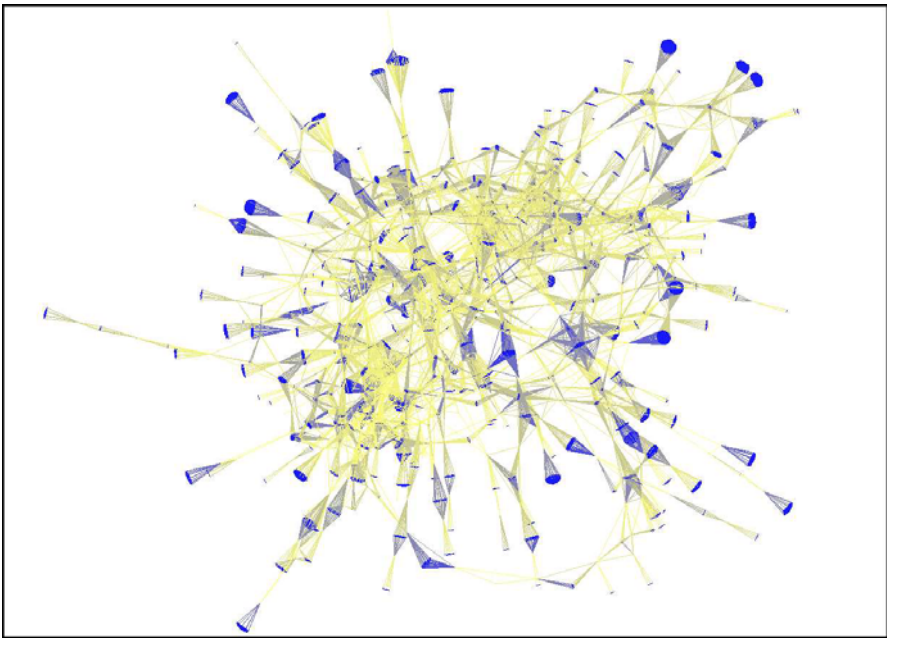


**Fig. 3** Subgraph of the "Hollywood graph". This subgraph corresponds to a set of 432 movies and contains 4025 vertices (actors) and 41216 edges. Color of each vertex corresponds to its strength value (from the lowest values in yellow to the highest in blue).

of structural stability, we compare the decomposition obtained on the original graph to a decomposition obtained after several structural modifications to the network.

## *4.2 Decomposition Quality*

Widely accepted definitions of a "good" decomposition of a graph is a high intra-cluster density and a low inter-cluster density. This can be evaluated using a generalisation of the $MQ$ measure introduced by Mancoridis *et al.* (1998) that takes into account overlapping clusters. This generalisation had been presented by Bourqui and Auber (2008). Considering a graph $G = (V,E)$ and a decomposition $C = \{C_1, C_2, ..., C_k\}$ of nodes in $G$. $MQ_{Over}$ is defined as follows:

$$MQ_{Over} = MQ^{+} - MQ^{-}_{Over} \tag{4}$$

Where

$$MQ^{+} = \frac{1}{k}\sum_{i} s(C_i, C_i) \tag{5}$$

And

$$MQ^{-}_{Over} = \frac{1}{k(k-1)} \sum_{i} \sum_{j \neq i} s_{Over}(C_i, C_j) \tag{6}$$

Where $s_{Over}(C_i, C_j) = \frac{|E(C_i, C_{j \setminus i})|}{|C_i| \cdot |C_{j \setminus i}|}$ and $C_{j \setminus i} = C_j \setminus (C_j \cap C_i)$. In this equation, $MQ^{+}$ models the internal cohesion of clusters $C_1, ..., C_k$ while $MQ^{-}_{Over}$ models the external cohesion of clusters.

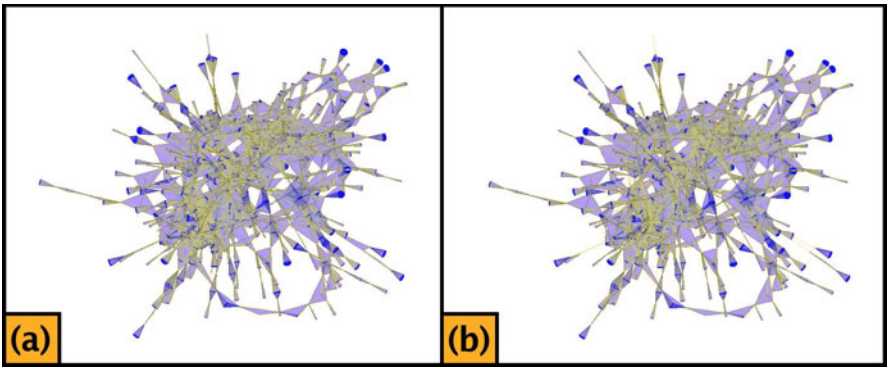



**Fig. 4** Optimal decomposition (a) and the result of our algorithm (b) of the graph from Figure 3. Each group is surrounded by a purple convex hull.

We applied our decomposition algorithm to the sub-network of the movie actor network (see Figure 3). Figure 4.(b) shows the result obtained on this graph. In this Figure, each cluster found by our algorithm is surrounded by a purple convex hull. The value of $MQ_{Over}$ is 0.95 showing that our algorithm gives excellent results according to this measure. Figure 4.(a) shows what we consider as the optimal decomposition (i.e. every cluster represents a movie). We can first visually note the similarity of these two decompositions. Upon further investigations, it appears that our algorithm finds 421 clusters of which 404 match perfectly the optimal decomposition. Our algorithm thus found 93% of the optimal decomposition (and 96% of the clusters found correspond to movies). Each of the 17 clusters obtained by our algorithm that do not fit the optimal solution are in within a cluster of the optimal decomposition.

## *4.3 Sensitivity to Modifications*

In order to evaluate the sensitivity of our algorithm to structural changes in the network, we compare the decomposition of the original network (reference decomposition) to the decompositions obtained after a given number of random additions/removal of edges. In this section, we explain the similarity measure used to compare the two decompositions. Then, we present the results of our case study.

### 4.3.1 Similarity Measure

To measure the similarity between two decompositions, we use a metric inspired by Brohée and van Helden (2006) which is based on *representativeness*. Two decompositions are considered similar if and only if the first one is representative of the second and vice-versa.

To define the representativeness of two decompositions, we first have to define representativeness of two clusters. Let $c_i$ and $c_j$ be two clusters, we say that the cluster $c_i$ is representative of cluster $c_j$ if and only if $c_i$ contains a large number of elements of $c_j$. Formally, the *directed cluster representativeness* is defined as follows:

$$\rho_{c_i \to c_j} = \frac{|c_i \cap c_j|}{|c_j|} \qquad \rho_{c_j \to c_i} = \frac{|c_i \cap c_j|}{|c_i|}$$

And the *undirected cluster representativeness* is defined as:

$$\rho_{c_i c_j} = \sqrt{\rho_{c_i \to c_j} \cdot \rho_{c_j \to c_i}}$$

This measure corresponds to the geometric mean of directed representativeness of clusters $c_i$ and $c_j$.

We can, in a similar way, define the degree of representativeness of a decomposition with regards to another one. Let us consider two decompositions $C$ and $C'$, we say that $C$ is representative of $C'$ if and only if for each cluster $c'$ of decomposition $C'$, the decomposition $C$ contains a representative cluster of $c'$. Given that clusters of "small" size tend to bias this metric, we gives a higher representativeness to large clusters. We define the *directed clustering representativeness* as follows:

$$\sigma_{C \to C'} = \frac{\sum_{c_i \in C'} \max_{c_j \in C} \rho_{c_j c_i} |c_i|}{\prod_{c_i \in C'} |c_i|}$$

This formula corresponds to the weighted average of best representativeness of each cluster in $C'$ by clusters in $C$.

We can then define the *undirected clustering representativeness* as follows:

$$\sigma_{CC'} = \sqrt{\sigma_{C \to C'} \cdot \sigma_{C' \to C}}$$

A possible modification of this metric could be obtained by using a simple product instead of the geometric mean during the computations of the undirected representativeness. It better distinguishes similar decompositions from different ones, as "bad" associations are more heavily penalised. In the next section, we use this modified version of the similarity metric.

### 4.3.2 Experimental results

To measure the sensitivity of our decomposition algorithm, we first generate a collection of 100000 graphs from the graph shown on figure 3 using algorithm 3.

**Algorithm 3.** Generation of the dataset used to evaluate the stability of our decomposition algorithm. The getOperation() function returns 'edge addition' with a probability 0.5, 'edge deletion' otherwise. The constants $NB\_TESTS$ and $MAX\_OPERATIONS$ were respectively set to 50 and 2000.

```
Input: subgraph G = (V,E) of the Hollywood graph
Output: A set Collection of graphs
for unsigned int i = 0 to i == NB_TESTS do
    Graph H = G;
    for unsigned int j = 0 to j == MAX_OPERATIONS do
        Operation op = getOperation();
        if op == 'edge deletion' then
            node src = getRandomNode();
            node tgt = getRandomNode();
            edge e = edge(src, tgt);
            while e is not element of H do
                src = getRandomNode();
                tgt = getRandomNode();
                e = edge(src, tgt);
            end
            deleteEdge(H, e);
        end
        else                                   /* op == 'edge addition' */
            node src = getRandomNode();
            node tgt = getRandomNode();
            edge e = edge(src, tgt);
            while e is element of H do
                src = getRandomNode();
                tgt = getRandomNode();
                e = edge(src, tgt);
            end
            addEdge(H, e);
        append(Collection, H);
    end
end
```

We then compare the decomposition obtained on the original graph to those obtained on the graphs of the generated sample collection. Figure 5 shows our results.

In Figure 5.(a), the blue line shows the average number of perfect cluster matches according to the number of edges removed or added to the original network. The standard deviation is depicted in red for each of these average values. We see on this plot that for up to 2000 modifications of the graph, our algorithm finds on average

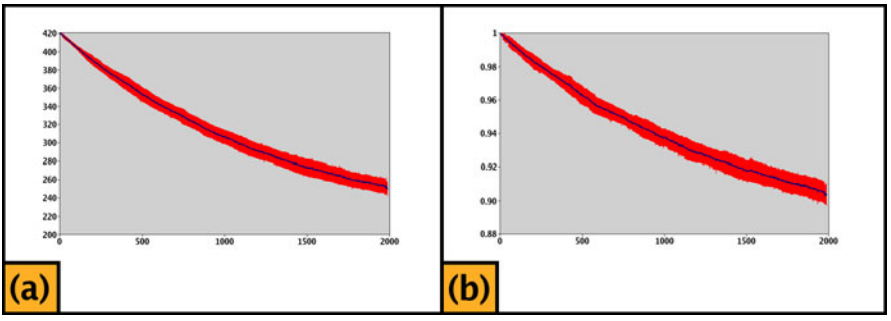



**Fig. 5** (a) Average number of perfect matching (in blue) according to the number of addition or deletion operations previously done and the corresponding standard deviation (in red). (b) Average value of the similarity metric (in blue) according the number of addition or deletion operations previously done and the corresponding standard deviation (in red).

between 250 and 421 perfect clusters when compared to the original decomposition. Moreover we can see that the standard deviations are relatively low, between 0.44 and 10.34. In Figure 5.(b), the blue line shows the average value of the similarity metric according the number of edges removed or added to the original network. The red line shows the standard deviation. Average values of the similarity metric remain between 0.9 and 1. This interval is very good in terms of similarity, and it shows standard deviation values between 0.0002 and 0.007.

Considering thereafter the naive sensitivity measure of computing the percentage of perfect matching, our algorithm preserves on average 78% of the clusters. Moreover, average values of this similarity measure are also high, showing the stability of our decomposition algorithm.

## 5 Conclusion

In this article, we present a new algorithm for the analysis of dynamic graphs as the main step of a framework which detects topological changes. This method is based on the transformation of the dynamic graph into a set of static graphs and on the graph decomposition in potentially overlapping clusters. Our main assumption is that is if the structure of the network does not change drastically in the dynamic process, then decomposition obtained on two consecutive graphs should contain a similar community structures.

We show in this article that our algorithm is stable with respect to minor changes in the network. Given that in our approach two similar graphs should have similar decompositions, our algorithm detects major changes in the network. In addition, our algorithm offers a $O(|E| \cdot deg_{max}^2 + |V| \cdot log(|V|))$ time complexity.

Finally, we give a generalisation of the Brohée and van Helden (2006) similarity measure for overlapping decompositions. This new measure allows us to compare

the decompositions of two graphs corresponding to two consecutive time frames to detect high impact structural changes in the network.

## References

Auber, D., Chiricota, Y., Jourdan, F., Melançon, G.: Multiscale Visualization of Small-World Networks. In: Proc. of IEEE Information Visualization Symposium, pp. 75–81 (2003)

Bader, G.D., Hogue, C.W.: An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics 4 (2003)

Bourqui, R., Auber, D.: Analysis of 4-connected components decomposition for graph visualization. Technical report, LaBRI (2008), http://www.labri.fr/

Brohée, S., van Helden, J.: Evaluation of clustering algorithms for protein-protein interaction networks. BMC Bioinformatics 7(488) (2006), http://www.biomedcentral.com/1471-2105/7/488

Chiricota, Y., Jourdan, F., Melançon, G.: Software Components Capture using Graph Clustering. In: 11th IEEE Int. Workshop on Program Comprehension (2003)

Gajer, P., Kobourov, S.G.: GRIP: Graph dRawing with Intelligent Placement. In: Marks, J. (ed.) GD 2000. LNCS, vol. 1984, pp. 222–228. Springer, Heidelberg (2000)

InfoVis 2007 Contest, IEEE InfoVis 2007 Contest: InfoVis goes to the movies (2007), http://www.apl.jhu.edu/Misc/Visualization/

Mancoridis, S., Mitchell, B.S., Rorres, C., Chen, Y., Gansner, E.R.: Using Automatic Clustering to Produce High-Level System Organizations of Source Code. In: IEEE Proc. Int. Workshop on Program Understanding (IWPC 1998, pp. 45–53 (1998)

Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E 69, 026113 (2004)

Palla, G., Barabasi, A.-L., Vicsek, T.: Quantifying social group evolution. Nature 446, 664–667 (2007)

Suderman, M., Hallett, M.: Tools for Visually Exploring Biological Networks. Bioinformatics Advanced Access (in press) (2007), doi10.1093/bioinformatics/btm401

# Part III
# Security and Data Streaming

# An Hybrid Data Stream Summarizing Approach by Sampling and Clustering

Nesrine Gabsi, Fabrice Clérot, and Georges Hébrail

**Abstract.** Computer systems generate a large amount of data that, in terms of space and time, is very expensive - even impossible - to store. Besides this, many applications need to keep an historical view of such data in order to provide historical aggregated information, perform data mining tasks or detect anomalous behavior in computer systems. One solution is to treat the data as streams being processed on the fly in order to build historical summaries. Many data summarizing techniques have already been developed such as sampling, clustering, histograms, etc. Some of them have been extended to be applied directly to data streams. This chapter presents a new approach to build such historical summaries of data streams. It is based on a combination of two existing algorithms: StreamSamp and CluStream. The combination takes advantages of the benefits of each algorithm and avoids their drawbacks. Some experiments are presented both on real and synthetic data. These experiments show that the new approach gives better results than using any one of the two mentioned algorithms.

**Keywords:** Data Streams, Non-specialized Summary, Sampling, Clustering.

Nesrine Gabsi
Institut TELECOM, TELECOM ParisTech, 46 Rue Barrault 75013 Paris,
and France Télécom RD, 2, avenue P.Marzin 22307 Lannion
e-mail: gabsi@enst.fr

Fabrice Clérot
France Télécom RD, 2, avenue P.Marzin 22307 Lannion
e-mail: fabrice.clerot@orange-ftgroup.com

Georges Hébrail
Institut TELECOM, TELECOM ParisTech,
Partially Suported by ANR (MIDAS Project ANR-07-MDO-008),
46 Rue Barrault 75013 Paris
e-mail: hebrail@enst.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 181–200.
springerlink.com 

## 1 Introduction

Nowadays, several modern applications generate large amounts of data in a continuous and unbounded fashion. Examples of such applications include network-monitoring applications, financial monitoring and many others. Storing, querying and mining of such data are highly computational challenging tasks. It is very costly to store all data before the analysis process. One solution is to adopt a real time processing called *Data Stream processing*. Golab and Özsu (2003) define a data stream as a continuous sequence of ordered elements, arriving in the real time with important rates. It is impossible to control the order in which elements arrive, neither is it feasible to locally store a stream in its entirety . It is necessary to process these data 'on the fly'.

Much works (Babcock *et al.*, 2002; Golab and Özsu, 2003; Ma *et al.*, 2007) show that it is not feasible to simply load the arriving data into a traditional database management system (DBMS). They are not designed to load large amounts of continuous data. Conventional DBMS are ill-equipped to fulfill the needs of applications. Therefore, a new class of systems 'the Data Stream Management Systems' (DSMS) are being developed by the database community to satisfy the requirements of stream based applications. Several academic and commercial DSMS, such as Stream (Arasu *et al.*, 2003), Aurora (Abadi *et al.*, 2003), have been designed to handle transient data streams on-line.

Moreover, DSMS enable users and applications to issue continuous queries that are evaluated over data streams or over windows (finite portion of stream) (Babcock *et al.*, 2002). Since data arrive continuously, these queries tend to be long running. To be processed in real time, queries must be specified before the beginning of streams. However, in many applications, there is a need to query any portion of the stream without specifying in advance what to analyze. In this case, the system will not respond to these queries as all the data have not been previously saved. It is therefore necessary to keep an historical summary of the stream. Many studies have been carried out on specialized summary techniques (Gemulla and Lehner, 2008; Guha *et al.*, 2001; Flajolet and Martin, 1985), which are structures dedicated to a particular task. The proposed approach focuses on non-specialized summaries, which aim at providing approximate results for any analysis tasks (e.g. answering queries or applying mining algorithms). These summaries must cover the whole stream and enable to run queries over any past part of the history of the stream. To the best of our knowledge, few works have approached this issue, with the exception of those of Csernel *et al.* (2006) and Aggarwal *et al.* (2003).

In this chapter, we focus on the long-term preservation of data streams. Our goal is to develop a data summary which is non-specialized (operational on a large set of applications), adaptable to stream changes, and of good quality (representative of the original stream) throughout a large period of time. The new approach builds such historical summaries of data streams and offers a good compromise between the two existing algorithms (StreamSamp and CluStream) both in terms of accuracy and run-time. The combination takes advantages of the benefits of each algorithm and avoids their drawbacks. Given that we use CluStream (Aggarwal *et al.*, 2003),

only the case of quantitative data can be considered. It is possible to run queries on the constructed summary, and get answers that approximate, as much as possible, the case where all data are used.

The chapter is organized as follows: in section 2, we discuss the different kinds and algorithms for data stream summaries; in section 3, we discuss our new approach called 'Hybrid approach' for summarizing data streams; section 4 reports the performances studied on real and synthetic data sets; we conclude in Section 5 with mentioning some directions for further work.

## 2 Data Stream Summaries

In recent years many summary structures have been developed which can be used with various mining and query processing techniques. Some structures are devoted to a particular type of treatment i.e. counting distinct elements (Flajolet and Martin, 1985), checking over the existence of an element in a set (Bloom, 1970). The choice of a specialized summary method depends on the needs and the problems to be solved. In general, we aim at building summary structures that have wide applicability across a broad class of problems. Those are non-specialized summaries, the focus of this chapter. We describe in section 2.1 some data stream summarizing techniques while section 2.2 and 2.3 are focused on describing algorithms used in our Hybrid approach.

### *2.1 Non-specialized Summary*

In an ideal context, a non-specialized summary has to provide approximate results for any analysis that we wish make on the original data. According to Csernel (Février 2008) this summary must meet four conditions: (1) responding to queries related to any fixed time horizon, (2) treating a wide range of queries (selection, median, etc.), (3) allowing monitored analysis of data like decision trees and (4) authorizing exploratory analysis tasks such as clustering. Queries may be addressed at the present and related to past data. A non-specialized summary can be defined by reducing the memory space, we call this *memory organization* or by temporal dimension, we call this *temporal dimension organization* as explained below.

#### 2.1.1 Memory Organization

These techniques conserve summaries either of the entire data stream or that focus on recent observations. There are two major techniques which can be used for summary construction: sampling and histograms.

**Sampling Methods.** These methods are among the simplest methods for summary construction in data streams. They are directly outcome from statistics aiming at providing information by a representative sample from a large population. Moreover, it is relatively easy to use these summaries with a wide variety of applications

since their representation is not specialized and uses the original representation of the elements. Usually these techniques need to have access to all elements in order to select a representative sample. This constraint makes it difficult to use in data streams context given the unbounded feature of streams. To overcome this problem, sequential sampling algorithms have been developed like reservoir sampling methods.

Reservoir based methods (Vitter, 1985; Al-Kateb *et al.*, 2007; Park *et al.*, 2004) are widely used to sample data streams. They allow the incremental construction of uniform, random and pre-defined size samples without knowing in advance the number of elements in the stream. Elements are added to the reservoir with probability $n/(t+1)$ where $n$ corresponds to the reservoir's length and $t$ is the index of the element to be inserted. The drawback of this technique is that it disfavors recent elements. More sophisticated techniques have been developed; they maintain a random sample on a sliding window (Aggarwal, 2006).

**Histograms.** Another method for data summarization is histogram construction. We consider here an histogram as a discrete representation of the distribution of both qualitative or quantitative data. In this method, data values are divided into classes or groups (called buckets) and the frequency of each bucket is stored. Thus, storage space is represented by the number of buckets in the histogram. This technique allows to keep more information on the observed data but with a larger memory space and sometimes greater complexity in the histogram update. However, in the histogram we loose the distribution of elements within a bucket; this is a source of inaccuracy. This technique is also employed for a multitude of tasks such as approximate query answering and data mining (Ioannidis and Poosala, 1999).

In the literature, several types of histograms have been proposed: V-Optimal Histogram, Equi-Width Histograms and End-Biased Histograms (Muthukrishnan *et al.*, 2005). These techniques are adapted in the context of data streams (Guha *et al.*, 2001; Puttagunta and Kalpakis, 2005). For example, in the case of a V-Optimal Histogram, Jagadish *et al.* (1998) showed how to compute such histograms with $O(N)$ space requirements and $O(N^2B)$ time, where $N$ is the size of the data set and $B$ is the number of buckets. This algorithm cannot be applied in the context of data streams because of its complexity. Guha and Harb (2005) extend this algorithm in the context of data streams. The proposed approach uses $O(B^2logN)$ space and $O(B^2logN)$ time per data element to construct a V-Optimal Histogram.

The methods described above do not keep informations describing the whole data stream. Only the recent past is considered. However, the aim of our approach is to answer queries about the distant past, expressed at a recent time. For that, we have to combine these methods with a temporal approach.

### 2.1.2 Temporal Dimension Organization

These techniques keep summaries which cover the full stream by using bounded space in a particular kind of windowing system. These windows have a variable size that increases with time. The idea is that the most recent elements are the most

interesting. The most recent time is kept at the finest granularity and the more distant one is registered at a coarser granularity. There are two main techniques for temporal dimension organization: the tilted time frame model and the progressive logarithmic tilted time frame model (Snapshot system) (Aggarwal, 2007).

**Tilted Time Frame Model.** There are two kind of tilted window models: the natural tilted time frame model and the logarithmic tilted time model. In the first model, the time frame is structured in multiple granularity based on natural time scale. As shown in Fig 1, in this model, we can compute an aggregate in the last quarter with the precision of a minute, or in the last hour with the precision of a quarter, and so on until the whole year with the precision of a month. In the second one, the time frame is structured in multiple granularity according to the logarithmic scale.

We can use these structures to maintain summaries covering time periods of varying sizes. Summaries have a set size but spread over time periods of varying length, shorter for the present and longer for the distant past.

**Progressive Logarithmic Tilted Time Frame Model.** As in the tilted time frame, this model allows an efficient processing of temporal dimension organization. The goal is to take different snapshots describing the system state (the system state can be characterized by some statistics, clusters position, etc.). Each snapshot is represented by its timestamp. Snapshots are stored at different levels of granularity in a pyramid-shaped structure. This structure favors recent time frames to older ones. Depending on time, the number of stored snapshots is increasingly weak. The Fig 2 illustrates an example of progressive tilted time frame model. In this example, we suppose that there are 5 frames and each takes maximal 3 snapshots. Given a snapshot number N, if N mod $2^a = 0$, then we insert the snapshot into the frame number $a$. If there are more than 3 snapshots, we delete the oldest snapshot.

These tilted time models ensure that the total amount of data to retain in memory is small. They provide a natural way to control the incremental data insertions in new frames.


**Fig. 1** Natural tilted time frame window

| Frame no. | Snapshots (by clock time) |
|---|---|
| 0 | 69 67 65 |
| 1 | 70 66 62 |
| 2 | 68 60 52 |
| 3 | 56 40 24 |
| 4 | 48 16 |
| 5 | 64 32 |

**Fig. 2** Progressive tilted time frame window

## 2.2 CluStream: A Clustering Approach

Recently, the clustering problem has also been studied in the context of data streams (Aggarwal *et al.*, 2003; Zhang *et al.*, 1996). They use a micro-clustering based stream mining framework which is designed to capture summary statistics about the stream. This summary information is defined by two structures: the micro-clusters and the pyramidal time frame. The first one allows to maintain statistical information about the data locality in terms of micro-clusters, while the second structure stores the micro-clusters at snapshots in time which follows a logarithmic time frame pattern. In this technique, the snapshots are stored at different levels of granularity depending on their age.

We discuss in this chapter the CluStream algorithm. It is essentially based on a clustering of numerical data. However, it provides a structure particularly adapted to summaries of data streams. Aggarwal has adapted the Cluster Feature Vector (CFV) structure to the context of data streams. This structure was already used by Zhang *et al.* (1996) for large databases. The CFV structure maintains statistics about elements in a micro-cluster. It contains the number of elements of a given micro-cluster and, for each attribute, the sum of its values as well as the sum of all their values squared. In Aggarwal *et al.* (2003), Aggarwal added to the CFVs the temporal attribute of each element.

CluStream proceeds in three steps: the first one is the initialization step in which it uses k-means algorithm to create the first $q$ micro-clusters. The second step manages the evolution of micro-clusters. It is an on-line phase and proceeds as follows: when a new stream element arrives, CluStream assigns it to the nearest cluster. For that, its distance from the centroid of each micro-cluster is calculated. Then, the CFV of this cluster is updated accordingly without storing the belonging of this element to the cluster. However, if no micro-cluster is found close enough to the new element, a new one is created. It will contain only that element. As a rule, a new cluster is created if the stream element is either an outlier or it corresponds to the beginning of a new cluster because of evolution of the stream. In order to create this new micro-cluster and preserving a constant number of micro clusters, the oldest cluster is destroyed or two older micro-clusters are merged. The algorithm keeps the elements of the identifiers of merged clusters. While the above process of updating is executed at the arrival of each stream element, another process is executed at each clock time.

At each clock time, the algorithm takes snapshots. It consists of storing on disk the CFV of all micro-clusters. Those snapshots are then saved according to a pyramidal time frame. They are classified into different orders which can vary from 1 to $log(T)$, where $T$ is the clock time elapsed since the beginning of the stream. To free memory space, the least recent snapshot are also deleted.

The third step is an off-line phase and is characterized by post-analysis which can be applied to the stored snapshots. The mathematical features of CFVs make possible to follow micro-clusters evolution. Information stored about clusters, such as timestamps and elements of the identifiers of merged clusters, allows implementing the subtraction of two snapshots. Thus, one can determine the approximate

micro-clusters for a pre-specified time horizon. This can provide a version of the summary for different time horizons.

On the basis of a clustering technique coupled with a pyramidal time frame structure, CluStream keeps representative snapshots even for old stream elements. This allows the monitoring of the data stream over time. However, one weakness of the algorithm is that the process of distance calculations is expensive. A second limitation concerns the high number of parameters which depend on the nature of the stream and the arrival speed of elements.

### *2.3 StreamSamp*

StreamSamp is an algorithm based on random sampling of data streams (Csernel *et al.*, 2006). Upon arrival, the stream elements are sampled in a purely random way with a fixed sampling rate $\alpha$ and placed in a sample. When this sample reaches a given size $T$, StreamSamp stores it with the dates marking its starting and ending point. The order 0 is associated with this sample. Because of the boundlessness of the stream, it is impossible to permanently store all the samples created. To reduce space, the StreamSamp summary structure is based on the tilted time frame model where samples cover periods of time of varying length.

When the number of samples of a given order $i$ reaches a given limit $L$, the two oldest samples of this order are merged into a single sample of size $T$ and it receives the order $i+1$. This new sample has the same size $T$, but it covers a time period twice longer. It is built by randomly keeping $T/2$ elements of each of its parent samples.

Moreover, StreamSamp allows the exploitation and analysis of the created summary. To process a time period, the samples belonging to this period are concatenated. Thus, a sample on any given part of the stream or the whole stream can be created. If samples of different orders have to be concatenated, they must be weighted differently (by giving a weight of $2^i$ to all the elements of a sample of order $i$) so as to keep the same representativity for each element of the final sample.

The summary created by StreamSamp has the particularity of being small and quickly designed. It does not depend on the speed of the stream. However, the quality of the summary produced for old time periods is likely to deteriorate. Indeed, old elements have an increasing weight for a constantly fixed sample size. Therefore, if a sample contains recent elements (much lower weight) and some old elements, the latter will increase the errors in the results of query answers. The use of tilted time frame model does not favor (in terms of accuracy) another period except the recent past.

## 3 An Hybrid Approach of Non-specialized Summaries

The work presented in this chapter is based on combining a sampling and a clustering approach. We propose a new algorithm for improving quality (in terms of accuracy and representativeness) of non-specialized summaries. This new approach

combines the benefits of StreamSamp and CluStream while avoiding their disadvantages. Since the new approach involves CluStream, only quantitative data can be considered.

Stream elements are first sent into the StreamSamp process which keeps random samples. When the samples are no longer representative in term of some criteria detailed below (section 3.1), the sample's elements are sent to the CluStream process.

To meet CluStream timeline, samples must be sent in order: samples with higher orders must be moved away before the lower order samples. The samples inclusion into CluStream must be done element by element. The insertion of elements depends on their weights. Two strategies can be applied: (i) Each element is inserted $w$ times ($w$ represents the weight of the corresponding sample), (ii) each element is multiplied by its weight and inserted once in the corresponding micro-cluster. The second technique is adopted because it reduces the complexity of the algorithm. The transition from one process to another is not a random procedure but has to respect two criteria. The transition criteria that is defined below allows us to determine when a sample is still representative.

## *3.1 Transition Criteria*

The choice of the transition criteria is based on inherent features of the two processes: StreamSamp is controlled by random sampling while CluStream is guided by updating evolutionary micro-clusters. In order to preserve a summary with good quality designed by our Hybrid approach, we have to maintain a good quality for these two processes. The first criterion called *variance criterion* monitors the random re-sampling which leads to a degradation of the StreamSamp summary. The second criterion is based on the preservation of centroid positions which is a CluStream representation mode. The main idea is to transit from one process to another based on a simultaneous checking of these two criteria. The first criterion is checked for each attribute while the second one is checked considering all attributes together.

### 3.1.1 Variance Criterion

The variance criterion aims to measure the 'quality' of samples resulting from StreamSamp. As explained in section 2.3, StreamSamp involves re-sampling and hence deteriorates the quality of summary. Indeed, the number of elements at a given time period is reduced by half at each re-sampling step. The quality of a sample is measured by estimating the relative error of aggregates such as mean, median, variance, etc. We consider in this paper only the mean aggregate for error estimating.

To check the quality of StreamSamp summaries, we consider the merger $(E_1 \cup E_2)$ of two independent samples $E_1$ and $E_2$ having the same weight and we compute the mean estimator $\widehat{\overline{x}}$. Since a simple and random sampling is used, we know that with a confidence of 95%, we have the inequality

$$\left|\overline{x} - \widehat{\overline{x}}(E_1 \cup E_2)\right| \leq Student(0.05)\sqrt{Var(\widehat{\overline{x}}(E_1 \cup E_2))}$$

Where $E_1$ and $E_2$: Two samples that have to be merged, $Student(0.05) = 1.96$, $\widehat{\overline{x}}$: the estimator of the mean and $\overline{x}$: the real mean.

The variance $Var(\widehat{\overline{x}}(E_1 \cup E_2))$ is estimated using the following formula:

$$Var(\widehat{\overline{x}}(E_1 \cup E_2)) = (1 - \frac{2n}{N})(\frac{1}{2n})[\frac{1}{2n-1} \sum_{k \in E_1 \cup E_2} (x_k - \overline{x})^2]$$

where $n$: sample size, $N$: size of the involved population

To ensure a satisfied quality for a merger sample, we define a bound $err$ that error estimator must not exceed. The criterion is expressed using the following inequality:

$$\frac{1.96\sqrt{Var(\widehat{\overline{x}}(E_1 \cup E_2))}}{\widehat{\overline{x}}(E_1 \cup E_2)} \leq err$$

However, even if the criterion is met, we do not decide to merge unless the second criterion about centroids's position is checked.

### 3.1.2 Centroids' Position Criterion

In our transition approach, the life cycle of a sample is composed by two complementary phases. The first step is the evolution of the sample in StreamSamp . The second is the insertion of the sample elements in the CluStream process. It is therefore important to take into consideration the constraints related to each process. In the previous section, we have defined the constraint based on the StreamSamp process (variance criterion). Within the same logic, we have to control the centroids' behaviors when inserting elements in CluStream. In fact, CluStream brings together the "closest" elements into the same micro-cluster. Unlike the classical approach of CluStream in which the algorithm processes the whole stream, in our CluStream version, the algorithm will only process a sampled stream.

The random re-sampling process leads to a deterioration of the built summary quality. This fact may cause a considerably change of the centroids' positions which will be calculated on the remaining samples. Consequently, the precision on the centroids' position will deteriorate. We aim at maintaining a minimum precision on the centroids' position. We have therefore added a second criterion which is based on the conservation of centroids' position.

In order to maintain this precision, we calculate at each re-sampling process, the distance between the centroid ($G$) (calculated from the samples to be merged ($E_1 \cup E_2$)) and the centroid ($\overline{G}$) (calculated from the estimated sample ($E_3$)). As shown in fig. 3, the distance between ($G$) and ($\overline{G}$) must be below a threshold $D$. Otherwise, the required precision is no longer respected.

$$D = \varepsilon \times \sum_{E_1 \cup E_2} (d^2(\overline{x}, x_i))$$

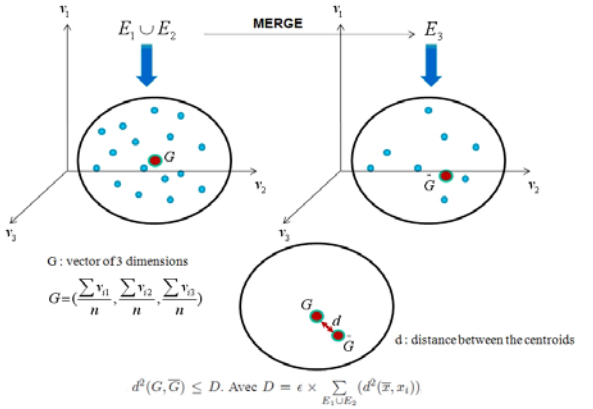



**Fig. 3** Distance computation between the centroids $G$ and $\overline{G}$

**Algorithm 1.** Hybrid approach algorithm

**Require:** $E1, E2$ : Samples to merge, $n$ : Sample size, $w$ : weight of $E_1$ and $E_2$
$err$ : Threshold for the variance criterion
$\varepsilon$ : Threshold for the centroids' position criterion
$Test$: Boolean variable
$Test \Leftarrow True$
**for** each attribute $i$ **do**
  **if** $Variance\_Criterion(i) == False$ **then**
    $Test \Leftarrow False$
  **end if**
**end for**
**if** $Test == True$ AND $Centroid\_Criterion() == TRUE$ **then**
  $mergeSamples(E_1, E_2)$
**else**
  $Max \Leftarrow Max(sampleWeights)$
  **for** $i = Max$ to $w$ **do**
    $MovetoClustream(E_i)$ {moving sample having $i$ as weight}
  **end for**
**end if**

where $\varepsilon$ is a user defined parameter fixed following the centroids' evolution, and $\sum_{E_1 \cup E_2} (d^2(\overline{x}, x_i))$ the intra-cluster inertia of the sample made up from $(E_1 \cup E_2)$.

### 3.1.3 Conclusion

As shown in Algorithm 1, if one of these two conditions (variance criterion and centroids' position criterion) is no longer respected, the sampling process is stopped and replaced by CluStream. Thus the corresponding sample elements move to CluStream.

# 4 Empirical Results

Series of tests have been conducted to evaluate our Hybrid approach. We aim at assessing the performance of our algorithm and comparing it with CluStream and StreamSamp used alone.

## *4.1 Environment of Tests and Data Sets*

All of our experiments have been run on a PC with an Intel Core 2 Duo 1.66 Ghz computer processor and 1000 MB memory which runs Windows XP professional operating system. All algorithms are coded in the Java programming language. For testing the effectiveness of our algorithm, we have compared it to CluStream and StreamSamp. The Hybrid approach is implemented according to the description in this chapter while CluStream and StreamSamp are done according to Aggarwal *et al.* (2003) and Csernel (Février 2008). To make the comparison fair, the three algorithms use the same amount of memory to store their summaries. To test the accuracy of estimating the mean query, median and the runtime evaluation a synthetic

**Algorithm 2.** MovetoClustream function

```
Require: E : Samples to be sent to CluStream
  x_i : Elements of sample E
  w : weight of sample E
  S : The current set of micro-clusters
  nClusters : number of micro-clusters
  S = {}
  for i = 0 to n do
      x_i = x_i * w
      if S is empty then
          createMicroCluster(x_i) {create a micro-cluster containing the singleton
          x_i}
      else
          computeDistance(x_i,M) {M is the closest micro-cluster to x_i}
          if x_i is inside Boundary then
              addStatistics(x_i,M)
          else
              createMicroCluster(x_i)
              |S| <= |S| + 1
              if |S| ==nClusters+1 then
                  remove(S) {remove the least recently updated
                  micro-cluster from S;}
              end if
          end if
      end if
  end for
```

dataset is mainly be used. However, we use a real dataset to compare algorithms' performance on classification tasks.

**Real dataset.** To test the algorithms' performance on classification tasks, we used a relatively stable dataset *the Forest CoverType*. This is one of the largest databases in the UCI Repository. It contains 581012 elements. The data set is defined by 54 variables of different types: continuous and categorical. Each element belongs to a class from 7 target classes. The elements represent the forest cover type at a 30 $\times$ 30 meter grid, obtained from US Forest Service Region 2 Resource Information System. The goal is to predict the forest cover type from these variables.

**Synthetic dataset.** To test performances of the different algorithms, we generated a synthetic data set containing 10000 elements. Each element is a vector with 3 continuous attributes which follow different distributions (Gaussian, Exponential and Uniform distribution). Table 1 illustrates the different distribution parameters. The parameters of different distributions are chosen in order to have the same mean and standard deviation values. We make older the generated elements to analyze the different distributions' behaviors on evaluating statistics for querying at different ages of the summaries.

In all experiments, we studied the robustness and efficiency of algorithms for estimating queries that grow old over time. In this section, we present the results for querying tasks (median and mean) and data mining tasks (clustering and classification). Furthermore, other kinds of queries can be applied such as Count, Sum, Quantiles, etc.

The algorithm parameters are presented in table 2. We repeated StreamSamp and Hybrid approach 100 times because these two techniques include a sampling step. If they are executed just once, the result will be depend on the drawing. The algorithm parameters $L$ (number of windows over a time period) and $T$ (window size) of StreamSamp were set to ensure more progressive merges.

**Table 1** The attributes used in tests

| Elements | Attribute 1 | Attribute 2 | Attribute 3 |
|---|---|---|---|
| $1 \rightarrow 10000$ | $Unif(-7.32, 27.32)$ | $Exp(1/10)$ | $N(10, 10)$ |

**Table 2** Algorithms parameters for evaluations

| StreamSamp | CluStream | Hybrid approach |
|---|---|---|
| $T = 500$ | # clusters = 75 | $err = 4.10^{-2}$ |
| $\alpha = 100\%$ | k-means = 10 iterations | $\varepsilon = 10^{-3}$ |
| $L = 12$ | # Snapshots per order (L) = 80 | |

## 4.2 Mean Evaluation

We estimate the mean aggregate over the kept summaries (built using StreamSamp, CluStream and the Hybrid approach). We compare the obtained results over the same time period [0-10000] evaluated at different timestamps. The synthetic data set is used to build these summaries.

To estimate this aggregate on the summary generated by StreamSamp, we take all elements having timestamp between 0 and 10000. We multiply each element value $x_i$ by its weight $w_i$ and we divide by the sum of weights ($\frac{\sum x_i w_i}{\sum w_i}$).

For Clustream, two snapshots are kept at the beginning of the stream and at the end of the studied period [0-10000]. The substraction of these snapshots gives all micro-clusters included in this period. The mean aggregate is correctly evaluated. It corresponds to the centroid of all these micro-clusters. To estimate this aggregate, we use the same formula as StreamSamp ($\frac{\sum x_i w_i}{\sum w_i}$). The values correspond to the micro-clusters' centroid and the weights are the number of elements in the micro-cluster.

In the Hybrid approach, we can find elements in the StreamSamp process and in the CluStream process. To calculate the mean, we firstly extract, on the period [0-10000], all elements from StreamSamp and all micro-clusters from CluStream. Then, we use the same formula as StreamSamp and CluStream.

At different observations time ($t = 10000$, $t = 20000$, etc.), we calculate the relative mean error to evaluate the algorithms' performance. The relative error of an estimate $\hat{X_0}$ is $|\hat{X_0} - X_0|/X_0$.

The same behavior is observed for all studied attributes. We only present results of the first attribute. As shown in fig. 4, the relative mean error calculated on StreamSamp increases with the aging period [0-10000].

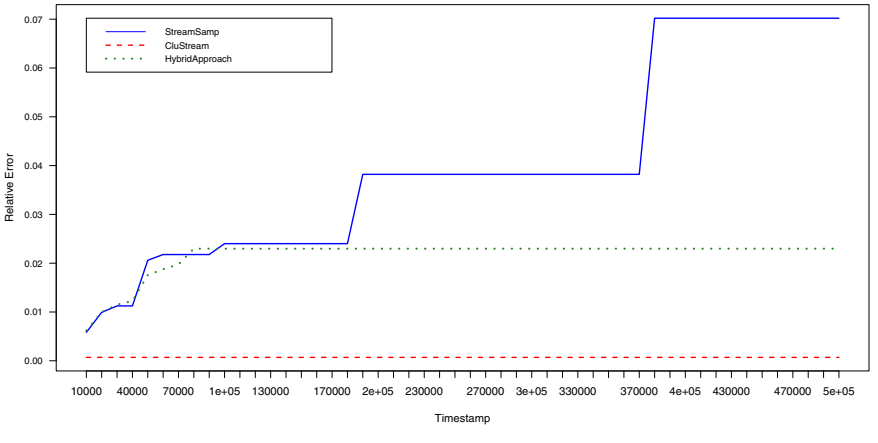



**Fig. 4** Mean Evaluation. [note that Clustream correctly estimate the mean]

Clustream has the same aggregated value for the mean given that keeps two snapshots (at the beginning and the end of the studied period). On the first time horizon [0-30000], the Hybrid approach is similar to StreamSamp. From $t = 30000$, samples in Hybrid approach begin to move to CluStream.

These results confirm the deterioration of StreamSamp's performance over time. It becomes difficult to estimate this aggregate on distant past because there is no elements covering the period [0-10000]. However, CluStream correctly estimates the mean queries because it keeps the value mean exactly. The Hybrid approach gives better mean's estimations than the pure StreamSamp algorithm for an old fixed time period. When all elements move to CluStream, the Hybrid approach is stabilizing.

### *4.3 Median Evaluation*

The aim of this experiment is to study the performance of the different approaches on median estimation. Like the mean evaluation, we study the aging period [0-10000]. The estimated error is calculated according to positions in ranking values:

$$\text{error} = |EstimatedPosition - RealPosition| \,/\, \text{Window Size}$$

The *Real Position* is calculated over the original dataset (5000 in our case) while, the *Estimated Position* is calculated over the resulted summary and the *Window Size* represents the original number of elements (10000 in our case).

With StreamSamp, the estimated position is easily calculated because the sampling process preserves elements' structure. We firstly extract all elements included on [0-10000] and we sort them according to attribute value. We choose the element which divides the distribution into two equal parts (until reaching $\frac{\sum weights}{2}$). The estimated position corresponds to the rank of this element in the original data set.

To calculate the estimated median position, we calculate the sum of all weights. Then, we sort elements according to the attribute value. We choose the element which divides the distribution into two equal parts.

However, in the Clustream algorithm, stream elements are absorbed in micro-clusters. For that, on period [0-10000], we use the centroids of micro-clusters as elements and the number of their elements as weight. We calculate the estimated position according to the process described above (with StreamSamp: sorting, extracting, ranking).

In the Hybrid approach, we can have the two processes (StreamSamp and CluStream) running in parallel. We extract from StreamSamp all samples included on [0-10000]. For CluStream, we search the closest snapshots kept between 0 and 10000 to extract the micro-clusters. We merge the elements from StreamSamp summary with the centroids of micro-clusters. Then, we calculate the estimated position, according to StreamSamp and CluStream:

- Sorting: we sort the new set of elements;
- Extracting: we extract the median value. It can be resulted from StreamSamp or CluStream. If the median value corresponds to a micro-cluster centroid, it can be not found in the original data set;

- Ranking: we extract the position of the median value from the original data set. If the value is not found, we search the position of the nearest lowest value and the the position of the nearest highest value from the original data set. The estimated position is the mean of these borders.

The results are shown in fig. 5. Like the mean evaluation, the relative median error calculated on StreamSamp increases with the aging period [0-10000]. The relative median error is calculated once on Clustream given that it keeps two snapshots. At the beginning, the Hybrid approach follows the behavior of StreamSamp. However, it tries to stabilize towards the end when all samples pass to CluStream.

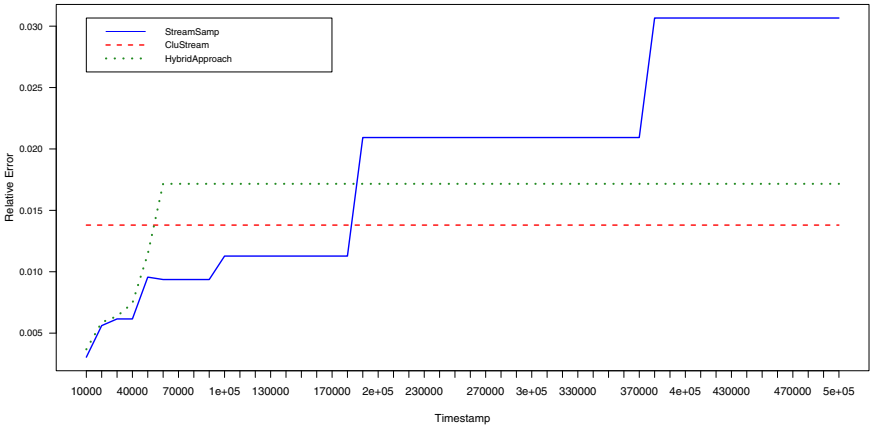



**Fig. 5** Median Evaluation

The re-sampling process (in StreamSamp) deteriorates the median estimation over time. However, the Hybrid approach takes advantages of StreamSamp before its degradation. Then, its performances follow the CluStream process.

## *4.4 Clustering Evaluation*

Clustering is the process of organizing objects into groups whose members are similar in some way. The similarity is based essentially on a distance criterion. To evaluate the performance of the different approaches on clustering process, we calculate at different timestamps, the intra-class inertia[1] for a clustering performed on the period [0-10000]. For CluStream, we initialize the process with 15 clusters. For StreamSamp, and for samples generated by StreamSamp in the Hybrid approach, we apply the K-means algorithm to create 15 clusters.

[1] Intra-class inertia : Sum of square of distances between elements and the centroid of the closest cluster.

We compare our approach to the K-means algorithm applied once to all the [1-10000] elements. As was expected in fig. 6, K-means gives the best results given that it is applied to the original data set. However, StreamSamp deteriorates more and more over time with the aging period [0-10000]. Clustream gives close performances to K-means. The hybrid approach has a similar behavior to StreamSamp on [0-40000] and converges to an accuracy close to the CluStream behavior.

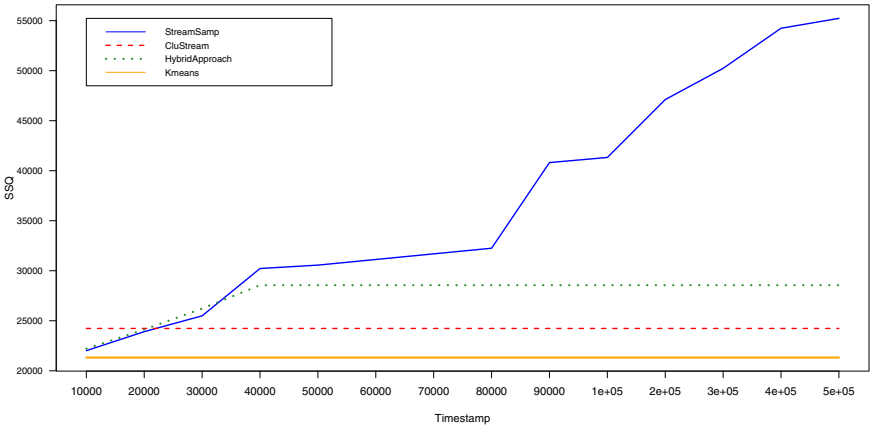



**Fig. 6** Intra Cluster Inertia

## *4.5 Classification Evaluation*

To evaluate the models generated by the different algorithms, we use a *cross-validation procedure*. This method estimates how accurately a predictive model performs in practice.

We evaluate the performances of the generated models using the different summarizing algorithms. The models are constructed using the CoverType dataset at different timestamps, over the fixed period [0-10000]. The model resulted from the original dataset ([0-1000] in this task) is our reference model.

To calculate the error in the StreamSamp process, we use the summary built over the period [0-10000]. To take into account the weight, each element value is multiplied by its weight. The model resulted will be evaluated with *cross-validation* method. The advantage of sampling based algorithms is to keep intact elements representation. Thus, it is easy to use this representation with the CART algorithm (another algorithms like SVM can be used) in order to construct the model.

CluStream keeps only the mean and variance of each micro-cluster. The original element values and the labels are lost. We need to process the resulted summary in order to be able to apply the CART algorithm over the data. Two pre-processing have to be considered : (i) generating element values, (ii) adding label attribute.

| Attribute 1 | Attribute 2 | Label |
|---|---|---|
| 1.5 | 3 | A |
| 1.7 | 1.8 | B |
| 7 | 7.3 | C |

| Attribute 1 | Attribute 2 | Label A | Label B | Label C |
|---|---|---|---|---|
| 1.5 | 3 | 1 | 0 | 0 |
| 1.7 | 1.8 | 0 | 1 | 0 |
| 7 | 7.3 | 0 | 0 | 1 |

**Fig. 7** Disjonctive table

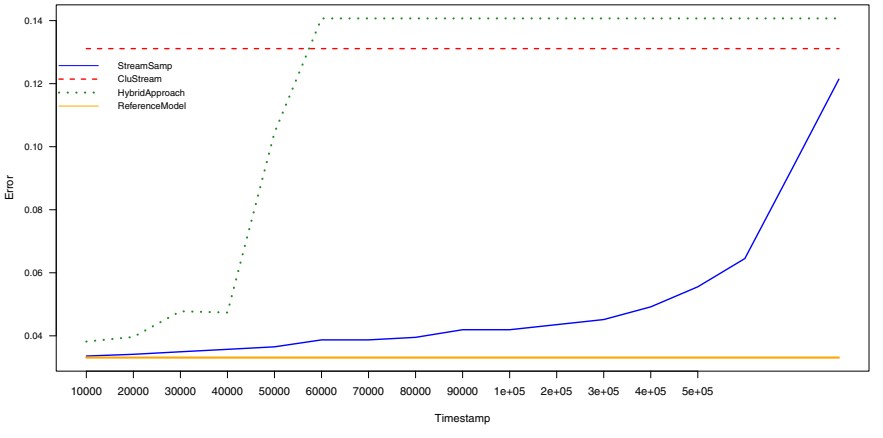



**Fig. 8** Classification evaluation

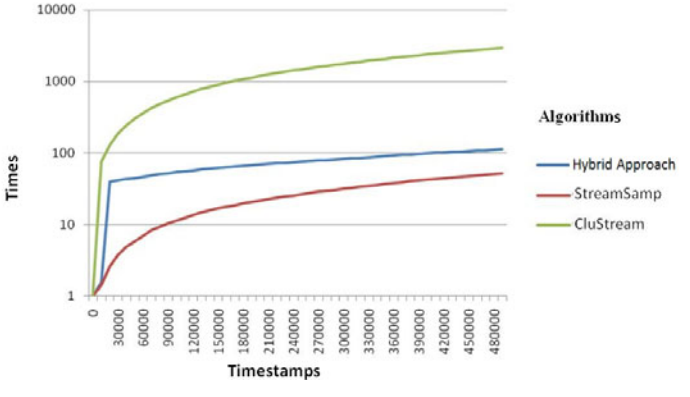



**Fig. 9** Runtime Evaluation (logarithmic scale)

For the first process, we use the information kept in micro-clusters to generate $n_i$ elements ($n_i$ is the number of elements in the micro-cluster), following a Gaussian distribution. This operation is repeated 100 times because of randomly generation process. The result corresponds to the mean of these different drawings.

For the second process, we have to associate each generated element to one label. To distinguish element belonging their labels, we need to transform the

label attribute in a disjunctive table. As shown in fig. 7, if we have three labels, we would add three binary attributes to the dataset. The label associated to an element will have '1' as value while all the others will have 'zero'. We use the probability $(Mean/n_i)$ to associate a label to an element. The Covertype data set contains 7 labels. However, predicting 7 labels is difficult. To achieve this, we transform the data set to 2 labels: most frequent label 'A' (a majority), and the rest 'B'.

The Hybrid approach contains samples generated by StreamSamp and micro-clusters generated by CluStream. We transform micro-clusters using the method described above and we concatenate it with samples. The CART algorithm is applied in order to generate the model.

As shown in fig. 8, we compared the derived models constructed by algorithms to the reference model (without summarizing operations). StreamSamp built the closest model to the reference because it used the real data unlike Clustream which built the worst model since all the data was generated. The Hybrid approach follows StreamSamp performances on recent time periods, and it tries to stabilize towards the end when all samples pass to CluStream.

The StreamSamp performances deteriorate over the time but it remains efficient for the classification task. The Hybrid approach presents better results than CluStream in more recent periods since it uses data from StreamSamp, however, the results deteriorated when elements moved to CluStream.

### *4.6 Runtime Evaluation*

In a data stream framework, the run-time execution is a very important feature of processing stream data. For this evaluation, we take account the global elapsed time for the data stream processing. We are not interested in the aging period [0-10000]. As shown in fig. 9, for the same volume of data in the stream, CluStream requires much more time than StreamSamp and the Hybrid approach. In CluStream and Hybrid approach curves, the peak corresponds to the initialization phase of micro-clusters. StreamSamp provides the best performances. However, it is important to clarify that the Hybrid approach is slower than StreamSamp but much faster than CluStream.

## 5 Discussion and Conclusions

In this chapter, we have developed an efficient method, called 'Hybrid approach', for summarizing data streams. The aim is to propose a representative summary of the history of the stream data. We have presented a transition strategy from the StreamSamp algorithm to the CluStream algorithm while taking into consideration the advantages of both methods. The transition from one process to another is not a random procedure but respects a number of criteria. If one of these criteria is no longer checked, the merge process of samples in StreamSamp is stopped and the elements of the samples are sent to CluStream.

Several experiments are presented with real and synthetic data. They show that the new approach almost always gives better results than using any one of the two mentioned algorithms.

We present the results for querying tasks (median and mean) and data mining tasks (clustering and classification). The mean of a sample is a sufficient statistic, but not the median. Therefore, re-estimating the mean over time is an easiest task than the median estimation.

To meet CluStream timeline, samples must be sent in order: samples with higher orders (lower timestamp) must be moved away before the lower order samples (higher timestamp). The drawback of this approach is that some samples are obliged to move early to CluStream while they are still checking the transition criteria. This occurs when the transition criteria are no longer observed for two samples of order $i$ while sample of order $k$ ($k \succ i$) still satisfy these criterion. Studies are underway to develop techniques in order to avoid this inconvenience.

A natural extension of this work concerns the integration of qualitative data. With StreamSamp the problem of using categorical variables does not arise. As for CluStream, there are extensions which deal with categorical variables (HClustream (Yang and Zhou, 2006), SCLOPE (Kok Leong Ong et al., 2004)). Integrating qualitative data in our approach is possible provided that the transition parameters are redefined.

We evaluated our approach over querying and data mining tasks. A second perspective could concern the evaluation of the Hybrid approach over different selections on the stream.

## References

Abadi, D.J., Carney, D., Çetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., Zdonik, S.: Aurora: a new model and architecture for data stream management. The VLDB Journal 12(2), 120–139 (2003), http://dx.doi.org/10.1007/s00778-003-0095-z

Aggarwal, C. (ed.): Data Streams – Models and Algorithms. Springer, Heidelberg (2007)

Aggarwal, C.C.: On biased reservoir sampling in the presence of stream evolution. In: VLDB 2006: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, pp. 607–618 (2006)

Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A Framework for Clustering Evolving Data Streams. In: VLDB, pp. 81–92 (2003)

Al-Kateb, M., Lee, B.S., Wang, X.S.: Adaptive-Size Reservoir Sampling over Data Streams. In: SSDBM, p. 22 (2007)

Arasu, A., Babcock, B., Babu, S., Datar, M., Ito, K., Nishizawa, I., Rosenstein, J., Widom, J.: STREAM: the stanford stream data manager (demonstration description). In: SIGMOD 2003: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, p. 665. ACM, New York (2003), http://doi.acm.org/10.1145/872757.872854

Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: PODS 2002: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 1–16. ACM, New York (2002), http://doi.acm.org/10.1145/543613.543615

Bloom, B.H.: Space/time trade-offs in hash coding with allowable errors. Commun. ACM 13(7), 422–426 (1970), http://doi.acm.org/10.1145/362686.362692

Csernel, B.: Résumé généraliste de flux de données. Ph.D. thesis, Ecole Nationale Supérieur des Télécommunications (Février 2008)

Csernel, B., Clérot, F., Hébrail, G.: StreamSamp: DataStream Clustering Over Tilted Windows Through Sampling. In: ECML PKDD 2006 Workshop on Knowledge Discovery from Data Streams (2006)

Flajolet, P., Martin, G.N.: Probabilistic counting algorithms for data base applications. J. Comput. Syst. Sci. 31(2), 182–209 (1985), http://dx.doi.org/10.1016/0022-00008590041-8

Gemulla, R., Lehner, W.: Sampling time-based sliding windows in bounded space. In: SIGMOD Conference, pp. 379–392 (2008)

Golab, L., Özsu, M.T.: Issues in data stream management. SIGMOD Rec. 32(2), 5–14 (2003), http://doi.acm.org/10.1145/776985.776986

Guha, S., Harb, B.: Wavelet synopsis for data streams: minimizing non-euclidean error. In: KDD 2005: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 88–97. ACM, New York (2005), http://doi.acm.org/10.1145/1081870.1081884

Guha, S., Koudas, N., Shim, K.: Data-streams and histograms. In: STOC 2001: Proceedings of the thirty-third annual ACM symposium on Theory of computing, pp. 471–475. ACM, New York (2001), http://doi.acm.org/10.1145/380752.380841

Ioannidis, Y.E., Poosala, V.: Histogram-Based Approximation of Set-Valued Query-Answers. In: VLDB, pp. 174–185 (1999)

Jagadish, H.V., Koudas, N., Muthukrishnan, S., Poosala, V., Sevcik, K.C., Suel, T.: Optimal Histograms with Quality Guarantees. In: VLDB, pp. 275–286 (1998)

Ma, L., Nutt, W., Taylor, H.: Condensative Stream Query Language for Data Streams. In: ADC, pp. 113–122 (2007)

Muthukrishnan, S., Strauss, M., Zheng, X.: Workload-Optimal Histograms on Streams. In: Brodal, G.S., Leonardi, S. (eds.) ESA 2005. LNCS, vol. 3669, pp. 734–745. Springer, Heidelberg (2005)

Park, B.-H., Ostrouchov, G., Samatova, N.F., Geist, A.: Reservoir-Based Random Sampling with Replacement from Data Stream. In: SIAM SDM International Conference on Data Mining (2004)

Puttagunta, V., Kalpakis, K.: Adaptive Clusters and Histograms over Data Streams. In: IKE International Conference on Information and Knowledge Engineering, pp. 98–104 (2005)

Vitter, J.S.: Random sampling with a reservoir. ACM Trans. Math. Softw. 11(1), 37–57 (1985), http://doi.acm.org/10.1145/3147.3165

Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. SIGMOD Rec. 25(2), 103–114 (1996), http://doi.acm.org/10.1145/235968.233324

# SPAMS: A Novel Incremental Approach for Sequential Pattern Mining in Data Streams

Lionel Vinceslas, Jean-Emile Symphor, Alban Mancheron, and Pascal Poncelet

**Abstract.** Mining sequential patterns in data streams is a new challenging problem for the datamining community since data arrives sequentially in the form of continuous rapid and infinite streams. In this paper, we propose a new on-line algorithm, SPAMS, to deal with the sequential patterns mining problem in data streams. This algorithm uses an *automaton-based structure* to maintain the set of frequent sequential patterns, *i.e.* SPA (Sequential Pattern Automaton). The sequential pattern automaton can be smaller than the set of frequent sequential patterns by two or more orders of magnitude, which allows us to overcome the problem of combinatorial explosion of sequential patterns. Current results can be output constantly on any user's specified thresholds. In addition, taking into account the characteristics of data streams, we propose a well-suited method said to be approximate since we can provide near optimal results with a high probability. Experimental studies show the relevance of the SPA data structure and the efficiency of the SPAMS algorithm on various datasets. Our contribution opens a promising gateway, by using an automaton as a data structure for mining frequent sequential patterns in data streams.

**Keywords:** Algorithm, Data Stream, Sequential Pattern, Automata.

Lionel Vinceslas · Jean-Emile Symphor
CEREGMIA, Université des Antilles et de la Guyane, Martinique, France
e-mail: lionel.vinceslas, je.symphor@martinique.univ-ag.fr

Alban Mancheron
LIRMM, 161 rue Ada 34392 Montpellier CEDEX 5, France
e-mail: alban.mancheron@lirmm.fr

Pascal Poncelet
LIRMM - UMR 5506 - 161 rue Ada 34392, Montpellier Cedex 5, France
e-mail: pascal.poncelet@lirmm.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 201–216.
springerlink.com 

## 1 Introduction

Concerned with many applications (*e.g.* medical data processing, marketing, safety and financial analysis), mining sequential patterns is a challenging problem within the datamining community. More recently these last years, many emerging applications, such as traffic analysis in networks, web usage mining or trend analysis, generate a new type of data, called data streams. A data stream is an ordered sequence of transactions, potentially infinite, that arrives in a timely order. The characteristics of a data stream can be expressed as follows (*cf.* Lin 2005):

- **Continuity.** Data continuously arrive at a high speed.
- **Expiration.** Data can be read only once.
- **Infinity.** The total amount of data is unbounded.

Therefore, mining in data streams should meet the following requirements as well as possible. Firstly, owing to the fact that past data cannot be stored, the methods can provide approximate results but accuracy guarantees are required. Secondly, the unbounded amount of data supposes that the methods are adjustable according to the available resources, especially for the memory. Lastly, a model is needed which adapts itself to continuous data stream over a time period.

**Previous Work**

Initially, the first work deal with the case of static databases and propose exact methods for mining sequential patterns. We can quote as an example, the algorithms GSP, PSP, FreeSpan, SPADE, PrefixSpan, SPAM and PRISM, respectively proposed by Srikant and Agrawal (1996); Masseglia *et al.* (1998); Han *et al.* (2000); Zaki (2001); Pei *et al.* (2001); Ayres *et al.* (2002); Gouda *et al.* (2007). Thus, the first algorithms mentioned above for mining sequential patterns are not adapted any more in the context of data streams. In Raïssi and Poncelet (2007), authors propose to use sampling techniques for extracting sequential patterns in data streams. Nevertheless, the context is quite different from our proposal since they mainly focus on a summarization of the stream by using a reservoir sampling-based approach. In that case, the sampling could be considered as a static database and then any sequential pattern mining algorithm can be applied. It was shown in Garofalakis *et al.* (2002), that methods, said to be approximate, are well adapted to the context of data streams. However, the principal difficulty resides in the search of a trade-off between time and memory performances, and the quality of the mining results as well as in recall as in precision. So far, the literature concerning the mining of sequential patterns in data streams is relatively poor. In Chang and Lee (2005), the authors proposed the algorithm eISeq, using a tree-based data structure. This algorithm is a one pass algorithm, which processes the stream sequentially, transaction per transaction. However, the longer the sequential patterns are, the less this algorithm is performant. That is due to the generation of all the sequential sub-patterns which increase exponentially. For example, if $< a1, \cdots, ai >$ is a sequential pattern, there are $(2^i - 1)$ sequential sub-patterns to be created. To alleviate this difficulty , the GraSeq algorithm have been presented in Li and Chen (2007). Their approach is based on an oriented graph data structure to

limit the sequential-sub-patterns generation phase. However, this approach supposes a costly pre-processing step for regrouping the transactions.

**Our Contribution**

In this paper, we propose a new one-pass algorithm: SPAMS (Sequential Pattern Automaton for Mining Streams). SPAMS is based on the on-line and the incremental building and updating of an automaton structure: the SPA. The SPA (Sequential Patterns Automaton) is a deterministic finite automaton, which indexes the frequent sequential patterns in data streams. The remainder of the paper is organized as follows. Section 2 states the problem formally. In Section 3, we recall some prerequisite preliminary concepts and we present our approach in section 4. Experimental studies are provided in the section 5 and the conclusion is presented in the last section.

## 2 Problem Definition

In this section, we give the formal definition of the problem of mining sequential patterns in data streams. First, we give a brief overview of the traditional sequence mining problem by summarizing the formal description introduced in Srikant and Agrawal (1996). Second, we examine the problem when considering streaming data. Let $\mathbb{I} = \{i_1, i_2, \ldots, i_m\}$ be a set of literals called items and let $DB$ a database of customer transactions where each transaction $T$ consists of customer-id, transaction time and a set of items involved in the transaction. An itemset is a non-empty set of items. A sequential pattern s is a set of itemsets ordered according to their timestamp and is denoted by $< s_1 s_2 \cdots s_n >$, where $s_j$, for $j \subseteq [1..n]$, is an itemset. A $k$ sequential pattern is a sequential pattern of $k$ items or of length $k$. A sequential pattern $S' =< s'_1 s'_2 \cdots s'_n >$ is a sub-pattern of another sequential pattern $S =< s_1\ s_2\ \cdots\ s_n >$, denoted $S' \prec S$ if there exists integers $i_1 < i_2 < \cdots i_j \cdots < i_n$ such that $s'_1 \subseteq s_{i_1}$, $s'_2 \subseteq s_{i_2}, \cdots, s'_n \subseteq s_{i_n}$. All transactions from the same customer are grouped together and sorted in an increasing order and are called a data sequence. A support value (denoted supp(S)) for a sequential pattern gives its number of actual occurrences in $DB$. Nevertheless, a sequential pattern in a data sequence is taken into account only once to compute the support even if several occurrences are discovered. A data sequence contains a sequential pattern S if S is a sub-pattern of the data sequence. In order to decide whether a sequential pattern is frequent or not, a minimum support value (denoted $\sigma$) is specified by the user, and the sequential pattern is said to be $\theta$-frequent if $supp(S) \geq \sigma$, where $\sigma = \ulcorner \theta \times |DB| \urcorner$ with $\theta \in ]0;1]$ and $|DB|$ the size of the database. Given a database of customer transactions, the problem of sequential pattern mining is to find all the sequential patterns whose support is greater than a specified threshold minimum support. Extended to the case of data streams, this problem can be expressed as follows. Formally, a data stream $DS$ can be defined as a sequence of transactions, $DS = (T_1, T_2, \cdots, T_i, \cdots)$, where $T_i$ is the i-th arrived transaction. Each transaction, identified by a Tid, is associated with an Cid identifier (cf. the example in Table 1). Mining frequent sequential patterns remains to find all the

**Table 1** A data sequence built on $\mathbb{I} = \{a,b,c,d\}$

| | |
|---|---|
| $C_1$ | $< (b,d)\ (a,b,d)\ (a,c,d) >$ |
| $C_2$ | $< (b,c,d)\ (b,d) >$ |
| $C_3$ | $< (a,b)\ (c) >$ |

sequential patterns, whose support value is equal or greater than the fixed minimum support threshold for the known part of the data stream at a given time.

## 3 Prerequisites on Statistical Covers

We briefly present in this section the required theoretical materials on statistical covers that we have presented in Laur *et al.* (2007). So, we recall the following theorem.

**Theorem 1.** $\forall \theta, 0 < \theta \leq 1, \forall \delta, 0 < \delta \leq 1$*, we denote by* $m$ *and* $m^*$ *respectively the (*$\theta$*-frequent and* $\theta$*-infrequent) number of sequential patterns in the known part of the stream and in the whole stream. If we choose* $\varepsilon$ *such that:*

$$\varepsilon \geq \sqrt{\frac{1}{2m} \ln \frac{m^*}{\delta}} \ ,$$

*then* `Recall` $= 1$ *and respectively* `Precision` $= 1$ *with a probability of at least* $1 - \delta$*, when discarding all the sequential patterns that are not* $\theta'$*-frequent from the observation, where* $\theta' = \theta - \varepsilon$ *and respectively* $\theta' = \theta + \varepsilon$*.*

*The parameter* $\delta$ *is the statistical risk parameter potentially fixed by the user and the values* $\theta' = \theta \pm \varepsilon$ *are the statistical supports.*

The **sup-**$(\theta,\varepsilon)$**-cover** is the near-optimal smallest set of sequential patterns with a probability of at least $1-\delta$) containing all the sequential patterns that are $\theta$-frequent in the whole stream (eventually infinite). There are no false negative results with high probability. The **inf-**$(\theta,\varepsilon)$**-cover** is the near-optimal biggest set of sequential patterns with a probability of at least $1-\delta$) containing only sequential patterns that are $\theta$-frequent in the whole stream (eventually infinite). In this set, there are no false positive results with high probability, but false negative ones. We provided the proof of this theorem in Laur *et al.* (2007). By near-optimal, we express that any technique obtaining better bounds is condemned to make mistakes (the criterion to be maximized is not equal any more to 1). We precised also, that there is no assumption on the distribution of the stream.

## 4 The SPAMS Approach

Our approach is based on the incremental construction of an automaton which indexes all frequent sequential patterns from a data stream. For the mining process, we

do not make the assumption of an ideal data stream where transactions are sorted by customers. In fact, we make no assumptions either on the order of data, or on customers, or on the alphabet of the data. It's a real incremental approach for knowledge discovery in data streams. Moreover, to obtain the best quality of approximation, in both recall and precision, we also index the $(\theta - \varepsilon)$-frequent sequential patterns of the statistical cover, in addition to those $\theta$-frequent. In this way, we retain the minimum number of candidates, which limits the combinatorial explosion.

## *4.1 SPA: The Sequential Patterns Automaton*

In a more formal way, we define in this section the automaton of sequential patterns, SPA. For further information on the automata theory, we suggest the presentation made by Hopcroft and Ullman (1979).

**Definition 1 (Finite state automaton).** A finite state automaton $\mathcal{A}$ is a 5-tuple such that $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, \mathcal{I}, \mathcal{F})$, where $\mathcal{Q}$ is a finite set of states, $\Sigma$ an input alphabet, $\delta \subseteq \mathcal{Q} \times \Sigma \times \mathcal{Q}$ is a set of transitions, $\mathcal{I} \subseteq \mathcal{Q}$ and respectively $\mathcal{F} \subseteq \mathcal{Q}$ are the set of initials and finals states.

**Definition 2 (Deterministic finite state automaton).** A finite state automaton $\mathcal{A} = (\mathcal{Q}, \Sigma, \delta, \mathcal{I}, \mathcal{F})$ is deterministic if and only if it exists a unique initial state (*i.e.* $|\mathcal{I}| = 1$) and if $\forall\ p, q \in \mathcal{Q}$ and $\alpha \in \Sigma$, $(p, \alpha, q), (p, \alpha, q') \in \delta \Rightarrow q = q'$.

The label of a transition $t$ going from a state $q_i$ to a state $q_j$, denoted $t = q_i \overset{\alpha}{\longmapsto} q_j$ is the symbol $\alpha$. A path in $\mathcal{A}$ is a sequence $c = t_1, \cdots, t_n$ of consecutive transitions. The label of a path $c$ is denoted $|c| = \alpha_1 \cdots \alpha_n$ , or $c : q_0 \overset{w}{\longmapsto} q_n$ with $w = |c|$. A label is also called a *word*. A path $c : q_i \overset{w}{\longmapsto} q_j$ is said to be *successful* if and only if $q_i \in \mathcal{I}$ and $q_j \in \mathcal{F}$. A word $w$ is said to be *accepted* or *recognised* by the automaton $\mathcal{A}$ if it is the label of a successful path.

**Definition 3 (Language accepted by a DFA).** Let $\mathcal{A} = (\mathcal{Q},\ q_0,\ \mathcal{F},\ \Sigma,\ \delta)$ be a deterministic finite state automaton (DFA). The *language accepted* or recognised by $\mathcal{A}$, denoted $\mathcal{L}(\mathcal{A})$, is the set of all accepted words:

$$\mathcal{L}(\mathcal{A}) = \left\{ w \subseteq \Sigma^* \mid \exists\ c : q_0 \overset{w}{\longmapsto} q_j,\ q_j \in \mathcal{F} \right\}$$

**Definition 4 (The Sequential Patterns Automaton).** The sequential patterns automaton (SPA) is a deterministic finite state automaton, *i.e.* a 5-tuple $SPA = (\mathcal{Q},\ q_0,\ F,\ \Sigma,\ \delta)$, whose accepted language $\mathcal{L}(\mathcal{SPA})$ is the set of frequent sequential patterns.

**Definition 5 (The sequence item).** Let $SPA = (\mathcal{Q},\ q_0,\ F,\ \mathbb{I},\ \delta)$ be the automaton of sequential patterns. We add to the set $\Sigma$, a special item called the *sequence item*,

denoted arbitrarily #. This item is an item that separates itemsets within sequential patterns (*cf.* figure 1).

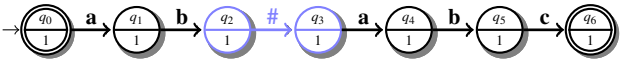



**Fig. 1** An automaton indexing the sequential pattern $< (a,b)(a,b,c) >$

**Definition 6 (The sink state).** Let $SPA = (\mathcal{Q},\ q_0,\ F,\ \mathbb{I},\ \delta)$ be the automaton of sequential patterns. We add to the set $\mathcal{Q}$, a special state called the *sink state*, denoted $q_\infty$. It's a temporary state used by the transition function to generate the other states of the automaton.

**Definition 7 (Support of a state).** Let $SPA = (\mathcal{Q},\ q_0,\ F,\ \mathbb{I},\ \delta)$ be the automaton of sequential patterns, and $q \in \mathcal{Q}$, a final state. We define the *support of the state q*, denoted $|q|$, as an integer representing the support of sequential patterns recognised in this state.

**Lemma 1.** *Let $\mathcal{L}_q \subseteq \mathcal{L}(SPA)$ be the set of words (i.e. sequential patterns) recognised in the state $q \in Q$. According to definition 7, the following assertion is definitely obvious:*

$$\forall\ w_i, w_j \in \mathcal{L}_q \subseteq \mathcal{L}(SPA)\ \ (1 \leq i, j \leq |\mathcal{L}_q|)\ \ ,\ supp(w_i) = supp(w_j)$$

*Property 1.* Let $SPA = (\mathcal{Q},\ q_0,\ F,\ \Sigma,\ \delta)$ be the sequential patterns automaton:

$$\forall q_i \overset{\alpha}{\longmapsto} q_j \in SPA\ \ (q_i, q_j \in \mathcal{Q},\ \alpha \in \Sigma)\ ,\ \ |q_i| \geq |q_j|$$

*Proof.* *Let $c_1 : q_0 \overset{w}{\longmapsto} q_i$ and $c_2 : q_i \overset{\alpha}{\longmapsto} q_j \in SPA$ be two paths ($\alpha \in \Sigma$; $w \in \Sigma^*$). According to the Apriori property (Agrawal and Srikant, 1994) (i.e. for any frequent itemset, all sub-itemsets are frequent), if $z = w \cdot \alpha$ is the label of a successful path $c_3 : q_0 \overset{z}{\longmapsto} q_j$, then $c_1$ is also a successful path and $supp(w) \geq supp(z)$. According to definition 7, $supp(w) = |q_i|$ and $supp(z) = |q_j|$.* ***This shows that*** $|\mathbf{q_i}| \geq |\mathbf{q_j}|$

*Property 2.* Let $SPA = (\mathcal{Q},\ q_0,\ F,\ \Sigma,\ \delta)$ be the sequential patterns automaton, $\mathcal{R}(\mathcal{Q}, \alpha)$ be the set of reachable states by $\alpha$ and $\mathcal{R}(\mathcal{Q}, \beta)$ be the set of reachable states by $\beta$:

$$\forall \alpha, \beta \in \Sigma\ ,\ \ \mathcal{R}(\mathcal{Q}, \alpha) \cap \mathcal{R}(\mathcal{Q}, \beta) = \emptyset$$

## 4.2 The SPAMS Algorithm

### 4.2.1 Notations

In the following, we define some of the notations used in SPAMS:

- $\mathcal{T}$ is the set of transition of the automaton.
- $\mathcal{T}_s$ is the set of *ingoing transitions* on the state $s \in \mathcal{Q}$.
- $|\mathcal{T}_s|$ is the number of *ingoing transitions* on the state $s \in \mathcal{Q}$.
- $\mathcal{Q}^{\#}$ is the set of *reachable states* by the *sequence item*.
- $\mathcal{Q}_{\texttt{cid}}$ is the set of reachable states for a customer id `cid`.
- $\mathcal{T}^r$ is the set of *reachable transitions*, *i.e.* transitions labelled by the item being processed.
- $\mathcal{C}$ is the set of customers id.
- $\mathcal{C}_s$ is the set of customers id for a state $s \in \mathcal{Q}$, *i.e.* the customers whose indexed sequential patterns use the state $s$.

### 4.2.2 Presentation

According to definition 7, a state may recognise several sequential patterns whose support is the same. So, if the support of one or more sequential patterns recognised in a state $q$, has to change (*i.e.* their support is incremented by 1), the definition 7 is no longer respected. To resolve this problem, we make a copy $q'$ of the state $q$: all sequential patterns recognised in the state $q$ are not moved. We move only on the state $q'$, the sequential patterns whose support has changed. This is done by a movement of some ingoing transitions from the state $q$ to the state $q'$. It is evident that all sequential patterns recognised in the state $q'$ have the same support (*cf.* definition 7). Finally, we create the same outgoing transitions of the state $q$ for the state $q'$.

Our algorithm is divided into three main modules which are INSERT, PRUNE and NEXT.

The INSERT module: This module is called by the SPAMS algorithm for each item read from the data stream. Let *cid* be the customer id, and $\alpha \in \Sigma$ the item being processed. This module is responsible for the creation of new transitions in the automaton, and therefore of the application of definition 7. So, the INSERT module will try to create all necessary transitions of the form $s \overset{\alpha}{\longmapsto} s'$. Therefore, we need to know the corresponding states $s$ and $s'$. The state $s$ is obtained by scanning the list of reachable states for the customer id *cid*, denoted $\mathcal{Q}_{cid}$. This means each customer id has its own set of reachable states. We proceed in the following way:

- First, if this customer id is new ($cid \notin \mathcal{C}$), we update the following sets: $\mathcal{C} = \mathcal{C} \cup \{\ cid\ \}$ , $\mathcal{Q}_{cid} = \{\ q_0\ \}$ and $\mathcal{C}_{q_0} = \mathcal{C}_{q_0} \cup \{cid\}$.
- Then, for each state $s \in \mathcal{Q}_{cid}$, if there is no state $s' \in \mathcal{Q}$ such that the transition $s \overset{\alpha}{\longmapsto} s'$ exist, we create a new transition to the sink state ($\mathcal{T} = \mathcal{T} \cup \{s \overset{\alpha}{\longmapsto} q_\infty\}$)

and update the set: $\mathcal{T}^r = \mathcal{T}^r \cup \{s \overset{\alpha}{\longmapsto} q_\infty\}$. Otherwise, if the transition $s \overset{\alpha}{\longmapsto} s'$ exists, we update the set: $\mathcal{T}^r = \mathcal{T}^r \cup \{s \overset{\alpha}{\longmapsto} s'\}$.

◇ For each state $s'$ such that $s \overset{\alpha}{\longmapsto} s' \in \mathcal{T}^r$, we make the following step:

- If the state $\mathfrak{s}' \neq \mathfrak{q}_\infty$ and $|\mathcal{T}_{\mathfrak{s}'}| = |\mathcal{T}_{\mathfrak{s}'} \cap \mathcal{T}^{\mathfrak{x}}|$, then:

1. we update the set: $\mathcal{Q}_{cid} = \mathcal{Q}_{cid} \cup \{\ s'\ \}$.
2. if the customer id $cid \notin \mathcal{C}_{s'}$, then $|s'| = |s'| + 1$ and we update the set: $\mathcal{C}_{s'} = \mathcal{C}_{s'} \cup \{\ cid\ \}$.
3. if $|s'| < min\_sup$, we call the prune module: PRUNE($s'$)

- Otherwise (*i.e.* $s' = q_\infty$ or $|\mathcal{T}_{s'}| \neq |\mathcal{T}_{s'} \cap \mathcal{T}^r|$):

1. we create a new state $p$ and update the set: $\mathcal{Q}_{cid} = \mathcal{Q}_{cid} \cup \{\ p\ \}$.
2. we also update the set: $\mathcal{C}_p = \mathcal{C}_{s'} \cup \{\ cid\ \}$
3. if the customer id $cid \notin \mathcal{C}_{s'}$, then $|p| = |s'| + 1$, otherwise $|p| = |s'|$
4. if the item $\alpha$ is the sequence item, we update the set: $\mathcal{Q}^{\#} = \mathcal{Q}^{\#} \cup \{\ p\ \}$
5. for each ingoing transition $s \overset{\alpha}{\longmapsto} s' \in \mathcal{T}^r$, we delete it and create the ingoing transition $s \overset{\alpha}{\longmapsto} p : \mathcal{T} = \mathcal{T} \setminus \left\{\ s \overset{\alpha}{\longmapsto} s'\ \right\} \cup \left\{s \overset{\alpha}{\longmapsto} p\right\}$
6. for each outgoing transition $s' \overset{\beta}{\longmapsto} s'' \in \mathcal{T}$ ($\beta \in \Sigma$, $s'' \in \mathcal{Q}$), we create the same outgoing transition for the state $p : \mathcal{T} = \mathcal{T} \cup \left\{p \overset{\beta}{\longmapsto} s''\right\}$
7. if $|p| < min\_sup$, we call the prune module: PRUNE($p$)

◇ We update the set of reachable transitions: $\mathcal{T}^r = \emptyset$

The PRUNE module: This module is called by the INSERT module in order to prune a state from the automaton. Not only does it erase the concerned state but also the states and transitions reachable from itself.

The NEXT module: When the module INSERT has processed all items of a transaction, for a given customer id (*cid*), the module NEXT is called. This module works as follows:

**1.** We save the set $\mathcal{Q}_{cid} : \mathcal{Z} = \mathcal{Q}_{cid}$
**2.** We update the set $\mathcal{Q}_{cid} : \mathcal{Q}_{cid} = \mathcal{Q}_{cid} \setminus \{\mathcal{Q}_{cid} \cap \mathcal{Q}^- \cup \{q_0\}\}$
**3.** We call the module INSERT giving as parameters the customer id (*cid*) and the sequence item (#).
**4.** We update the set $\mathcal{Q}_{cid} : \mathcal{Q}_{cid} = \mathcal{Z} \cap \mathcal{Q}^{\#} \cup \{q_0\}$

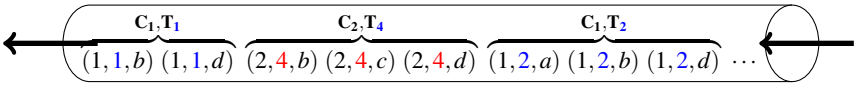
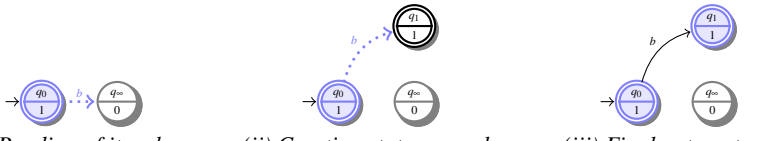



**Fig. 2** Example of an unordered data stream generated from table 1

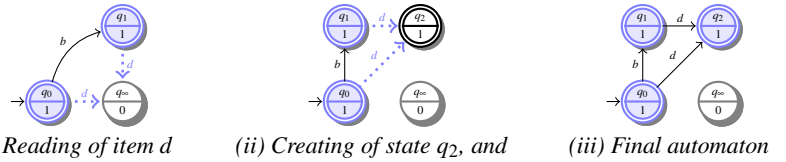
### 4.2.3 An Example of Construction

To illustrate the functioning of our algorithm, we process the example of Table 1, as an unordered stream database (*cf.* figure 2), using $\theta = 0.4$ as the support threshold. Thus, we work in the general case of data streams, *i.e.* without assuming any ordering of transactions by customer id. Figures 3, 4, 6, 7, 8 and 10 illustrate the module INSERT, *i.e.* the reading and the insertion of an item (*cf.* Section 4.2.2 for further explanation). Figures 5 and 9 illustrate the module NEXT, *i.e.* the end of the call to the module INSERT, which also corresponds to the end of processing every item of a transaction (*cf.* Section 4.2.2 for further explanation).



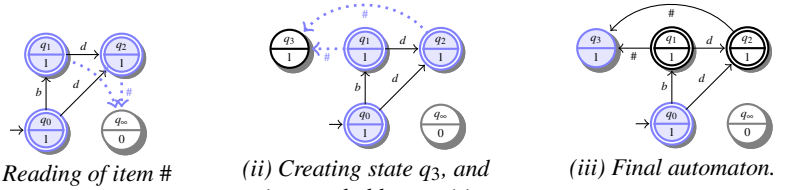

*(i) Reading of item b* *(ii) Creating state $q_1$, and moving reachable transitions.* *(iii) Final automaton*

**Fig. 3** Reading and insertion of item *b* (transaction 1)



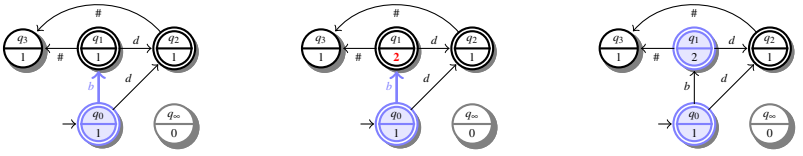

*(i) Reading of item d* *(ii) Creating of state $q_2$, and moving reachable transitions.* *(iii) Final automaton*

**Fig. 4** Reading and insertion of item *d* (transaction 1)



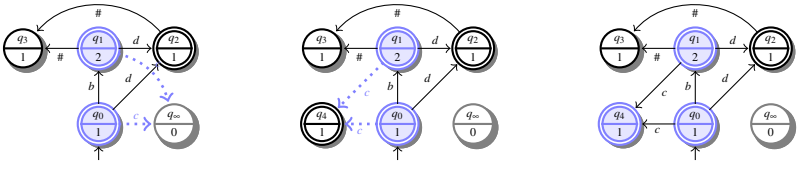

*(i) Reading of item #* *(ii) Creating state $q_3$, and moving reachable transitions.* *(iii) Final automaton.*

**Fig. 5** End of processing transaction 1



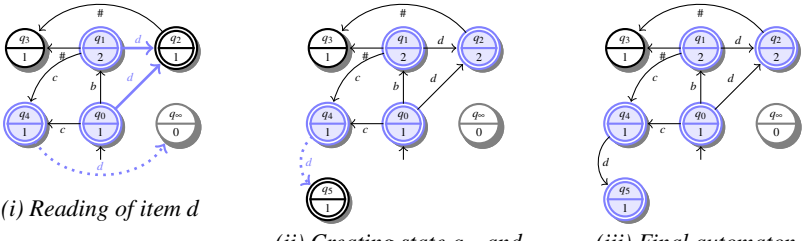
*(i) Reading of item b* *(ii) Incrementing support of state $q_1$* *(iii) Final automaton*

**Fig. 6** Reading and insertion of item *b* (transaction 2)



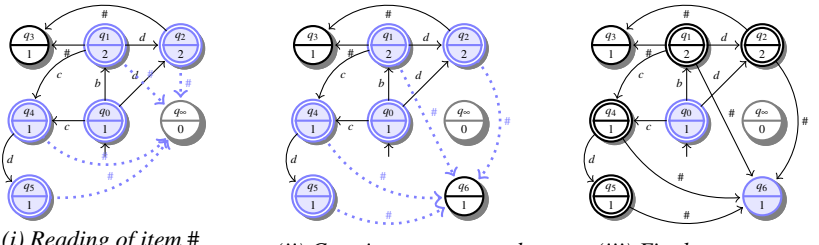
*(i) Reading of item c* *(ii) Creating state $q_4$, and moving reachable transitions.* *(iii) Final automaton*

**Fig. 7** Reading and insertion of item *c* (transaction 2)



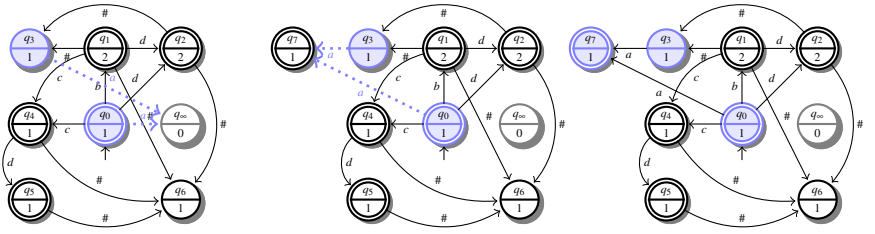
*(i) Reading of item d* *(ii) Creating state $q_5$, and moving reachable transitions* *(iii) Final automaton*

**Fig. 8** Reading and insertion of item *d* (transaction 2)



*(i) Reading of item #* *(ii) Creating state $q_6$, and moving reachable transitions.* *(iii) Final automaton*

**Fig. 9** End of processing transactions 2



*(i) Reading of item a* *(ii) Creating state $q_7$, and moving reachable transitions.* *(iii) Final automaton*

**Fig. 10** Reading and insertion of item *a* (transaction 3)

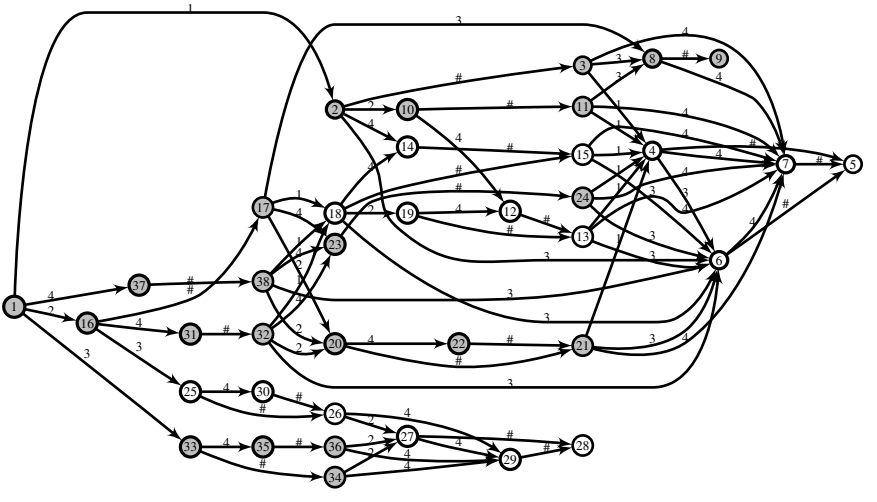


**Fig. 11** This is the resulting automaton generated by SPAMS, indexing all frequent sequential patterns of the statistical cover ($\theta = 0.4$): the filled states have a support equal or greater than $\theta$, while the white states have a support belonging to $[\theta - \varepsilon; \theta]$.

After processing Table 1 as a stream database, the resulting automaton has 38 states and 80 transitions, and contains 233 sequential patterns: this automaton indexes sequential patterns whose support is equal or greater than the statistical support threshold $\theta - \varepsilon$ (*cf.* Fig. 11). By traversing the automaton, we can extract the sequential patterns whose support is strictly greater than $\theta$. In this case, 12 states and 19 transitions are used to index the corresponding sequential patterns, *i.e.* 19 sequential patterns (*cf.* Table 2).

**Table 2** Set of sequential patterns extracted by SPAMS using Table 1 as stream database ($\theta = 0.4$)

| | | | | |
|---|---|---|---|---|
| $<(1)>$:2 | $<(2)>$:3 | $<(2)(4)>$:2 | $<(2,4)(4)>$:2 | $<(4)(2)>$:2 |
| $<(1)(3)>$:2 | $<(2)(2)>$:2 | $<(2,4)>$:2 | $<(3)>$:3 | $<(4)(2,4)>$:2 |
| $<(1,2)>$:2 | $<(2)(2,4)>$:2 | $<(2,4)(2)>$:2 | $<(3,4)>$:2 | $<(4)(4)>$:2 |
| $<(1,2)(3)>$:2 | $<(2)(3)>$:2 | $<(2,4)(2,4)>$:2 | $<(4)>$:2 | |

### 4.2.4 SPAMS Pseudo-code

In the following, we present the pseudo-code of our algorithm. In Section 4.2.2, the module INSERT is the subject of a detailed explanation from which it is easy to deduce the pseudo-code. It's the same for the module NEXT. Thus, we choose to present only the pseudo-code of the main module of our algorithm as well as that of the module PRUNE (*cf.* Algorithms 1 & 2).

**Algorithm 1.** MAIN()

**Data**: $Stream$, $\theta$
**Result**: $\text{SPA}_\theta$
**begin**
  Create two states $q_0$ and $q_\infty$ : $\mathcal{Q} \longleftarrow \{ q_0, q_\infty \}$
  $\mathcal{T} \longleftarrow \emptyset$
  $cid \longleftarrow NULL$
  $tid \longleftarrow NULL$
  $\mathcal{C} \longleftarrow \emptyset$
  $\mathcal{C}_{q_0} \longleftarrow \emptyset$
  $\mathcal{C}_{q_\infty} \longleftarrow \emptyset$
  $\delta \longleftarrow 0.01$
  $minSup \longleftarrow 0$
  **for each** $(cid', tid', \alpha) \in Stream$ **do**
    **if** $(cid \neq cid')$ **or** $(tid \neq tid')$ **then**
      NEXT($cid$)
      $cid \longleftarrow cid'$
      $tid \longleftarrow tid'$
    INSERT($\alpha$, $cid$)
**end**

## 5 Experimental Results

We have now designed a great number of performance tests in order to highlight our algorithm efficiency. We have used a SPAMS implementation in C++, using the Standard Template Library (STL) and the ASTL (Maout, 1997) library, compiled

**Algorithm 2.** PRUNE()

**Data**: $s'$, $\alpha$, $\mathcal{T}^r$, $cid$
**begin**
  **for each** $s \xmapsto{\alpha} s' \in \mathcal{T}$ **do**
    Delete the transition $s \xmapsto{\alpha} s' : \mathcal{T} \longleftarrow \mathcal{T} \setminus \left\{ s \xmapsto{\alpha} s' \right\}$
    **if** $s \xmapsto{\alpha} s' \in \mathcal{T}^r$ **then**
      $\mathcal{T}^r \longleftarrow \mathcal{T}^r \setminus \left\{ s \xmapsto{\alpha} s' \right\}$
  **for each** $s' \xmapsto{\beta} s'' \in \mathcal{T}$ **do**
    PRUNE($s''$, $\beta$, $\mathcal{T}^r$, $cid$)
  $\mathcal{Q}_{cid} \longleftarrow \mathcal{Q}_{cid} \setminus \{ s' \}$
  **for each** $cid' \in \mathcal{C}_{s'}$ **do**
    $\mathcal{Q}_{cid'} \longleftarrow \mathcal{Q}_{cid'} \setminus \{ s' \}$
  Delete the set $\mathcal{W}_{s'}$
  Delete the state $s' : \mathcal{Q} \longleftarrow \mathcal{Q} \setminus \{ s' \}$
**end**

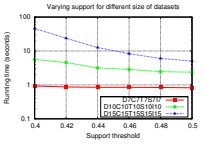
with the option -03 of the g++ compiler on a 700MHz Intel Pentium(R) Core2 Duo PC machine with 4G memory, running Linux Debian Lenny.

Several experiments have been carried out in order to test the efficiency of our approach. Empirical experiments were done on synthetic datasets (*cf.* Table 3) generated by the IBM data generator in Srikant and Agrawal (1996).

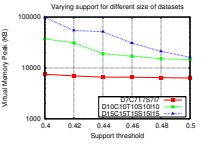
**Table 3** Parameters used in datasets generation

| Symbols | Meaning |
|---|---|
| D | Number of customers in 000s |
| C | Average number of transactions per customer |
| T | Average number of items per transaction |
| N | Number of different items in 000s |
| S | Average length of maximal sequences |

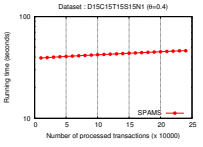
We illustrate on Figs. 12-(i), 12-(ii) the time and the memory consumption performances of SPAMS, for different support values, on small medium and large datasets, respectively *D7C7T7S7N1*, *D10C10T10S10N1* and *D15C15T15S15N1*. Figures 12-(iii), 12-(iv), 12-(v), 12-(vi) represent the evolution of the running time, the memory and the number of customers in relation to the number of transactions on the dataset *D15C15T15S15N1*, with a fixed support value of *40%*. Figure 12-(viii) illustrates that the statistical support used tends to the support threshold $\theta$ during the insertion of new transactions, which reduce the $(\theta - \varepsilon)$-frequent patterns



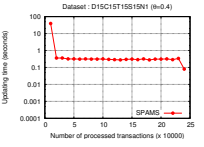

*(i) Running time*



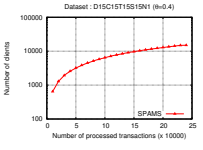

*(ii) Memory consumption*



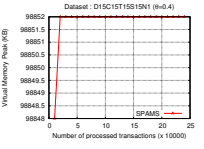

*(iii) Running time on D15C15T15S15N1*



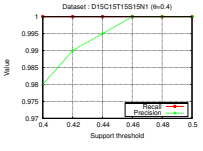

*(iv) Updating time on D15C15T15S15N1*



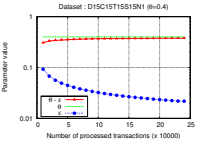

*(v) Number of clients on D15C15T15S15N1*




*(vi) Memory consumption on D15C15T15S15N1*




*(vii) Recall and precision on D15C15T15S15N1*




*(viii) Evolution of the statistical support on D15C15T15S15N1*

**Fig. 12** Self performance evaluation of SPAMS over small, medium and large datasets

of the statistical cover. To calculate the $\varepsilon$ parameter (see section 3), we have chosen the value of *0.01* for the statistical risk parameter $\delta$. These experiments show that we have found a satisfactory compromise between time performances, memory consumption and the quality of the mining results in recall as well as in precision (*cf.* Fig. 12-(vii)). They also show the applicability and the scalability of the SPAMS algorithm for mining data streams.

## 6 Conclusion

In this paper, we bring an original contribution by proposing a new one-pass algorithm, named SPAMS, enabling the building and the maintaining of an automaton data structure: the SPA, which indexes the frequent sequential patterns in a data stream. The SPA is built from scratch and is updated on the volley, as a new transaction is inserted. The current frequent patterns can be output in real time based on

any user's specified thresholds. Thus, the SPA is a very informative and flexible data structure, well-suited for mining frequent sequential patterns in data streams. With the SPA, our contribution opens a promising gateway, by using an automaton as a data structure for mining frequent sequential patterns in data streams. Furthermore, taking into account the characteristics of data streams, we propose a well-suited method, said to be approximate, since we can provide near optimal results with a high probability, while maintaining satisfactory performances of the SPAMS algorithm. Experimental studies show the scalability and the applicability of the SPAMS algorithm. In the future, we will examine how to extend this work to mine closed sequential patterns on sliding windows.

## References

Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: 20th VLDB Conf. (1994)

Ayres, J., Flannick, J., Gehrke, J., Yiu, T.: Sequential pattern mining using a bitmap representation (2002)

Chang, J.H., Lee, W.S.: Efficient mining method for retrieving sequential patterns over online data streams. J. Inf. Sci. 31(5), 420–432 (2005), http://dx.doi.org/10.1177/0165551505055405

Garofalakis, M.N., Gehrke, J., Rastogi, R.: Querying and mining data streams: you only get one look a tutorial. In: Franklin, M.J., Moon, B., Ailamaki, A. (eds.) SIGMOD Conference, p. 635. ACM, New York (2002), http://dblp.uni-trier.de/db/conf/sigmod/sigmod2002.html#GarofalakisGR02

Gouda, K., Hassaan, M., Zaki, M.J.: Prism: A Primal-Encoding Approach for Frequent Sequence Mining. In: ICDM, pp. 487–492. IEEE Computer Society, Los Alamitos (2007), http://dblp.uni-trier.de/db/conf/icdm/icdm2007.html#GoudaHZ07

Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q., Dayal, U., Hsu, M.: FreeSpan: frequent pattern-projected sequential pattern mining. In: KDD, pp. 355–359 (2000)

Hopcroft, J.E., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation. Addison Wesley, Reading (1979)

Laur, P.-A., Symphor, J.-E., Nock, R., Poncelet, P.: Statistical supports for mining sequential patterns and improving the incremental update process on data streams. Intell. Data Anal. 11(1), 29–47 (2007)

Li, H., Chen, H.: GraSeq: A Novel Approximate Mining Approach of Sequential Patterns over Data Stream. In: Alhajj, R., Gao, H., Li, X., Li, J., Zaïane, O.R. (eds.) ADMA 2007. LNCS (LNAI), vol. 4632, pp. 401–411. Springer, Heidelberg (2007), http://dblp.uni-trier.de/db/conf/adma/adma2007.html\#LiC07

Maout, V.L.: Tools to Implement Automata, a First Step: ASTL. In: Wood, D., Yu, S. (eds.) WIA 1997. LNCS, vol. 1436, pp. 104–108. Springer, Heidelberg (1997)

Masseglia, F., Cathala, F., Poncelet, P.: The PSP Approach for Mining Sequential Patterns. In: Zytkow, J.M., Quafafou, M. (eds.) PKDD 1998. LNCS, vol. 1510, pp. 176–184. Springer, Heidelberg (1998)

Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.: PrefixSpan: Mining Sequential Patterns by Prefix-Projected Growth. In: ICDE, pp. 215–224. IEEE Computer Society, Los Alamitos (2001)

Raïssi, C., Poncelet, P.: Sampling for Sequential Pattern Mining: From Static Databases to Data Streams. In: ICDM, pp. 631–636 (2007)

Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)

Zaki, M.J.: SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning 42(1/2), 31–60 (2001)

# Mining Common Outliers for Intrusion Detection

Goverdhan Singh, Florent Masseglia, Céline Fiot, Alice Marascu,
and Pascal Poncelet

**Abstract.** Data mining for intrusion detection can be divided into several sub-topics, among which unsupervised clustering (which has controversial properties). Unsupervised clustering for intrusion detection aims to i) group behaviours together depending on their similarity and ii) detect groups containing only one (or very few) behaviour(s). Such isolated behaviours seem to deviate from the model of normality; therefore, they are considered as malicious. Obviously, not all atypical behaviours are attacks or intrusion attempts. This represents one drawback of intrusion detection methods based on clustering. We take into account the addition of a new feature to isolated behaviours before they are considered malicious. This feature is based on the possible repeated occurrences of the bahaviour on many information systems. Based on this feature, we propose a new outlier mining method which we validate through a set of experiments.

**Keywords:** Intrusion Detection, Anomalies, Outliers, Data Streams.

## 1 Introduction

Intrusion detection is a very important topic of network security and has received much attention (Lee and Stolfo, 1998; Dokas *et al.*, 2002; Lazarevic *et al.*, 2003; Patcha and Park, 2007) since potential cyber threats make the organizations vulnerable. *Intrusion Detection Systems (IDS)* are intended to protect information systems

Goverdhan Singh · Florent Masseglia · Céline Fiot · Alice Marascu
INRIA, 2004 route des lucioles - BP 93, FR-06902 Sophia Antipolis
e-mail: g.singh@iitg.ernet.in, Florent.Masseglia@sophia.inria.fr, celine.fiot@gmail.com, Alice.Marascu@sophia.inria.fr

Pascal Poncelet
LIRMM UMR CNRS 5506, 161 Rue Ada, 34392 Montpellier Cedex 5, France
e-mail: poncelet@lirmm.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 217–234.
springerlink.com 

against intrusions and attacks and are traditionally based on the signatures of known attacks (Roesch, 1998; Barbara *et al.*, 2001). Therefore, new kinds of attacks have to be added to the signature list regularly. The main drawback is that in case of an emerging attack (the recent discovery of a new security hole for instance), the IDS will ignore it since this new attack has not been listed yet in the signature database.

Protecting a system against new attacks, while keeping an automatic and adaptive framework is an important topic in this domain. One solution to this problem can be based on data mining tools. Data mining tools have been used to provide IDS with more adaptive detection approaches of cyber threats (Dokas *et al.*, 2002; Bloedorn *et al.*, 2001; Wu and Zhang, 2003). Among these data mining approaches, predictive models are built to improve the database of signatures used by existing IDS (Wu and Zhang, 2003). Other ones, whose category this chapter refers to, make use of data mining to detect anomalies from which the intrusions are deduced (Lazarevic *et al.*, 2003; Eskin *et al.*, 2002; Chimphlee *et al.*, 2005). The overall principle is generally to build clusters (or classes) of usage and, afterwards, to find the outliers (*i.e.* events that do not belong to any class or cluster corresponding to a normal usage). Actually, outlier detection aims to find records that deviate significantly from a well-defined notion of normality. It has a wide range of applications, such as fraud detection for credit card (Aleskerov *et al.*, 1997), health care (Spence *et al.*, 2001), cyber security (Ertoz *et al.*, 2004) or safety of critical systems (Fujimaki *et al.*, 2005). However, the main drawback of detecting intrusions by means of anomaly (outliers) detection is the high rate of false alarms. In both cases (building a model or detecting outliers) an alarm can indeed be triggered because of a new kind of usages that has never been seen before; so it is considered abnormal. Considering the large amount of new usage patterns emerging in the Information Systems, even a weak percentage of false positive gives a very large amount of spurious alarms that would be overwhelming for the analyst. Reducing the rate of false alarms is thus crucial for a data mining based intrusion detection system in a real-world environment.

Therefore, the goal of this chapter is to propose an intrusion detection algorithm based on the analysis of usage data coming from multiple partners in order to reduce the number of false alarms. Our main idea is that a new usage is likely to be related to the context of the information system on which it occurs (so it should only occur on this system). Meanwhile, when a new security hole has been found on a system, the hackers would use it in as many information systems as possible. Thus a new anomaly occurring on two (or more) information systems is rather an intrusion attempt than a new kind of usage. Let us consider $A_x$, an anomaly detected in the usage of web site $S_1$ corresponding to a php request on the staff directory for a new employee: John Doe, who works in room 204, floor 2, in the R&D department. The request has the following form: `staff.php?FName=John\&LName=Doe\&room=204\&floor=2\&Dpt=RD`. This new request, due to the recent recruitment of John Due in this department, should not be considered as an attack.

Let us now consider $A_y$, an anomaly corresponding to a real intrusion. $A_y$ is caused by a security hole of the system (for instance a php vulnerability) and might, for instance, look like: `staff.php?path=../etc/passwd%00`. In this request, one can see that the parameters are not related to the data accessed by the

php script, but rather to a security hole discovered on the *staff* script. If two or more firms use the same script (say, a directory requesting script bought from the same software company) then the usage of this security hole is certainly repeated from one system to another and the request having parameter `path=../etc/passwd%00` will be the same for all the victims.

We propose to provide the end-user with a method that has only one parameter: $n$, the number of desired alarms. Based on the analysis of the usage data coming from the different partners, our algorithm will detect $n$ common outliers they share. Such common outliers are likely to be true attacks and will trigger an alarm. In a real-world application of this technique, privacy preserving will be a major issue in order to protect partners' data. We focus on clustering and outlier detection techniques in a distributed environment. However, privacy issues in our framework are currently being studied.

The chapter is organized as follows. In Section 2 we present the motivation of this approach and our general framework. Section 3 gives an overview of existing works in this domain. Section 4 presents COD, our method for detecting outliers and triggering true alarms. Eventually, our methods is tested through a set of experiments in Section 5 and Section 6 gives the conclusion.

## 2 Motivation and General Principle

In this section, we present the motivation of our work, based on the main drawbacks of existing anomaly-based methods for intrusion detection and we propose COD, a new algorithm for comparing the anomalies on different systems.

### *2.1 Motivation*

Anomaly-based IDS (Eskin *et al.*, 2002; Chimphlee *et al.*, 2005) can be divided into two categories: semi-supervised and unsupervised. Semi-supervised methods use a model of "normal" behaviours on the system. Every behaviour that is not considered as normal is an anomaly and should trigger an alarm. Unsupervised methods do not use any labelled data. They usually try to detect outliers based on a clustering algorithm.

Obviously, *anomaly-based IDS will suffer from a very high number of false alarms since a new kind of behaviour will be considered as an anomaly (and an attack)*. Actually, anomalies are usually extracted by means of outlier detection, which are records (or sets of records) that significantly deviate from the rest of the data. Let us consider, for instance, a dataset of 1M navigations collected during one week on the Web site of a company (say, a search engine). In this case, a false alarm rate of 2% represents 20,000 alarms that could be avoided. Reducing the number of false alarms is linked to the detection rate. However, the primary motivation of our work is to lower the rate of false alarms. We propose to improve the results of

unsupervised IDS by means of a collaborative framework involving different network-based systems. Section 3 gives an overview of existing IDS based on the principles presented above and on the existing collaborative IDS. However, to the best of our knowledge, our proposal is the first unsupervised IDS using the common anomalies of multiple partners in order to detect the true intrusion attempts. The main idea of our proposal is that multiple partners do not share the same data, but they share the same systems (the Web server can be Apache or IIS, the data server can run Oracle, the scripts accessing the data can be written in PHP or CGI, etc). When a security hole has been found in one system (for instance, a php script with specific parameters leading to privileged access to the hard drive), then this weakness will be the same for all the partners using the same technology. Our goal is to reduce the rate of false alarm based on this observation, as explained in section 2.2.

## 2.2 General Principle

In this chapter we present COD (Common Outlier Detection) a framework and algorithm intended to detect the outliers shared by at least two partners in a collaborative IDS. Outliers are usually small clusters. Some outlier detection methods are presented in section 3. As explained in section 2.1 the main drawback of clustering-based IDS is that they obtain a list of outliers containing both normal atypical usages and real intrusions; so the real intrusions are not separated from the normal atypical behaviors. Our goal is to compare such outlier lists from different systems (based on a similar clustering, involving the same distance measure). If an outlier occurs for at least two systems, then it is considered as an attack. COD is based on the following assumptions:

- An intrusion attempt trying to find a weakness of a script will look similar for all the victims of this attack.
- This attack will be quite different from a normal usage of the system.
- The distance between normal usage patterns will be low, which makes it possible for most of them to group in large clusters (remaining unclassified normal patterns are the false alarms of methods presented in Section 3).

We propose to detect intrusion attempts among the records of a Web server, such as an Apache access log file. For each access on the Web site, such a file keeps record of: the IP, the date, the requested URL and the referrer (as well as other information less important in our situation). Our main idea is that the anomalies occuring on two different systems, are highly probable to be attacks. Let us detail the short illustration given in section 1 with $A_x$, an anomaly that is not an attack on site $S_1$. $A_x$ is probably a context based anomaly, such as a new kind of usage specific to $S_1$. Therefore, $A_x$ will not occur on $S_2$. As an illustration, let us consider a php request on the staff directory for a new employee: John Doe, who works in room 204, floor 2, in the R&D department. The request will have the following form: `staff.php?FName=John\&LName=Doe\&room=204\&floor=2\&Dpt=RD`. This new request, due to the recent recruitment of John Due in this department should not be considered as an attack.

However, with an IDS based on outlier detection, it is likely to be considered as an intrusion, since it is not an usual behaviour.

Let us now consider $A_y$, an anomaly corresponding to a true intrusion. Let us consider that $A_y$ is based on a security hole of the system (for instance a php vulnerability). Then $A_y$ will be the same for every site attacked through this weakness. For instance, a php request corresponding to an attack might look like: `staff.php?path=/../etc/passwd%00`. In this request, one can see that the parameters are not related to the data accessed by the php script, but rather to a security hole that has been discovered on the *staff* script that returns passwords. If this script is provided by a software company to many firms, the usage of this security hole will repeatedly occur on different sites and the request having parameter `path=/../etc/passwd%00` will be the same for all the victims.

For clarity of presentation we present our framework on the collaboration of two Web sites, $S_1$ and $S_2$ and we consider the requests that have been received by the scripts of each site (cgi, php, sql, etc). Our goal is to perform a clustering on the usage patterns of each site and to find the common outliers. However, that would not be enough to meet the second constraint of our objective: the request of only one parameter, $n$, the number of alarms to return. Our distance measure (presented in section 4.1) will allow normal usage patterns to be grouped together rather than to be mixed with intrusion patterns. Moreover, our distance measure has to distinguish an intrusion pattern from normal usage patterns and also from other intrusion patterns (since different intrusion patterns will be based on a different security hole and will have very different characteristics). Our algorithm performs successive clustering steps for each site. At each step we check the potentially matching outliers between both sites. The clustering algorithm is agglomerative and depends on the maximum distance (MD) requested between two objects.

Let us consider that $n$, the desired number of alarms, is set to 1 and the usage patterns are distributed as illustrated in figure 1. Let us also consider that, for these sites, cluster $A$ at step 1 is the only one corresponding to an intrusion attempt. For the first step, MD is initialized with a very low value, so the clusters will be as tight and small as possible. Then we check correspondences between outliers of $S_1$ and $S_2$. Let us consider the clustering results on $S_1$ and $S_2$ at step 1 in figure 1. There are two matching outliers between both sites ($A$ and $B$). That would lead to 2 alarms (just one of the alarms being true) which represents more than the number of alarms desired by the user. We thus have to increase the clustering tolerance (*i.e.* increase MD) so that bigger clusters could be built. The clusters configuration at step $n$ is illustrated in figure 1. The only common outlier is $A$, which corresponds to the intrusion attempt. Furthermore, this will trigger one alarm, as desired by the user, and there is no need to continue increasing MD until step $m$.

As explained in section 1, we want to propose an algorithm that requires only one parameter, $n$, the maximum number of alarms desired by the end-user. Actually, this work is intended to explore the solutions for monitoring a network in real time. Then, the potential alarms will be triggered at each step of the monitoring (for

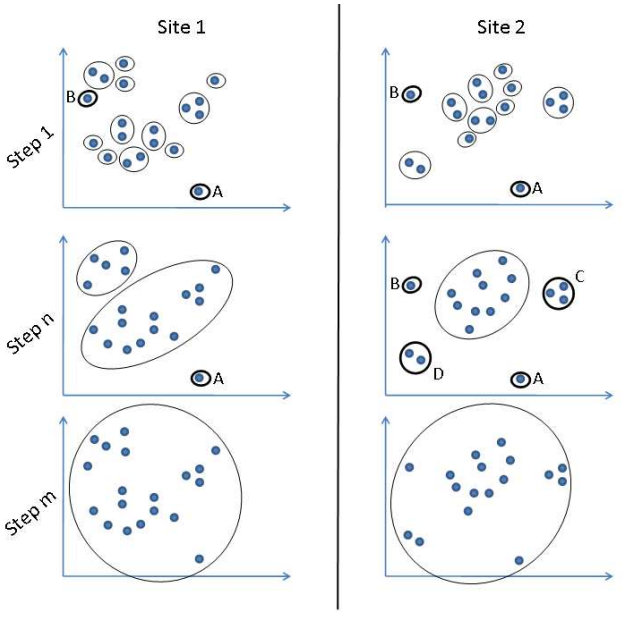



**Fig. 1** Detection of common outliers in the usage patterns of two Web sites

instance with a frequency of one hour). A first batch of usage data is clustered on each site and $n$ alarms are triggered. Depending on the number of true or false alarms, the user might want to adjust $n$ for the next step, until no false alarm is returned. Our assumption is that the intrusions are situated in the first part of the list representing the common outliers sorted by similarity.

Obviously, such a framework requires a good privacy management of each partner's data. This is a very important issue in our framework and we propose solutions in this volume (Verma *et al.*, 2010).

Our challenge is to reply to important questions underlying our method ; what is the distance between two usage patterns? How to separate clusters in order to give the list of outliers? How to detect common outliers?

Our main algorithm, corresponding to the framework presented in this section, is given in section 4.1. Our distance measure and our clustering algorithm are given in section 4.2. As explained in section 4.3 our outlier detection method is parameterless, thanks to a wavelet transform on the cluster distribution. In contrast to most previous methods (Jin *et al.*, 2001; Zhong *et al.*, 2007; Portnoy *et al.*, 2001; Joshua Oldmeadow *et al.*, 2004) it neither requires a percentage of clusters nor depends on a top-n parameter given by the user. The correspondance between outliers of $S_1$ and $S_2$ also has to be parameterless. As explained in section 4.4 it will find the clusters that are close enough to trigger an alarm automatically.

## 3 Related Work

Atypical data discovery is a really active research topic for now few decades. The problem of finding in databases patterns that deviate significantly from a well-defined notion of normality, also called *outlier detection*, has indeed a wide range of applications, such as fraud detection for credit card (Aleskerov *et al.*, 1997), health care (Spence *et al.*, 2001), cyber security (Ertoz *et al.*, 2004) or safety of critical systems (Fujimaki *et al.*, 2005).

Over time many outlier detection techniques have been developped, leading to an important number of surveys and review articles (Hodge and Austin, 2004; Chandola *et al.*, 2008). Some of them focus on the topic of outlier detection in the context of intrusion detection in computer networks (Lazarevic *et al.*, 2003; Patcha and Park, 2007). We focus on this specific area and we propose an *unsupervised* anomaly-based detection system. On the opposite to *semi-supervised* anomaly detection systems, consisting of describing normal behaviours to detect deviating patterns (Marchette, 1999; Wu and Zhang, 2003; Vinueza and Grudic, 2004), *unsupervised* techniques do not require a preliminary identification of the normal usage by a human expert. Our application will thus be more usable in a real-world context.

Statistic community has quite extensively studied the concept of outlyingness (Barnett and T. Lewis, 1994; Markou and Singh, 2003; Kwitt and Hofmann, 2007). Statistical approaches construct probability distribution models where outliers are objects of low probability (Rousseeuw and Leroy, 1996; Billor *et al.*, 2000; Lee and Xiang, 2001) However, within the context of intrusion detection, dimensionality of data is high. Therefore, to improve overall performance and accuracy, it has become necessary to develop data mining algorithms using the whole data distribution as well as most of data features (Knorr and Ng, 1998; Breunig *et al.*, 2000; Aggarwal and Yu, 2001).

Most of these approaches are based on clustering-based outlier detection algorithms (Ester *et al.*, 1996; Portnoy *et al.*, 2001; Eskin *et al.*, 2002; He *et al.*, 2003; Papadimitriou *et al.*, 2003). Such techniques rely on the assumption (Chandola *et al.*, 2008) that normal points belong to large and dense clusters while anomalies (or outliers, atypical instances) either do not belong to any clusters (Knorr and Ng, 1998; Ramaswamy *et al.*, 2000; Duan *et al.*, 2006) or form very small (or very sparse) clusters (Otey *et al.*, 2003; Chimphlee *et al.*, 2005; Pires and Santos-Pereira, 2005; Fan *et al.*, 2006). In other words anomaly detection consists in identifying the elements situated very far from significant clusters; these elements can be either isolated or grouped in small clusters. Depending on the approach, the number of parameters required to run the algorithm can be high and will lead to different outliers. To avoid this, some works return a ranked list of potential outliers and limit the number of parameters to be specified (Ramaswamy *et al.*, 2000; Jin *et al.*, 2001; Fan *et al.*, 2006).

However, the major drawback of all the anomaly-based intrusion detection techniques is represented by the very high number of false alarms triggered. On the contrary, misuse techniques (*i.e.* approaches that detect elements similar to well-known

malicious usage) will precisely detect attacks, but they will miss every intrusion that differs from those already known attack signatures. Therefore, some works proposed collaborative frameworks in order to improve the performance and both true and false alarm rates (Valdes and Skinner, 2001; Locasto *et al.*, 2004; Yegneswaran *et al.*, 2004). These approaches rely on a propagation process in a distributed IDS IP blacklist after individual misuse or anomaly detection. This communication can also lead to more accurate results and it does not allow the system to uncover totally unknown attacks or to avoid high false alarm rates.

For these reasons we propose an anomaly detection approach that uses collaboration between systems in order to discriminate attacks from emerging or novel usage behaviours, thus leading to a reduced number of false alarms. To the best of our knoweldge, this is the first proposal for such an IDS.

## 4 COD: Common Outlier Detection

The principle of COD consists in successive clusterings on usage patterns of different partners sites, until the number of common outliers become equal to the number of alarms desired by the user. We present here an algorithm designed for two information systems. Extending this work to more than two systems would require a central node coordinating the comparisons and triggering the alarms, or a peer-to-peer communication protocol. This is not the goal of this chapter, since we want to focus on proposing solutions to the following issues:

- Clustering the usage patterns of a Web site with different levels of MD.
- Proposing a distance measure adapted to intrusion detection.
- Identifying the outliers after having clustered the usage patterns.
- Comparing the outliers given by each partner.

Our objects are the parameters given to script files in the requests received on a Web site. In other words, the access log file is filtered and we only keep lines corresponding to requests with parameters to a script. For each such line, we separate the parameters and for each parameter we create an object. Let us consider, for instance, the following request: `staff.php?FName=John&LName=Doe`. The corresponding objects are $o_1$ =`John` and $o_2$ =`Doe`. Once the objects are obtained from the usage data of multiple Web sites, COD is applied and gives their common outliers.

### *4.1 Main Algorithm*

As explained in section 2.2, COD processes the usage patterns of both sites step by step. For each step, COD gives a set of clusters and analyzes them in order to detect the intrusions. The pseudo-code of COD is given in figure 2. First, *MD* is set to obtain very tight and numerous clusters (a very short distance is allowed between two objects in a cluster). Afterwards, *MD* is relaxed by 0.05 step by step in

order to increase the size of the resulting clusters and to decrease their number and to obtain less alarms. When the number of alarms desired by the user is reached, COD ends.

**Algorithm** Cod
**Input:** $U_1$ and $U_2$ the usage patterns of sites $S_1$ and $S_2$
and $n$ the number of alarms.
**Output:** $I$ the set of clusters corresponding
to malicious patterns.

1. $MD \leftarrow 0$ ;
2. $MD \leftarrow MD + 0.05$ ;
3. $C_1 \leftarrow Clustering(U_1, MD)$ ;
   $C_2 \leftarrow Clustering(U_2, MD)$ ;
4. $O_1 \leftarrow Outliers(C_1)$ ; $O_2 \leftarrow Outliers(C_2)$ ;
5. $I \leftarrow CommonOutliers(O_1, O_2, MD)$ ;
6. If $|I| \leq n$ then return $I$ ;
7. If $MD = 1$ then return $I$ ; // No common outlier
8. Else return to step 2 ;

**End algorithm** Cod

**Fig. 2** Algorithm Cod

## *4.2 Clustering*

COD clustering algorithm (given in figure 3) is based on an agglomerative principle. The goal is to increase the volume of clusters by adding candidate objects, until the Maximum Distance (MD) is broken (*i.e.* there is one object $o_i$ in the cluster such that the distance between $o_i$ and the candidate object $o_c$ is greater than MD).

**Distance between objects**. We consider each object as a sequence of characters. Firstly, we need to introduce the notion of subsequence in definition 1.

**Definition 1.** Let $S = s_1, s_2, \ldots, s_n$ be a sequence of characters having length $n$, a *subsequence* is a subset of the characters of $S$ with respect to their original order. More formally, $V = v_1, v_2, \ldots, v_k$, having length $k \leq n$, is a subsequence of $S$ if there exist integers $i_1 < i_2 < \ldots < i_k$ such that $s_1 = v_{i_1}$, $s_2 = v_{i_2}$, $\ldots$ $s_k = v_{i_k}$.

Our distance is based on the longest common subsequence (LCS), as described in definition 2.

**Definition 2.** Let $s_1$ and $s_2$ be two sequences. Let $LCS(s_1, s_2)$ be the length of the longest common subsequence corresponding to $s_1$ and $s_2$. The *distance* $d(s_1, s_2)$ between $s_1$ and $s_2$ is defined as follows:

$$d(s_1, s_2) = 1 - \frac{2 \times LCS(s_1, s_2)}{|s_1| + |s_2|}$$

*Example 1.* Let us consider two parameters $p_1$=`intrusion` and $p_2$=`induction`. The LCS between $p_1$ and $p_2$ is $L$=`inuion`. $L$ has length 6 and the dissimilarity between $p_1$ and $p_2$ is $d = 1 - \frac{2 \times L}{|p_1|+|p_2|} = 33.33\%$. Which also means a similarity of 66.66% between both parameters.

**Centre of clusters.** When an object is inserted into a cluster we maintain the centre of this cluster, as it will be used in the CommonOutliers algorithm described in Figure 6. The centre of a cluster $C$ is the LCS among all the objects in $C$. When object $o_i$ is added to $C$, its center $C_c$ is updated. The new value of $C_c$ is the LCS between the current value of $C_c$ and $o_i$.

**Algorithm** Clustering
**Input:** $U$, the usage patterns
and $MD$, the Maximum Distance.
**Output:** $C$, the set of as large clusters as possible,
respecting $MD$.

1. Build $M$, the distance matrix between each pattern in $U$;
2. $\forall p \in M, Neighbours_p \leftarrow$ sorted list of neighbours for $p$ (the first usage pattern in the list of $p$ is the closest to $p$).
3. $DensityList \leftarrow$ sorted list of patterns by density ;
4. $i \leftarrow 0$ ; $C \leftarrow \emptyset$ ;
5. $p \leftarrow$ next unclassified pattern in $DensityList$ ;
6. $i++$ ; $c_i \leftarrow p$ ;
7. $C \leftarrow C + c_i$ ;
8. $q \leftarrow$ next unclassified pattern in $Neighbours_p$ ;
9. $\forall o \in c_i$, If $distance(o,q) > MD$ then return to step 5 ;
10. add $q$ to $c_i$ ;
11. $C_c \leftarrow LCS(C_c, q)$ ; //$C_c$ is the center of $C$
12. return to step 8 ;
13. If unclassified patterns remain then return to step 5 ;
14. return $C$ ;

**End algorithm** Clustering

**Fig. 3** Algorithm Clustering

## *4.3 Wavelet-Based Outlier Detection*

Most previous work in outlier detection requires a parameter (Jin *et al.*, 2001; Zhong *et al.*, 2007; Portnoy *et al.*, 2001; Joshua Oldmeadow *et al.*, 2004), such as the percentage of small clusters that should be considered as outliers, or the top-$n$ outliers. Their key idea is generally to sort the clusters by size and/or tightness. We consider that the clusters given by COD are as tight as possible. In order to extract outliers, our idea is to sort the clusters by size and to consider the smallest clusters as outliers. Therefore, the problem is how to separate the "big" clusters from the "small" ones, small and big being subjective measurements. Our solution is based on an analysis

of the clusters distribution, after the sorting operation. An example of general distribution of clusters is illustrated by Figure 4 (screenshot made with our real data). In Marascu and Masseglia (2009), the authors proposed to cut down the distribution by means of a wavelet transform. This technique is illustrated by figure 4, where the *y* axis stands for the size of the clusters, whereas their index in the sorted list is represented on *x*, and the two plateaux allow separating small and big clusters. With a prior knowledge on the number of plateaus (we want two plateaus, the first one standing for small clusters, or outliers, and the second one standing for big clusters) we can cut the distribution in a very effective manner. Actually, each cluster mapped to the first plateau will be considered as an outlier.

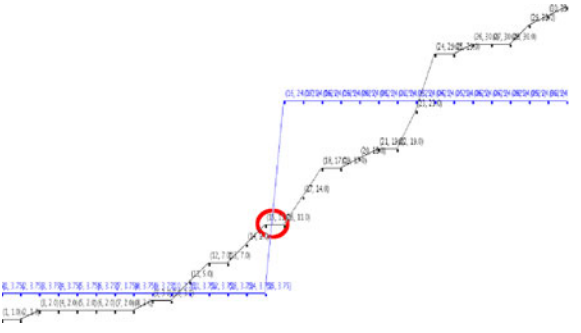


**Fig. 4** Detection of outliers by means of Haar Wavelets

The advantages of this method, for our problem, are illustrated in figure 5. Depending on the distribution, wavelets will give different indices (where to cut). For instance, with few clusters having the maximum size (see graph with solid lines from figure 5 ), wavelets cut the distribution in the middle. Meanwhile, with a large number of big clusters (see graph with dashed lines from figure 5), wavelets increase

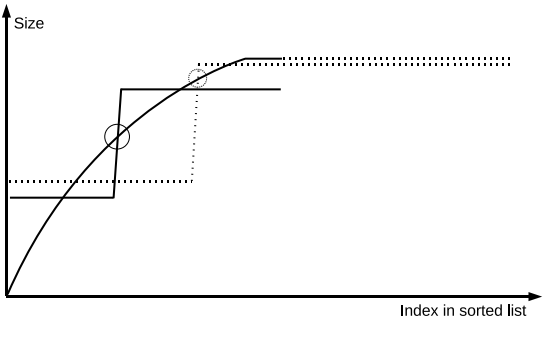



**Fig. 5** Self-adjusting detection of outliers

accordingly the number of clusters in the little plateau (taking into account the large number of big clusters).

### *4.4 Comparing Outliers*

Since we aim to propose an algorithm requiring only one parameter (the number of alarms), we must avoid introducing a similarity degree for comparing two lists of outliers. For this purpose, our algorithm (given in figure 6) uses the centre of outliers. For each pair of outliers, CommonOutliers calculates the distance between the centers of these outliers. If this distance is below the current MD (C.f. Subsection 4.2), then we consider these outliers similar and we add them to the alarm list. The centre of an outlier is the LCS of all the objects contained by the outlier. The distance between two outliers is given by the LCS between their centers.

**Algorithm** CommonOutliers
**Input:** $O_1$ and $O_2$, two lists of outliers
and $MD$, the maximum distance.
**Output:** $A$, the list of alarms (common outliers).

1. $A \leftarrow \emptyset$
2. $\forall i \in O_1$ do
3. $\quad \forall j \in O_2$ do
4. $\quad\quad centre_i \leftarrow centre(i)$ ;
5. $\quad\quad centre_j \leftarrow centre(j)$ ;
6. $\quad\quad$ If $distance(centre_i, centre_j) < MD$
$\quad\quad\quad$ Then $A \leftarrow A + i \cup j$ ;
7. $\quad$ done ;
8. done ;
9. Return $A$ ;

**End algorithm** CommonOutliers

**Fig. 6** Algorithm CommonOutliers

## 5 Experiments

The goal of this section is to offer an analysis of our results (*i.e.* the number of outliers and of true intrusions and the kind of intrusions we have detected).

### *5.1 Datasets*

Our datasets come from two different research organizations: Inria Sophia-Antipolis and IRISA. We have analyzed their Web access log files from March 1 to March 31. The first log file represents 1.8 Gb of rough data and the total number

of objects (parameters given to scripts) is 30,454. The second log file represents 1.2 Gb of rough data and the total number of objects is 72,381. COD was written in Java and C++ on a PC (2.33GHz i686) running Linux with 4Gb of main memory. The parameters generated automatically by the scripts were removed from the datasets since they cannot correspond to attacks (for instance "`publications.php?Category=Books`"). This can be done by listing all the possible combinations of parameters in the scripts of a Web site.

## 5.2 Detection of Common Outliers

As described in Section 2.2, COD proceeds by steps and slowly increases the value of MD, which stands for a tolerance value used in the clustering process. In our experiments, MD has been increased by steps of 0.05 from 0.05 to 0.5. For each step the measures are reported in table 1. The meaning of each measure is as follows. $C_1$ (resp $C_2$) is the number of clusters in site 1 (resp. site 2). $O_1$ (resp. $O_2$) is the number of outlying objects in site 1 (resp. site 2). $\%_1$ (resp $\%_2$) is the fraction of outlying objects on the number of objects in site 1 (resp. site 2). For instance, when MD is set to 0.3, for site 1 we have 6,940 clusters (built from the 30,454 objects) and 5,607 outlying objects, which represents 18.4% of the total number of objects in site 1. *COD* is the number of common outliers between both sites and $\%_{FP}$ is the percentage of false positive alarms among the common outliers. For instance, when MD is set to 0.05, we find 101 alarms among which 5 are false (which represents 4.9%). One first observation is that outliers cannot be directly used to trigger alarms. Obvioulsy, it is not realistic to check a number of alarms as high as 5,607, even in one month. Meanwhile, the results of COD show its ability to separate malicious behaviours from normal usages.

Our false positive patterns correspond to rare normal requests common to both sites. For instance, on the references interrogation script of Inria Sophia-Antipolis, a user might request the papers of "John Doe" and the request would look like `publications.php?FName=John\&LName=Doe`. If another user requests the papers of "John Rare" from the Web site of IRISA, the request would be

**Table 1** Results on real data

| Measure \MD | 0.05 | 0.1 | 0.15 | 0.2 | 0.25 | 0.3 | 0.35 | 0.4 | 0.45 | 0.5 |
|---|---|---|---|---|---|---|---|---|---|---|
| $C_1$ | 14165 | 11922 | 10380 | 8974 | 7898 | 6940 | 6095 | 5390 | 4863 | 4316 |
| $O_1$ | 13197 | 10860 | 8839 | 7714 | 6547 | 5607 | 5184 | 4410 | 3945 | 3532 |
| $\%_1$ | 43.3% | 35.6% | 29% | 25.3% | 21.5% | 18.4% | 17% | 14.4% | 12.9% | 11.6% |
| $C_2$ | 37384 | 30456 | 25329 | 21682 | 19080 | 16328 | 14518 | 12753 | 10984 | 9484 |
| $O_2$ | 35983 | 27519 | 24032 | 20948 | 18152 | 14664 | 12738 | 11680 | 10179 | 8734 |
| $\%_2$ | 49.6% | 37.9% | 33.1% | 28.9% | 25% | 20.2% | 17.5% | 16.1% | 14% | 12.1% |
| *COD* | 101 | 78 | 74 | 70 | 67 | 71 | 71 | 85 | 89 | 90 |
| $\%_{FP}$ | 4.9% | 5.12% | 4% | 2.85% | 1.5% | 2.8% | 2.8% | 10.6% | 11.2% | 16.6% |

`biblio.php?FName=John\&LName=Rare` and the parameter "John" would be given as a common outlier and would trigger an alarm. As we can see, $\%_{FP}$ is very low (usually we have at most 5 false alarms in our experiments for both Web sites) compared to the thousands of outliers that have been filtered by COD.

Another deduction from these experiments is that a low MD implies very small clusters and numerous outliers. These outliers are shared by both sites, among which some are false alarms due to rare but common normal usage. When MD increases, the clustering process becomes more agglomerative and alarms are grouped together. In this case, one alarm can cover several ones of the same kind (*e.g.* the case of easter eggs explained further). At the same time, the number of outliers corresponding to normal usage decreases (since they are also grouped together). Eventually, a too large value of MD implies the building of clusters that do not really make sense. In this case, outliers become larger, and the matching criterion would be too tolerant, leading to a large number of matching outliers which includes also the normal usages.

In a streaming environment, one could decide to keep 70 as the number of desired alarms and watch the ratio of false positive alarms. If this ratio decreases, the end-user should increase the number of desired alarms.

## 5.3 Execution Times

**Processing one month of real data:** We want to show that COD is suitable both for off-line and for on-line environments. First, regarding off-line environments, we report in Figure 7 the time responses of COD for the results presented in subsection 5.2. These results have been obtained using real log files corresponding to one month navigations from Inria Sophia-Antipolis and IRISA Websites; this represents approximately 1,8 Bg et 1.2Gb. We consider that preprocessing the log files and

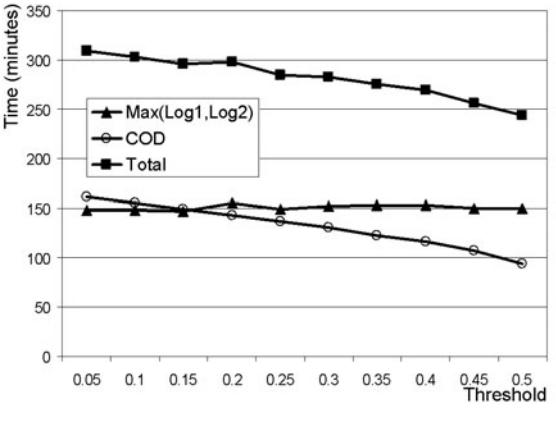



**Fig. 7** Execution times of COD for one month of real data

obtaining the clusters and outliers for each site can be done separately and by different computers. That is why Figure 7 reports the maximum time for Sophia-Antipolis (Log1) and IRISA (Log2) for these three steps (preprocessing, clustering and outlier detection). Foreach user threshold, we also report the execution time of common outlier detection (COD). Eventually, the total time (the addition of preprocessing, clustering, outlier detection and common outlier detection times) foreach threshold is reported. The global time for these two log files (corresponding to one month) is 2819 minutes (the addition of all the total times).

Once this knowledge is obtained (*i.e.* the outliers for each site), when new transactions arrive in the system (new navigations on the site, for instance) we want to extract the outliers for this set of new navigations, and compare them to the existing ones. The time responses obtained for this real-time situation are reported hereafter.

**Time response for one day:** After one day the navigations represent approximately 60 Mo of rough data and 2500 objects in average after preprocessing. Parsing one day of data needs 73 seconds in average. Clustering and outlier detection needs 82 seconds. Common outlier detection requires to compare 715 outliers (average number of outliers for one day) to 21,460 known outliers (average number for one month) over 5 thresholds. The total time of common outlier detection for one day of navigations is 43 minutes.

**Time response for one hour:** The total time for detecting outliers shared by one hour of navigations of the first site, with one month of navigations of the partner site is less than 2 minutes.

### *5.4 A Sample of Our Results*

None of the attacks found in our experiments has been successful on the considered Web sites. However, our security services and our own investigations allow us to confirm the intrusion attempts that have been discovered by our method:

- **Code Injection:** A recent kind of attack aims to inject code in PHP scripts by giving a URL among the parameters. Here is a sample of such URLs detected by COD:
  - http://myweddingphotos.by.ru/images?
  - http://levispotparty.eclub.lv/images?
  - http://0xg3458.hub.io/pb.php?

  Depending on the PHP settings on the victim's Web server, the injected code allows modifying the site. These URLs are directly, automatically and massively given to scripts as parameters through batches of instructions.
- **Passwords:** Another kind of (naive and basic) attack aims to retrieve the password file. This leads to outliers containing parameters like `../etc/password` with a varying number of `../` at the beginning of the parameter. This is probably the most frequent attempt. It is generally not dangerous but shows the effectiveness of our method.

- **Easter Eggs:** This is not really an intrusion but if one adds the code ?=PHPE9568F36-D428-11d2-A769-00AA001ACF42 to the end of any URL that is a PHP page, he will see a (funny) picture on most servers. Also on April 1st (April Fool's Day), the picture will replace the PHP logo on any phpinfo() page. This code (as well as two other ones, grouped into the same outlier) has been detected as a common outlier by COD.

## 6 Conclusion

We have proposed i) an unsupervised clustering scheme for isolating atypical behaviours and ii) a new feature for characterizing intrusions. This new feature is based on the repetition of an intrusion attempt from one system to another. Actually, our experiments show that atypical behaviours (up to several thousands for one day at Inria Sophia-Antipolis) cannot be directly used to trigger alarms since most of them correspond to normal (though atypical) requests. Yet, this very large number of outliers can be effectively filtered in order to find true intrusion attempts (or attacks) if we consider more than one site. In our experiments, by comparing the outliers of two sites, our method kept only less than one hundred alarms, reducing the amount of atypical behaviours up to 0.21%. Eventually, our method guarantees a very low ratio of false alarms, thus making unsupervised clustering for intrusion detection effective and efficient.

**Acknowledgements.** The authors want to thank Laurent Mirtain, the person in charge of intrusion detection of Inria Sophia-Antipolis, for his assistance in identifying attacks in our access log files.

## References

Aggarwal, C.C., Yu, P.S.: Outlier detection for high dimensional data. SIGMOD Records 30(2), 37–46 (2001)

Aleskerov, E., Freisleben, B., Rao, B.: Cardwatch: A neural network based database mining system for credit card fraud detection. In: IEEE Computational Intelligence for Financial Engineering (1997)

Barbara, D., Wu, N., Jajodia, S.: Detecting novel network intrusions using Bayes estimators. In: 1st SIAM Conference on Data Mining (2001)

Barnett, V., Lewis, T. (eds.): Outliers in statistical data. John Wiley & Sons, Chichester (1994)

Billor, N., Hadi, A.S., Velleman, P.F.: BACON: blocked adaptive computationally efficient outlier nominators. Computational Statistics and Data Analysis 34 (2000)

Bloedorn, E., Christiansen, A.D., Hill, W., Skorupka, C., Talbot, L.M.: Data Mining for Network Intrusion Detection: How to Get Started. Technical report, MITRE (2001)

Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. SIGMOD Records 29(2), 93–104 (2000)

Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection - A Survey. ACM Computing Surveys (2008)

Chimphlee, W., Abdullah, A.H., Md Sap, M.N., Chimphlee, S.: Unsupervised Anomaly Detection with Unlabeled Data Using Clustering. In: International conference on information and communication technology (2005)

Dokas, P., Ertoz, L., Kumar, V., Lazarevic, A., Srivastava, J., Tan, P.: Data mining for network intrusion detection. In: NSF Workshop on Next Generation Data Mining (2002)

Duan, L., Xiong, D., Lee, J., Guo, F.: A Local Density Based Spatial Clustering Algorithm with Noise. In: IEEE International Conference on Systems, Man and Cybernetics (2006)

Ertoz, L., Eilertson, E., Lazarevic, A., Tan, P.-N., Kumar, V., Srivastava, J., Dokas, P.: MINDS - Minnesota Intrusion Detection System. In: Data Mining - Next Generation Challenges and Future Directions (2004)

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., Stolfo, S.: A geometric framework for unsupervised anomaly detection: Detecting intrusions in unlabeled data. Applications of Data Mining in Computer Security (2002)

Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density–based algorithm for discovering clusters in large spatial databases with noise. In: 2nd International Conference on Knowledge Discovery and Data Mining (1996)

Fan, H., Zaiane, O.R., Foss, A., Wu, J.: A nonparametric outlier detection for effectively discovering top-N outliers from engineering data. In: Pacific-Asia conference on knowledge discovery and data mining (2006)

Fujimaki, R., Yairi, T., Machida, K.: An approach to spacecraft anomaly detection problem using kernel feature space. In: 11th ACM SIGKDD international conference on Knowledge discovery in data mining (2005)

He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. Pattern Recognition Letters 24 (2003)

Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review 22 (2004)

Jin, W., Tung, A.K.H., Han, J.: Mining top-n local outliers in large databases. In: 7th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 293–298 (2001)

Joshua Oldmeadow, J., Ravinutala, S., Leckie, C.: Adaptive Clustering for Network Intrusion Detection. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 255–259. Springer, Heidelberg (2004)

Knorr, E.M., Ng, R.T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. In: 24th International Conference on Very Large Data Bases, pp. 392–403 (1998)

Kwitt, R., Hofmann, U.: Unsupervised Anomaly Detection in Network Traffic by Means of Robust PCA. In: International Multi-Conference on Computing in the Global Information Technology (2007)

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J.: A comparative study of anomaly detection schemes in network intrusion detection. In: 3rd SIAM International Conference on Data Mining (2003)

Lee, W., Stolfo, S.J.: Data mining approaches for intrusion detection. In: 7th conference on USENIX Security Symposium (1998)

Lee, W., Xiang, D.: Information-Theoretic Measures for Anomaly Detection. In: IEEE Symposium on Security and Privacy (2001)

Locasto, M., Parekh, J., Stolfo, S., Keromytis, A., Malkin, T., Misra, V.: Collaborative Distributed Intrusion Detection. Technical Report CUCS-012-04, Columbia Unviersity Technical Report (2004)

Marascu, A., Masseglia, F.: Parameterless outlier detection in data streams. In: SAC, pp. 1491–1495 (2009)

Marchette, D.: A statistical method for profiling network traffic. In: 1st USENIX Workshop on Intrusion Detection and Network Monitoring, pp. 119–128 (1999)

Markou, M., Singh, S.: Novelty detection: a review - part 1: statistical approaches. Signal Processing 83 (2003)

Otey, M., Parthasarathy, S., Ghoting, A., Li, G., Narravula, S., Panda, D.: Towards nic–based intrusion detection. In: 9th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 723–728 (2003)

Papadimitriou, S., Kitagawa, H., Gibbons, P., Faloutsos, C.: LOCI: fast outlier detection using the local correlation integral. In: 19th International Conference on Data Engineering (2003)

Patcha, A., Park, J.-M.: An overview of anomaly detection techniques: Existing solutions and latest technological trends. Comput. Networks 51 (2007)

Pires, A., Santos-Pereira, C.: Using clustering and robust estimators to detect outliers in multivariate data. In: International Conference on Robust Statistics (2005)

Portnoy, L., Eskin, E., Stolfo, S.: Intrusion detection with unlabeled data using clustering. In: ACM CSS Workshop on Data Mining Applied to Security (2001)

Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. SIGMOD Records 29(2), 427–438 (2000)

Roesch, M.: SNORT (1998), `http://www.snort.org`

Rousseeuw, P., Leroy, A.M. (eds.): Robust Regression and Outlier Detection. Wiley-IEEE (1996)

Spence, C., Parra, L., Sajda, P.: Detection, synthesis and compression in mammographic image analysis with a hierarchical image probability model. In: IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (2001)

Valdes, A., Skinner, K.: Probabilistic Alert Correlation. In: Lee, W., Mé, L., Wespi, A. (eds.) RAID 2001. LNCS, vol. 2212, pp. 54–68. Springer, Heidelberg (2001)

Verma, N., Trousset, F., Poncelet, P., Masseglia, F.: Intrusion Detections in Collaborative Organizations by Preserving Privacy. In: Guillet, F., Ritschard, G., Briand, H., Zighed, D.A. (eds.) Advances in Knowledge Discovery and Management. SCI, vol. 292, pp. 237–250. Springer, Heidellberg (2010)

Vinueza, A., Grudic, G.: Unsupervised outlier detection and semi–supervised learning. Technical Report CU-CS-976-04, Univ. of Colorado, Boulder (2004)

Wu, N., Zhang, J.: Factor analysis based anomaly detection. In: IEEE Workshop on Information Assurance (2003)

Yegneswaran, V., Barford, P., Jha, S.: Global Intrusion Detection in the DOMINO Overlay System. In: Network and Distributed Security Symposium (2004)

Zhong, S., Khoshgoftaar, T.M., Seliya, N.: Clustering-based Network Intrusion Detection. International Journal of Reliability, Quality and Safety Engineering 14 (2007)

# Intrusion Detections in Collaborative Organizations by Preserving Privacy

Nischal Verma, François Trousset, Pascal Poncelet, and Florent Masseglia

**Abstract.** To overcome the problem of attacks on networks, new Intrusion Detection System (IDS) approaches have been proposed in recent years. They consist in identifying signatures of known attacks to compare them to each request and determine whether it is an attack or not. However, these methods are set to default when the attack is unknown from the database of signatures. Usually this problem is solved by calling human expertise to update the database of signatures. However, it is frequent that an attack has already been detected by another organization and it would be useful to be able to benefit from this knowledge to enrich the database of signatures. Unfortunately this information is not so easy to obtain. In fact organizations do not necessarily want to spread the information that they have already faced this type of attack. In this paper we propose a new approach to intrusion detection in a collaborative environment but by preserving the privacy of the collaborative organizations. Our approach works for any signature that may be written as a regular expression insuring that no information is disclosed on the content of the sites.

**Keywords:** Privacy, Intrusion Detection, Collaborative Approach.

Nischal Verma
Indian Institute of Technology Guwahati, Assam, India
e-mail: nischaliit@gmail.com

Fran ccois Trousset
LGI2P - Ecole des Mines d'Alès, Parc Scientifique G. Besse, 30035 Nîmes, France
e-mail: francois.trousset@mines-ales.fr

Pascal Poncelet
LIRMM UMR CNRS 5506, 161 Rue Ada, 34392 Montpellier Cedex 5, France
e-mail: poncelet@lirmm.fr

Florent Masseglia
INRIA Sophia Antipolis, route des Lucioles - BP 93, 06902 Sophia Antipolis, France
e-mail: florent.masseglia@sophia.inria.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 235–247.
springerlink.com 

## 1 Introduction

The fast growing computational Grid environments has increased risk of attack and intrusion. Thus misuse detection has become a real concern for companies and organizations. Whereas earlier attacks focused on Web servers which were often misconfigured or poorly maintained, the most recent ones take advantage of Security service and Web application weaknesses which become more vulnerable (Heady *et al.*, 1990; Graham, 2001; Escamilla, 1998). To overcome this problem, new approaches called Intrusion Detection Systems (IDS) have been developed. Installed on networks, they aim to analyze traffic requests and detect malicious behavior (eg Prelude-IDS, Snort). They can be classified into two broad categories (*e.g.* McHugh *et al.* 2000; Proctor 2001): the *Anomaly Detection Systems* which attempt to detect attacks and the *Abuse Detection Systems* which detects unknown comportment so called *abuse* from a specification of allowed ones. Within this paper, we particulary focus on anomaly detection. Their principle mostly consist of matching new requests which signatures of attacks represented as regular expressions. For example, an attack which seeks to recover the password file of a system (*e.g. abc/../de/../../../fg/../etc/passwd*) may be detected by matching with the following regular expression *(/[ˆ ./]*/..)*/etc/passwd*. These signatures are often obtained by using machine learning techniques or from specialized sites (*e.g.* OSVDB Database 2008).

Even if these systems are widely used today, the essential problem is that they do not know how to manage attacks outside their own signature database. When a request is not recognized by the IDS, an alarm is triggered to require external valuation.

Recently approaches called Collaborative Intrusion Detection Systems (CIDS) (*e.g.* Cuppens and Miege 2005; Zhou *et al.* 2007; Janakiraman *et al.* 2003; Locasto *et al.* 2005; Zhang and Parashar 2006) have been proposed. In comparison with isolated IDS, CIDS significantly improve time and efficiency of misuse detections by sharing information on attacks between distributed IDS from one or more organizations. The main principle of these approaches is to exchange information using peer to peer links. However the exchanged information are mostly limited to IP addresses of requests (*e.g.* Cuppens and Miege 2005; Janakiraman *et al.* 2003; Locasto *et al.* 2005) and consider that data can be freely exchanged among the peers. The last constraint is very strong: companies, for reasons of confidentiality, do not want to spread out that they were attacked and therefore are unwilling to give any information on it. In this article we propose a secure collaborative detection approach, called SREXM (*Secure Regular Expression Mapping*), which ensures that private data will not be disclosed. Via our approach, regular expressions from the various collaborative sites can be matched without disclosing any information from the local IDS to the outside. Collaborative sites are free to work with signatures of attacks or non-attacks and may give information on the type of intrusion detected. Thus, when new request is checked, the response will be one of: it is an attack (with its type if available), it is a non-attack, or undefined (if none of the IDS data leads to a positive or negative conclusion). To our knowledge, very few studies are concerned

with this topic of security in such collaborative environment. The only works (Wang *et al.*, 2005; Locasto *et al.*, 2005) consider both collaborative and security aspects. In its context, security mainly concerns information on IP addresses and ports. It uses Bloom's filters to manage data exchanges. Our problem is different in that, we want to exchange data, *i.e.* more complex than IP addresses and ports. In fact we wants to exchange and parse regular expressions on the full request.

The article is organized as follows. In section 2, we present the problem. An overview of our approach is given in section 3. The various algorithms are described in section 4. Finally section 5 concludes and presents various perspectives.

## 2 Problem Statement

$DB$ is a database such as $DB = DB_1 \bigcup DB_2 \ldots \bigcup DB_D$. Each database $DB_i$ is equivalent to a tuple $< id, S_{exp} >$ where $id$ is the identifier of the database and $S_{exp}$ is a set of regular expressions. Each regular expression $exp_i \in S_{exp}$ is expressed as a deterministic automaton (*e.g.* Hopcroft *et al.* 2000) by the tuple $a_{exp_i} =< State, Trans, Init, Final >$. In this tuple $a_{exp_i}$, $State$ is the set of states of the automaton, $Init$ is the initial state, $Final$ is the set of final states and $Trans$ is the set of transitions. Each transition is a quadruplet $(S_{Initial}, Condition, S_{Final}, Length)$ meaning that if the automaton is in state $S_{Initial}$ and that $Condition$ is checked then automaton current state changes to $S_{Final}$ and move the current position in the filtered string of the amount given by $Length$. In our approach, we also associate a value to each final state. This value is used to specify whether or not it is an attack (boolean 0 or 1), but may also provide the type of the attack (integer).

**Example 1.** *Consider the following regular expression: (/[:/]*/..)*/etc/passwd. Its associated automaton is described in Figure 1. The left table is the matrix of transitions where Conditions are indexes in the second table which contains the effective patterns to be matched with the request string. For example, to move from state* $S_6$ *to final state F, we have to check that the request string at current position contains the word "passwd".*

**Definition 1.** Given a database $DB = DB_1 \bigcup DB_2 \ldots \bigcup DB_D$ and a request string $R$, the securized approach in such a collaborative environment consist in finding a regular expression $exp$ from $DB$ such that $matching(exp, R) = TRUE$ while ensuring that none of the databases $DB_i$ provide any information from its content to anyone.

## 3 The SREXM Approach

This section will provide an overview of the secure architecture SREXM (*Secure Regular Expression Mapping*). It is to answer the problem of privacy preserving in a collaborative environment. Inspired by the work of Kantarcioglu and Vaidya (2002), this architecture offers the advantage of achieving the various operations while ensuring that neither party may have access to private data contained in the initial databases. In addition to the client site $S$ which is responsible to provide the

| | | | |
|---|---|---|---|
| $I$ | $cond_1$ | $S_1$ | 1 |
| $S_1$ | $cond_2$ | $I$ | 2 |
| $S_1$ | $cond_3$ | $S_2$ | 1 |
| $S_1$ | $cond_4$ | $S_3$ | 1 |
| $S_2$ | $cond_5$ | $S_2$ | 1 |
| $S_2$ | $cond_6$ | $I$ | 3 |
| $S_3$ | $cond_7$ | $S_2$ | 1 |
| $S_3$ | $cond_8$ | $S_4$ | 1 |
| $S_4$ | $cond_9$ | $S_2$ | 1 |
| $S_4$ | $cond_{10}$ | $S_5$ | 1 |
| $S_5$ | $cond_{11}$ | $S_2$ | 1 |
| $S_5$ | $cond_{12}$ | $S_6$ | 1 |
| $S_6$ | $cond_{13}$ | $I$ | 2 |
| $S_6$ | $cond_{14}$ | $F$ | 6 |

| | |
|---|---|
| $cond_1$ | / |
| $cond_2$ | .. |
| $cond_3$ | [^./e] |
| $cond_4$ | e |
| $cond_5$ | [^./] |
| $cond_6$ | /.. |
| $cond_7$ | [^./t] |
| $cond_8$ | t |
| $cond_9$ | [^./c] |
| $cond_{10}$ | t |
| $cond_{11}$ | [^./] |
| $cond_{12}$ | / |
| $cond_{13}$ | .. |
| $cond_{14}$ | passwd |

**Fig. 1** Automaton associated to the Regular Expression *exp*

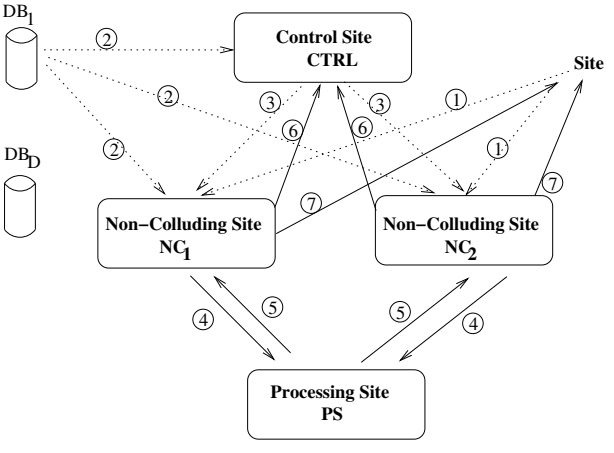



**Fig. 2** General Architecture of SREXM

request to be tested, the architecture requires four non-collaborative and semi honest sites (Goldreich, 2000): they follow the protocol correctly, but are free to use the information they have collected during the execution of the protocol. These independent sites collect, store and evaluate information in a secure way. The different functions provided by these sites are:

- **The Control Site** *CTRL***:** *CTRL* is used to rule the various operations needed to match the regular expression. To do this, it interacts with the two non colluding sites $NC_1$ and $NC_2$.

- **Non Colluding Sites $NC_1$ and $NC_2$:** These two symmetric sites collect garbled data from all databases as well as the garbled request to be tested from $S$. Under the control of $CTRL$ and by interaction with $PS$, they perform several secure operations in order to insure that none of them will be able to infer any of the intermediate results or the final result which is returned to site $S$.
- **The Processing Site $PS$:** This site is used both by $NC_1$ and $NC_2$ to process, the various operations needed, in a secure way. Like $NC_1$ and $NC_2$, $PS$ also cannot deduce any pertinent value of intermediate or final result from the data it processes.

The exchange of data between the different sites is done by using the secure method $SEND^S(\overset{+}{V}|\overset{-}{V})'$ which sends the vector of bits $V = \overset{+}{V} \oplus \overset{-}{V}$ to $NC_1$ and $NC_2$. It is defined in order to send $\overset{+}{V}$ to $NC_1$ and $\overset{-}{V}$ to $NC_2$ (or vice versa). A random vector $R$ is used for secure transmission such that $\overset{+}{V} = R$ and $\overset{-}{V} = V \oplus R$. This method is used in particular to send the data from the databases $DB_i$ and to send the request from site $S$. Thus, the process described in figure 2 starts in the following way. First, the site $S$ sends its request to $NC_1$ and $NC_2$ using the $SEND^S$ method (See arrow number 1 in figure 2). More precisely, the request $R$ is taken in its boolean form: a vector of bits. A random vector of bits $A_R$ is then generated with the same size as the request $R$ to compute the new vector $Z_R = A_R \bigoplus R$. $Z_R$ is sent to $NC_1$ and $A_R$ to $NC_2$ (or vice versa). Each database $DB_i$ decompose the transition matrix in three tables: the first contains the transitions of the automaton, the second the conditions and the third the lengths of the shifts. To encode the transition matrix, the indexes of these tables are randomly mixed. The databases first send the table of transition to $CTRL$, then, using $SEND^S$, the associated tables of conditions and lengths are sent to $NC_1$ and $NC_2$ in the same order of indexes as the one used when sent to $CTRL$ (See arrow number 2). From this point, the computation of the request is done under the control of $CTRL$. Via the $NCOMPARE^S$, it will ask $NC_1$ and $NC_2$ to test the condition of index $i$ from the table of conditions (See arrow number 3). At this point, $NC_1$ has part of the request to be tested $\overset{+}{R}$, part of the condition $\overset{+}{ST}_{R_i}$ and the current position $pos$ in the request $R$. In the same way, $NC_2$ has $\overset{-}{R}$, $\overset{-}{ST}_{R_i}$ and the position $pos$ in the request. Then $NC_1$ et $NC_2$ just have to extract the substring of the request starting at position $pos$ and of same length the string to compare ($\overset{+}{ST}_{R_i}$ or $\overset{-}{ST}_{R_i}$). The next step consist in the comparison of the two string in a secure way. This is performed by sending requested data to $PS$ using the $NCMP^S$ protocol (See arrow number 4). Under completion, the result of the comparison is divided in two parts, one is owned by $NC_1$ and the other by $NC_2$ such that none are able to infer its real value. Both parts are then securely returned to $CTRL$ (see arrows 5 and 6) which uses the result to change the state of the automaton. The process is repeated under control of $CTRL$ unless the automaton is ended (it moves to a final state of the automaton or the request does not match). The action of maintaining the position $pos$ in the request is done by $CTRL$ through the secure operation $INCR^S$ whose aim is to shift the position according to the displacement length associated with the transition. This is done by sending the index in the table of lengthes to $NC_1$ and $NC_2$ that will update

the value of *pos*. When the automaton reaches a final state, *CTRL* or matching of the request fails, *CTRL* aggregates the results (attack, non-attack, unknown) using the secure method $AGGREGATE^S$. The aggregated result is split between $NC_1$ and $NC_2$ and kept while data has to be performed on the same request. At the end of the process, the final aggregated result is sent to *S*.

## 4 The Secure Algorithms

In this section, we present the various algorithms used in SREXM approach. In order to simplify writing, we consider the following notations: Let $(\overset{+}{x}|\overset{-}{x}) \leftarrow h^S(\overset{+}{y_1}...\overset{+}{y_n}|\overset{-}{y_1}...\overset{-}{y_n})$ be a tripartite computation of any function $h^S$ between $NC_1$, $NC_2$ and *PS* where $NC_1$ owned some of the entries $\overset{+}{y_1}...\overset{+}{y_n}$ and gets part of the result $\overset{+}{x}$ and similarly $NC_2$ owned some of the entries $\overset{-}{y_1}...\overset{-}{y_n}$ and gets part of the result $\overset{-}{x}$. The final result is obtained by applying the binary operator XOR ($\bigoplus$) between $\overset{+}{x}$ and $\overset{-}{x}$. However, this does not mean that $NC_1$ sends exactly $\overset{+}{y_1}...\overset{+}{y_n}$ to *PS* an receives the result $\overset{+}{x}$ from *PS*. In fact, $NC_1$ garbles its inputs $\overset{+}{y_1}...\overset{+}{y_n}$ by adding random noise and gets $\overset{+}{y_1'}...\overset{+}{y_n'}$ which are securely sent to *PS*. Similarly, $NC_2$ sends its garbled inputs to *PS*. At the end of the process, both sites receive a part of garbled result from *PS* (respectively $\overset{+}{x'}$ and $\overset{-}{x'}$). This intermediate result may now be used as input of further computation. We will also use the following simplifications:

1. $g^S(\overset{+}{x},\overset{+}{y}|\overset{-}{x},\overset{-}{y}) \Leftrightarrow g^S(\overset{+}{x}|\overset{-}{x};\overset{+}{y}|\overset{-}{y})$
2. Si $h^S()$ is a 2 argument function then $h^S(\overset{+}{x_1},\cdots,\overset{+}{x_n}|\overset{-}{x_1},\cdots,\overset{-}{x_n})$ will correspond to $h^S(h^S(\cdots h^S(h^S(\overset{+}{x_1},\overset{+}{x_2}|\overset{-}{x_1},\overset{-}{x_2});\overset{+}{x_3}|\overset{-}{x_3})\cdots);\overset{+}{x_n}|\overset{-}{x_n})$.

### 4.1 The Algorithm $NCOMPARE^S$

The evaluation of the condition $NCOMPARE^S(Str|Str)$ (See Algorithm 1) associated with a transition is controlled by the controller *CTRL*. It sends the index *i* of a string in the table of conditions to $NC_1$ and $NC_2$. Thus $NC_1$ only hold the part $\overset{+}{STR_i}$ and $NC_2$ the other part $\overset{-}{STR_i}$ such that the real string is $STR_i = \overset{+}{STR_i} \oplus \overset{-}{STR_1}$. Each $NC_1$ and $NC_2$ sites also holds its part of the request $(\overset{+}{R}|\overset{-}{R})$ and the current position in the request (*pos*). After extracting the substring of the request *R* starting at position *pos* and of same length with $STR_i$, the comparison is performed via $NCMP^S$. The operator $NCMP^S(\overset{+}{S_1},\overset{+}{S_2}|\overset{-}{S_1},\overset{-}{S_2}) \rightarrow (\overset{+}{b}|\overset{-}{b})$ (see section 4.4) compares two sequence of bits of same length $S_1 = \overset{+}{S_1} \oplus \overset{-}{S_1}$ and $S_2 = \overset{+}{S_2} \oplus \overset{-}{S_2}$ and returns a boolean value $b = \overset{+}{b} \oplus \overset{-}{b}$ such that *b* is false if $S_1$ and $S_2$ are identical and otherwise true. The final result is returned to *CTRL*.

**Algorithm 1.** Algorithm $NCOMPARE^S$

**Data**: $(i|i)$ The index of the condition to be tested is sent to $NC_1$ and $NC_2$ by $CTRL$.

**Result**: $(\overset{+}{b}|\overset{-}{b})$ two booleans such that $b = \overset{+}{b} \oplus \overset{-}{b}$ is false when $STR_i$ matches the substring starting a current position of the request. Otherwise, it is true.

1. $NC_1$ ccomputes $Len_1 = length(\overset{+}{STR_i})$; $NC_2$ ccomputes $Len2 = length(\overset{-}{STR_i})$.
   // By definition $Len_1 = Len_2$
2. If ( $(pos + Len_1 > length(\overset{+}{R}))||(pos + Len_2 > length(\overset{-}{R}))$ )
   then return $(\overset{+}{b}|\overset{-}{b}) = (1|0)$
3. $NC_1$ computes $\overset{+}{S} = \overset{+}{R}_{pos} \cdots \overset{+}{R}_{pos+Len_1-1}$
4. $NC_2$ computes $\overset{-}{S} = \overset{-}{R}_{pos} \cdots \overset{-}{R}_{pos+Len_2-1}$
5. compute $(\overset{+}{b}|\overset{-}{b}) = NCMP^S(\overset{+}{S}|\overset{-}{S})$ using $PS$, $NC_1$ and $NC_2$.

*Complexity:* The complexity of $NCOMPARE^S$ is same as the one of $NCMP^S$ (see section 4.4).

$COMPARE^S$ does not allow $NC_1$ or $NC_2$ to get knowledge on the result of the comparison. They can only deduce the length of left part of the request which have been successfully matched by the automaton (in fact the value of *pos*). But even if they could obtain the list of strings that has been matched successfully, as they only hold random data in the table of condition, they can only infer that a random sequence of length *pos* has matched the beginning of the request. However, they can not deduce neither whether the filtering was successful or not nor the value associated with the final state in case of successful filtering. At the level of $CTRL$ no information on the length of the filtered part of the query can be inferred. Indeed $CTRL$ has no access to the real data (request, condition strings, lengthes). It only knows indexes. The only information it can obtain is the path followed by the automaton to provide an answer.

## 4.2 The Algorithm $INCR^S$

The request $R$ to be tested is split between $NC_1$ and $NC_2$, in a secure way. The starting position *pos* is known by both $NC_1$ and $NC_2$. Any modification to this position is controlled by $CTRL$ via the $INCR^S(len|len)$ operation. When the automaton is sent to $CTRL$ and data to $NC_1$ and $NC_2$, these two also receives a table with indexes aleatory sorted and which contains the lengths of movements. The goal of this sort is to avoid any direct correspondence between the index of conditions and lengths. The $INCR^S$ method just sends the index to be used to $NC_1$ and $NC_2$ and each one updates the position *pos* according to the value found in the table.

When an increment is triggered by $CTRL$, there is no way for $NC_1$ or $NC_2$ to know which condition had activated it. In fact $CTRL$ may execute unnecessary

computation. On *CTRL* side, neither the information of the length may be available nor inferred as it knows only indexes.

### 4.3 The Algorithm $AGGREGATE^S$

Aggregation of results simply consists to securely retain the first valid result obtained by *CTRL*, *i.e.* when an automaton has matched the request and lead in a final state. The objective of $AGGREGATE^S$ is to conceal from $NC_1$ and $NC_2$, the fact that an automaton has filtered the request (and associated value $W_f$) or not. This is done by setting a state bit to 1 if the automaton has filtered the request and 0 otherwise. Depending on the value of this bit, the information stored in the accumulator between $NC_1$ and $NC_2$ will be either the value of the final state $W_f$ or a random vector. The implementation of $AGGREGATE^S$ require the secure operators $\bigvee^S(\overset{+}{s_1},\overset{+}{s_2}|\overset{-}{s_1},\overset{-}{s_2}) \rightarrow (\overset{+}{v}|\overset{-}{v})$ and $\bigwedge^S(\overset{+}{s_1},\overset{+}{s_2}|\overset{-}{s_1},\overset{-}{s_2}) \rightarrow (\overset{+}{v}|\overset{-}{v})$ which implements respectively a secure computation of bitwise operators OR and AND on vectors of bits of same length ($S_1$ and $S_2$) and returns the sequence $V$. At the end of the process SREXM, $NC_1$ and $NC_2$ both sends the value of their part of the accumulator to the client site $S$. Finally $S$ has just need to take XOR of the received values to get the result.

For each regular expression (automaton), the values $V_f$ associated with final states are encoded with random numbers $R_1$ and $R_2$ by computing $W_f = V_f \oplus R_1 \oplus R_2$. *CTRL* knows $W_f$, $NC_1$ knows $R_1$ and $NC_2$ knows $R_2$. We consider that the length of $W_F$ is identical in all databases.

**Algorithm 2.** Algorithm $AGGREGATE^S$

**Data**: $Y = \overset{+}{Y} \oplus \overset{-}{Y}$ of length $n+1$ whose first bit is the bit of state set by *CTRL*.
// $\overset{+}{A}$ and $\overset{-}{A}$ are the aggregated values respectively kept by $NC_1$ and $NC_2$.
// $n+1$ is the length of A.
1. $NC_1$ computes $\overset{+}{Z} = \overset{+}{Y} \oplus 0R_1$; $NC_2$computes $\overset{-}{Z} = \overset{-}{Y} \oplus 0R_2$;
2. $\forall k \in 1..n$ $NC_1$,$NC_2$ and $PS$ compute
   $(\overset{+}{B_k}|\overset{-}{B_k}) = \bigwedge^S(\ \bigvee^S(\overset{+}{Z_0},\overset{+}{A_k}|\overset{-}{Z_0},\overset{-}{A_k})\ ;\ \bigvee^S(\neg\overset{+}{Z_0},\overset{+}{Z_k}|\overset{-}{Z_0},\overset{-}{Z_k})\ )$
3. $NC_1$, $NC_2$ and $PS$ compute $(\overset{+}{B_0}|\overset{-}{B_0}) = \bigvee^S(\overset{+}{Z_0},\overset{+}{A_0}|\overset{-}{Z_0},\overset{-}{A_0})$
4. $NC_1$ and $NC_2$ respectively computes $\overset{+}{A} = \overset{+}{B}$ and $\overset{-}{A} = \overset{-}{B}$.

**Property 1.** *$AGGREGATE^S$ prohibits $NC_1$ and $NC_2$ to access the value stored in the accumulator. They even do not know if the value stored in the accumulator has changed or not.*

*Proof*: The data $(\overset{+}{Y}|\overset{-}{Y})$ held by $NC_1$ and $NC_2$ are randomized by *CTRL*. It is therefore impossible to know the value of $Y$ and obviously that of $Y_0$ (*i.e.* the state bit indication whether the automaton has reached a final state or not). As operators $\bigvee^S$ and $\bigwedge^S$ returns values garbled with random noise, from the point of view of $NC_1$ (respectively $NC_2$) the received value $\overset{+}{B}$ (respectively $\overset{-}{B}$) is pure random and thus

independent from the values of $\overset{+}{Y}$ and $\overset{+}{A}$ (respectively $\bar{Y}$ and $\bar{A}$). In particular, although $NC_1$ and $NC_2$ know the initial value of $A$ (0 at the beginning of the process), it is impossible for them to deduce whether this value has been changed or not, once $AGGREGATE^S$ has been used.

*Complexity:* The methods $\bigvee^S$ and $\bigwedge^S$ are used $2n+1$ and $n$ times respectively on one bit. By reusing the complexity of operators $\bigvee^S$ and $\bigwedge^S$ (see section 4.4), $NC_1$ and $NC_2$ therefore perform $34n+12$ binary operations, generate $6n+2$ aleatory bits, send $12n+4$ bits and receive $10n+4$ bits (including parameters). *PS* performs $12n+4$ binary operations, generates $3n+1$ aleatory bits, receives $12n+4$ bits ($6n+2$ from $NC_1$ and $NC_2$ each) and sends $6n+2$ bits ($3n+1$ to $NC_1$ and $NC_2$ each). Obviously this has to be compared with the length of inputs ($n+1$ bits).

*Remarks:* The two mechanisms *bufferization of data sent by the databases* and *aggregation of results* fulfil databases anonymization. Indeed, even if the client can identify which databases are sending data to SREXM, it can not infer the one which gave the final result. The aggregated value may be returned to the client immediately after a valid match. However, in this case, $NC_1$ and $NC_2$ are able to infer the identity of the database who gave the answer. To improve the anonymization, it is necessary to wait, for example until each data from all databases have been processed. Meanwhile this approach is secure, but it is unfortunately not effective because too expensive in term of time. To minimize time cost, we can return intermediate values to the clients each time $n$ results are aggregated which lower the time overcost to $n/2$. In fact both anonymization mechanisms have different costs: the buffering essentially introduces space cost while aggregation introduces computing time cost. It is of course possible to mix the two mechanism and adapt parameters to adjust anonymization process according to the needs and bearable costs.

**Algorithm 3.** The Algorithm $\bigwedge^S$

**Data**: $(\overset{+}{X},\overset{+}{Y}|\bar{X},\bar{Y})$ vector of bit/s are such that $\overset{+}{X}$ and $\overset{+}{Y}$ are in $NC_1$, and $\bar{X}$ and $\bar{Y}$ are in $NC_2$
**Result**: $(A^R|B^R)$ is such that $A^R \oplus B^R = (\overset{+}{X} \oplus \bar{X}) \bigwedge (\overset{+}{Y} \oplus \bar{Y})$

1. $NC_1$ and $NC_2$ mutually generate and exchange four random vectors of bits $R_A$, $R'_A$, $R_B$ and $R'_B$ such that: $\overset{+}{X}' = \overset{+}{X} \bigoplus R_A$, $\overset{+}{Y}' = \overset{+}{Y} \bigoplus R'_A$, $\bar{X}' = \bar{X} \bigoplus R_B$ and $\bar{Y}' = \bar{Y} \bigoplus R'_B$.
2. $NC_1$ sends $\overset{+}{X'}$ and $\overset{+}{Y'}$ to *PS*.
3. $NC_2$ sends $\bar{X'}$ and $\bar{Y'}$ to *PS*.
4. *PS* computes $\overset{+}{C} = \overset{+}{X'} \bigwedge \bar{Y'}$ and $\bar{C} = \overset{+}{Y'} \bigwedge \bar{X'}$ and generates a random vector of bit/s $R_{PS}$.
5. *PS* sends $A'_{PS} = \overset{+}{C} \bigoplus R_{PS}$ to $NC_1$ and $B'_{PS} = \bar{C} \bigoplus R_{PS}$ to $NC_2$.
6. $NC_1$ computes $A^R = A'_{PS} \bigoplus (\overset{+}{X} \bigwedge R'_B) \bigoplus (\overset{+}{Y} \bigwedge R_B) \bigoplus (\overset{+}{X} \bigwedge \overset{+}{Y}) \bigoplus (R_B \bigwedge R'_A)$
7. $NC_2$ computes $B^R = B'_{PS} \bigoplus (\bar{X} \bigwedge R'_A) \bigoplus (\bar{Y} \bigwedge R_A) \bigoplus (\bar{X} \bigwedge \bar{Y}) \bigoplus (R_A \bigwedge R'_B)$.

### 4.4 The Algorithms $NCMP^S$, $\bigwedge^S$ and $\bigvee^S$

In this section, we define three algorithms used to implement the secure operator for string comparison, the basic principle of these algorithms is to add uniform random noise to the data which could be deleted from the final result.

The $\bigwedge^S$ protocol begins with $NC_1$ and $NC_2$ who modify their data by doing XOR them with random values (see step 1 in algorithm ). $NC_1$ and $NC_2$ share these random values (also see step 1). Garbled data are then send to $PS$ (step 2 and 3) which is now able to compute $\bigwedge$ in a secure way (step 4). In fact, $PS$ gets only garbled inputs indistinguishable from random and unrelated to each others and thus calculates random values from its point of view. To avoid $NC_1$ and $NC_2$ from inferring the final result, it does XOR with random noise to the values it calculates before sending them back to $NC_1$ and $NC_2$ (step 5). Now $NC_1$ and $NC_2$ may both obtain their part of the final result by removing the random noise they added on step 1 (see step 6 and 7). The final result is obtained bay computing $A^R \bigoplus B^R = A'_{PS} \bigoplus (\overset{+}{X} \bigwedge R'_B) \bigoplus (\overset{+}{Y} \bigwedge R_B) \bigoplus (\overset{+}{X} \bigwedge \overset{+}{Y}) \bigoplus (R_B \bigwedge R'_A) \bigoplus B'_{PS} \bigoplus (\bar{X} \bigwedge R'_A) \bigoplus (\bar{Y} \bigwedge R_A) \bigoplus (\bar{X} \bigwedge \bar{Y}) \bigoplus (R_A \bigwedge R'_B)$ où $A'_{PS} \bigoplus B'_{PS} = (\overset{+}{X} \bigwedge R'_B) \bigoplus (\overset{+}{Y} \bigwedge R_B) \bigoplus (\bar{X} \bigwedge R'_A) \bigoplus (\bar{Y} \bigwedge R_A) \bigoplus (\overset{+}{X} \bigwedge \overset{+}{Y}) \bigoplus (\bar{X} \bigwedge \bar{Y}) \bigoplus (R_A \bigwedge R'_B) \bigoplus (R_B \bigwedge R'_A) \bigoplus R_{PS} \bigoplus R_{PS}$.

Using the property of the XOR operator: $R \bigoplus R = 0$, we get the desired result: $A^R \bigoplus B^R = \overset{+}{X} \bigwedge \overset{+}{Y} \bigoplus \overset{+}{X} \bigwedge \bar{Y} \ \bar{X} \bigwedge \overset{+}{Y} \bigoplus \bar{X} \bigwedge \bar{Y}$. Which is a re-written form of $(\overset{+}{X} \bigoplus \bar{X}) \bigwedge (\overset{+}{Y} \bigoplus \bar{Y})$. However, this operation is never performed by the non collaborative sites and the final result is kept shared between $NC_1$ and $NC_2$.

The $\bigvee^S$ protocol is identical to the $\bigwedge^S$ protocol except for the last two steps (steps 6 and 7) performed by $NC_1$ and $NC_2$. Thus we get the final result: $A^R \bigoplus B^R = \overset{+}{C'} \bigoplus (\overset{+}{X} \bigwedge R'_B) \bigoplus (\overset{+}{Y} \bigwedge R_B) \bigoplus \overset{+}{X} \bigoplus \overset{+}{Y} \bigoplus (\overset{+}{X} \bigwedge \overset{+}{Y}) \bigoplus (R_B \bigwedge R'_A) \bigoplus \bar{C'} \bigoplus (\bar{X} \bigwedge R'_A) \bigoplus (\bar{Y} \bigwedge R_A) \bigoplus \bar{X} \bigoplus \bar{Y} \bigoplus (\bar{X} \bigwedge \bar{Y}) \bigoplus (R_A \bigwedge R'_B)$. This reduce to the desired result: $A^R \bigoplus B^R = \overset{+}{X} \bigoplus \overset{+}{Y} \bigoplus (\overset{+}{X} \bigwedge \overset{+}{Y}) \bigoplus (\overset{+}{X} \bigwedge \bar{Y}) \bigoplus \bar{X} \bigoplus \bar{Y} \bigoplus (\bar{X} \bigwedge \overset{+}{Y}) \bigoplus (\bar{X} \bigwedge \bar{Y})$.

Which is a re-written form of $(\overset{+}{X} \oplus \bar{X}) \bigvee (\overset{+}{Y} \oplus \bar{Y})$.

---

**Algorithm 4.** The algorithm $\bigvee^S$

---

**Data**: $(\overset{+}{X}, \overset{+}{Y} | \bar{X}, \bar{Y})$ vectors of bits such that $\overset{+}{X}$ et $\overset{+}{Y}$ belongs to $NC_1$, $\bar{X}$ and $\bar{Y}$ belongs to $NC_2$.

**Result**: $(A^R | B^R)$ is such that $A^R \bigoplus B^R = (\overset{+}{X} \bigoplus \bar{X}) \bigvee (\overset{+}{Y} \bigoplus \bar{Y})$.

1..5. These steps are same as initial 5 steps of $\bigwedge^S$ function.

6. $NC_1$ computes $A^R = A'_{PS} \bigoplus (\overset{+}{X} \bigwedge R'_B) \bigoplus (\overset{+}{Y} \bigwedge R_B) \bigoplus \overset{+}{X} \bigoplus \overset{+}{Y} \bigoplus (\overset{+}{X} \bigwedge \overset{+}{Y}) \bigoplus (R_B \bigwedge R'_A)$.
7. $NC_2$ computes $B^R = B'_{PS} \bigoplus (\bar{X} \bigwedge R'_A) \bigoplus (\bar{Y} \bigwedge R_A) \bigoplus \bar{X} \bigoplus \bar{Y} \bigoplus (\bar{X} \bigwedge \bar{Y}) \bigoplus (R_A \bigwedge R'_B)$.

---

**Property 2.** *$\bigwedge^S$ and $\bigvee^S$ forbid $NC_1$ to gain any information of private data of $NC_2$ (and vice versa). Moreover, the PS learns none of their private inputs.*

*Proof*: From the protocol, $B'_{PS}$ is the only value that $NC_2$ can learn from the private data of $NC_1$. Due to the noise, $R_{PS}$, added by $PS$, $NC_2$ is still not able to deduce the values of $\overset{+}{x}$ or $\overset{+}{y}$. As the roles of $NC_1$ and $NC_2$ are interchangeable, the same argument holds for $NC_1$, not able to learn the private inputs $\bar{x}$ or $\bar{y}$ of $NC_2$. However, one key security aspect of not leaking any information to $PS$ is achieved by randomizing the inputs before transmitting them to the Processing Site. Due to the randomization performed during the initial step, it just infers a stream of uniformly distributed values, and cannot distinguish between a genuine and a random value.

*Complexity: Length of bit vector is 1:* For the operator $\bigwedge^S$, $NC_1$ and $NC_2$ each performs 10 binary operations ($6\bigoplus$ and $4\wedge$). $\bigvee^S$ does two more $\bigoplus$ that means 12 binary operations. For both operators $NC_1$ and $NC_2$ generate 2 random bits, exchange $2\times 2$ random bits and send $2\times 1$ bits to $PS$. $PS$ generates 1 random bit and performs 4 binary operation ($2\bigoplus$ and $2\bigwedge$) and returns 2 bits to $NC_1$ and $NC_2$ each.

The $NCMP^S()$ method compares two vectors of bits by using the secure $\bigvee^S$ method. The result of $NCMP^S()$ consists of 2 bits. One is sent to $NC_1$ and the other is sent to $NC_2$. XOR of these two bits is 0 if the vectors are similar, otherwise 1.

**Algorithm 5.** The Algorithm $NCMP^S$

**Data**: Half part of $V$ and $W$ is owned by $NC_1$ and the other part is owned by $NC_2$

**Result**: $(\overset{+}{R}|\bar{R})$ is such that $\overset{+}{R}\oplus\bar{R}=0$ if $V=W$ else 1

1. $NC_1$ computes $X\leftarrow\overset{+}{V}\oplus\overset{+}{W}$ where $X=(X_1,X_2,\cdots,X_l)$ and $l$ is the length of vector $V$ and $W$.
2. $NC_2$ computes $Y\leftarrow\bar{V}\oplus\bar{W}$ where $Y=(Y_1,Y_2,\cdots,Y_l)$.
3. $(\overset{+}{R}|\bar{R})\leftarrow OR^S(X_1,X_2,\cdots,X_l|Y_1,Y_2,\cdots,Y_l)$

*Complexity: Length of bit vector is l: $CMP^S$* executes $l\oplus$ operations and $l-1$ $\bigvee^S$. Thus $NC_1$ and $NC_2$ compute $13l-12$ binary operations, generate $2l-2$ aleatory bits, receive $4l-3$ bits (including inputs) and send $5l-4$ bits (including the result). On $PS$ side, $PS$ computes $4l-4$ binary operations, generates $l-1$ aleatory bits, receives $4l-4$ bits and sends $2l-2$ bits.

**Property 3.** *$NC_1$ and $NC_2$ gain no information of the real values which are compared and of the result of the comparison.*

*Proof*: The input data sent to $NC_1$ and $NC_2$ are garbled with random values. Thus they cannot distinguish them from random values. In the same way, all values returned by $\bigvee^S$ are also garbled with unrelated random bits. Thus $NC_1$ and $NC_2$ only gets random values and then cannot infer the actual values of the inputs or results. If $PS$ keeps history of intermediate results, it might deduce a part of the aleatory bits

that were used to encode its results sent to $NC_1$ and $NC2$. However, this gives no information of actual data.

## 5 Conclusion

In this paper, we proposed a new approach of secured intrusion detection in a collaborative environment. Via our approach an application can use knowledge from foreign databases to identify whether a request corresponds to an attack or not. We have demonstrated that the proposed architecture ensured that it is impossible to identify which database has given the answer and that none of the internal components of the architecture can infer knowledge on the databases or on the request from the data they got. Our approach may also provide the type of the attack when they are specified in the databases. Our current work concern the study of the removal of the fourth semi-honest site *CTRL* by trying to dispatch its proceedings on the automaton on the three other ones. In parallel, we try to improve the management of the automaton (*i.e.* introduce more powerful comparison operators).

## References

Cuppens, F., Miege, A.: Alert correlation in a cooperative intrusion detection framework. In: Proc. of the IEEE International Conference on Networks (ICON 2005), pp. 118–123 (2005)

The Open Source Vulnerability Database (2008), `http://osvdb.org/`

Escamilla, T.: Intrusion Detection: Network Security beyond the firewall. John Wiley and Sons, New York (1998)

Goldreich, O.: Secure multi-party computation - working draft (2000), `citeseer.ist.psu.edu/goldreich98secure.html`

Graham, R.: FAQ: Network Intryusion Detection System (2001), `http://www.robertgraham.com/pubs/network-intrusion-detection.html`

Heady, R., Luger, G., Maccabe, A., Servilla, M.: The Architecture of a Network Level Intrusion Detection System. Technical Report CS9020 (1990)

Hopcroft, J., Motwanu, R., Rotwani, Ullman, J.: Introduction to Automata Theory, Languages and Computability. Addison-Wesley, Reading (2000)

Janakiraman, R., Waldvoge, M., Zhang, Q.: Indra: a peer-to-peer approach to network intrusion detection and prevention. In: Proc. of the 12th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 226–231 (2003)

Kantarcioglu, M., Vaidya, J.: An architecture for privacy-preserving mining of client information. In: Proc. of the Workshop on Privac, pp. 27–42 (2002)

Locasto, M., Parekh, J., Keromytis, A., Stolfo, S.: Towards Collaborative Security and P2P Intrusion Detection. In: Proceedings of the 2005 IEEE Workshop on Information Assurance and Security, West Point, NY (2005)

McHugh, J., Christie, A., Allen, J.: Defending yourself: the role of intrusion detection systems. IEEE Software, 42–51 (2000)

Proctor, P.: Practical Intrusion Detection Handbook. Prentice-Hall, Englewood Cliffs (2001)

Wang, K., Cretu, G., Stolfo, S.: Anomalous Payload-based Worm Detection and Signature Generation. In: Proceedings of the 8th International Symposium on Recent Advances in Intrusion Detection (2005)

Zhang, G., Parashar, M.: Cooperative Defence Against DDoS Attacks. Journal of Research and Practice in Information Technology 38(1) (2006)

Zhou, C.V., Karunasekera, S., Leckie, C.: Evaluation of a Decentralized Architecture for Large Scale Collaborative Intrusion Detection. In: Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Network Management (IM 2007), pp. 80–89 (2007)

# Part IV
# Ontologies and Semantic

# Alignment-Based Partitioning of Large-Scale Ontologies

Fayçal Hamdi, Brigitte Safar, Chantal Reynaud, and Haïfa Zargayouna

**Abstract.** Ontology alignment is an important task for information integration systems that can make different resources, described by various and heterogeneous ontologies, interoperate. However very large ontologies have been built in some domains such as medicine or agronomy and the challenge now lays in scaling up alignment techniques that often perform complex tasks. In this paper, we propose two partitioning methods which have been designed to take the alignment objective into account in the partitioning process as soon as possible. These methods transform the two ontologies to be aligned into two sets of blocks of a limited size. Furthermore, the elements of the two ontologies that might be aligned are grouped in a minimal set of blocks and the comparison is then enacted upon these blocks. Results of experiments performed by the two methods on various pairs of ontologies are promising.

**Keywords:** Ontology Matching, Ontology Partitioning.

## 1 Introduction

The fast development of internet technology engendered a growing interest in research on sharing and integrating sources in a distributed environment. The Semantic Web (Berners-Lee *et al.*, 2001) offers possibility for software agents to exploit

Fayçal Hamdi · Brigitte Safar · Chantal Reynaud
LRI, Université Paris-Sud 11 - CNRS UMR 8623, Bât. G, INRIA Saclay - Île-de-France, 2-4 rue Jacques Monod, F-91893 Orsay, France
e-mail: firstname.lastname@lri.fr

Haïfa Zargayouna
LIPN, Université Paris 13 - CNRS UMR 7030, 99 av. J.B. Clément, 93440 Villetaneuse, France
e-mail: haifa.zargayouna@lipn.univ-paris13.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 251–269.
springerlink.com 

representations of the sources contents. Ontologies have been recognized as an essential component for knowledge sharing and the realisation of the Semantic Web vision. By defining the concepts of specific domains, they can both describe the content of the sources to be integrated and explain the vocabulary used by users in requests. However, it is very unlikely that a single ontology covering whole distributed systems can be developed. In practice, ontologies used in different systems are developed independently by different communities. Thus, if knowledge and data must be shared, it is essential to establish semantic correspondences between the ontologies of these systems. The task of alignment (search for mappings between concepts) is thus particularly important for integration systems because it allows several heterogeneous systems, which each has its own ontology, to be used jointly. This research subject has resulted in numerous works (Shvaiko and Euzenat, 2005).

The current techniques of alignment are usually based upon similarity measures between pairs of concepts, one from each ontology. These measures are mostly based on the lexical characteristics of the concept labels and/or on the structural characteristics of the ontologies (Rahm and Bernstein, 2001; Noy and Musen, 2000; Reynaud and Safar, 2007) which involve comparing the description of each concept in one ontology with the description of all concepts in the other. Theses techniques are often tested on small ontologies (a few hundred concepts). When ontologies are very large, for example in Agronomy or Medicine, ontologies include tens of thousands of concepts (AGROVOC[1] : 28 439, NALT [2] : 42 326, NCI[3] : 27 652), and the effectiveness of the automatic alignment methods decreases considerably in terms of execution time, size of memory used or accuracy of resulting mappings. A possible solution to this problem is to try to reduce the number of concepts given to the alignment tool, and for this purpose to partition both ontologies to be aligned into several blocks, so the processed blocks have a reasonable size.

We propose two methods of partitioning guided by the task of alignment. These methods are partially inspired by co-clustering techniques, which consist in exploiting, besides the information expressed by the relations between the concepts within one ontology, the information which corresponds to the inter-concept relations which can exist across both ontologies. The fact that concepts of both ontologies can have exactly the same label and can be connected by a relation of equivalence is an example of relation easy to calculate even on large ontologies, and which we will use to our benefit. Our methods will thus start by identifying, with a similarity measure strict and inexpensive to calculate, the couples of concepts from the ontologies which have identical labels, and will base itself on these concepts, called *anchors*, to make the partitions.

The rest of the paper is organized as follow. In the next section, we present the context of our work and some related works in the domain of partitioning, and then we detail more precisely the algorithm of partitioning PBM used by the alignment system FALCON (Hu *et al.*, 2006, 2008) on which we based our propositions. In Section 3 we detail our two methods of partitioning. In Section 4 we present and

[1] http://www4.fao.org/agrovoc/

[2] http://agclass.nal.usda.gov/agt/

[3] http://www.mindswap.org/2003/CancerOntology/

analyse the experimental results which demonstrate the relevance of these methods. Finally, we conclude and we give some perspectives in section 5.

## 2 Context and State of the Art

The problem which we are interested in is the scalability of the ontologies alignment methods.

### *2.1 Context*

An ontology corresponds to a description of an application domain in terms of concepts characterized by attributes and connected by relations. The ontology alignment task consists in generating in the most automatic way relations between the concepts of two ontologies. The types of these matching relations can be equivalence relations *isEq*, subsumption relations *isA* or proximity relations *isClose*. When the ontologies are very large, the efficiency of automatic alignment methods decreases considerably. The solution which we consider is to limit the size of the input sets of concepts given to the alignment tool. In order to do this we partition both ontologies to be aligned into several blocks, so only blocks of reasonable size are processed. The two sets of blocks obtained will then be aligned in pairs, each pair made from a block from each set, and the objective consists in minimizing the number of pairs to be aligned.

Our contribution is the elaboration of a partitioning algorithm adapted to the task of alignment and usable on all ontologies containing a hierarchy of labelled concepts. It only exploits the relations of subsumption between concepts and their labels. Partitioning a set $E$ consists in finding disjoined subsets $E_1, E_2, ..., E_n$, of elements semantically close i.e. connected by an important number of relations. The realisation of this objective consists in maximizing the relations within a subset and in minimizing the relations between different subsets.

The quality of the result of a partitioning will be appreciated according to the following criteria:

- The size of generated blocks: blocks must be smaller than the maximum number of elements that the alignment tool can handle.
- The number of generated blocks: this number must be as low as possible to limit the number of pairs of blocks to be aligned.
- The degree of blocks cohesiveness: a block will have a strong cohesiveness if the structural relations are strong inside the block and weak outside. This degree groups the elements which can possibly match into a minimal number of blocks and thus reduces the number of comparisons to be made.

The fact that the partitioning algorithm only uses, in a light treatment, the subsumption relationships between the concepts allows very large ontologies to be partitioned. It is thus a scalable approach.

## 2.2 State of the Art

In real application domains the ontologies are becoming increasingly large and many works as Stuckenschmidt and Klein (2004), Grau *et al.* (2005) and Hu *et al.* (2006) are interested in ontology partitioning.

Thus the work reported in Stuckenschmidt and Klein (2004) aims at decomposing ontologies into independent sub-blocks (or *islands*), in order to facilitate different operations, such as maintenance, visualization, validation or reasoning, on the ontologies. This method is not adapted to our problem because the process of blocks generation imposes a constraint on the minimal size of the generated blocks which is not appropriate for alignment. In addition, it builds many small blocks, which has a negative impact on the final step of alignment. Works presented in the Modular Ontology conference (Haase *et al.*, 2007) focus specifically on the problems of reasoning and seek to build modules centred on coherent sub-themes and self-sufficient reasoning. For example, the work of Grau *et al.* (2005) are very representative of this issue, and guarantee that all the concepts connected by links of subsumption are grouped together into a single module. For ontologies containing tens of thousands of subsumption relations (as AGROVOC and NALT) this type of constraint can lead to the creation of blocks with badly distributed sizes, unusable for alignment. However, this technique is used by the MOM system to align (theoretically) large ontologies, but the tests presented in Wang *et al.* (2006) are only applied on ontologies of less than 700 concepts.

In our knowledge, only PBM Partition-based Block Matching system, integrated into the ontology matching system FALCON (Hu *et al.*, 2006, 2008) has been created in order to align ontologies, but we will see that its method of decomposition does not take completely into account all the constraints imposed by this context, in particular the fact of working simultaneously with two ontologies.

## 2.3 The PBM Method

The PBM[4] method proposed in Hu *et al.* (2006) consists in decomposing into blocks each ontology independently, by the clustering ROCK algorithm (Guha *et al.*, 2000), and then by measuring the proximity of each block of an ontology with every block of the other ontology in order to align only the pairs of concepts belonging to the closest blocks.[5] To make the partition, while ROCK considers that the links between the concepts all have the same value, PBM introduced the concept of *weighted links* mainly based on a structural similarity between concepts.

[4] The description of the PBM algorithm we present here is based upon the implementation available at: http://iws.seu.edu.cn/projects/matching/

[5] The blocks are built as sets of concepts, and an intermediate step, used by PBM but which we are not describing here, is needed to retransform them into ontology fragments.

### 2.3.1 Weighted Links

Let $c_i$, $c_j$ be two concepts of the same ontology $O$, $c_{ij}$ their smallest common ancestor and $depthOf(c)$ the distance in number of edges between the concept $c$ and the root of $O$. PBM measures the value of the link connecting $c_i$ and $c_j$ called $Link_s(c_i, c_j)$ using the measure of Wu and Palmer (1994):

$$Link_s(c_i, c_j) = \frac{2 * depthOf(c_{ij})}{depthOf(c_i) + depthOf(c_j)}$$

To prevent high calculation cost of similarity between each pair of concept, PBM considers only the concepts which satisfy the following relation:

$$\left| depthOf(c_i) - depthOf(c_j) \right| \leq 1$$

### 2.3.2 Partitioning Algorithm

For partitioning two ontologies in blocks, PBM is based on two essential notions: the *cohesiveness* within a block and the *coupling* between two separate blocks. Cohesiveness is a measure of the weight of all links connecting concepts belonging to the same block, and coupling is a measure of the weight of all links connecting concepts of two different blocks. Theses notions are calculated with the same measure called *goodness*:

$$goodness(B_i, B_j) = \frac{\sum_{c_i \in B_i, c_j \in B_j} Link_s(c_i, c_j)}{sizeOf(B_i) \cdot sizeOf(B_j)}$$

$Cohesiveness(B_i) = goodness(B_i, B_i)$, $Coupling(B_i, B_j) = goodness(B_i, B_j)$ where $B_i \neq B_j$.

Given an ontology $O$, the algorithm takes for input the set $B$ of $n$ blocks to partition, where each block is initially reduced to a single concept of $O$, and a $k$ desired number of output blocks or a parameter $\varepsilon_1$ limiting the maximum number of concepts in each block. It first initializes the cohesiveness value of each block as well as the coupling value. For each iteration, the algorithm chooses the block which has the maximum cohesiveness value and the block which has the maximum coupling value with the first block. It replaces theses two blocks by the result of their fusion and updates coupling values of all blocks by taking this new block into account. The algorithm stops when it reaches the desired number of blocks or when all blocks have reached the size limit or there is no block whose cohesiveness is larger than zero.

### 2.3.3 Identification of Pairs of Blocks to Align

Once the separate partitioning of both ontologies is achieved, the evaluation of the proximity between blocks is based on *anchors*, i.e. from previously known

mappings between the terms of both ontologies, defined by string comparison techniques or defined by an expert. The more two blocks contain common anchors, the more they are considered close.

Let $k$ (resp. $k'$) be the number of blocks generated by the partitioning of an ontology $O$ (resp. $O'$) and $B_i$ (resp. $B'_j$) be one of these blocks. Let the function $anchors(B_u, B'_v)$ that calculates the number of anchors shared by two blocks $B_u$ and $B'_v$ and let $\sum_{v=1}^{k'} anchors(B_i, B'_v)$ be the number of anchors contained in a block $B_i$. The *Proximity* relation between two blocks $B_i$ and $B'_j$ is defined as follows:

$$Proximity(B_i, B'_j) = \frac{2.anchors(B_i, B'_j)}{\sum_{u=1}^{k} anchors(B_u, B'_j) + \sum_{v=1}^{k'} anchors(B_i, B'_v)}$$

The aligned pairs of blocks are all the pairs whose proximity is greater than a given threshold $\varepsilon_2 \in [0, 1]$. A block may be aligned with several blocks of the other ontology or with none, depending on the value chosen for this threshold.

**Example.** We applied the PBM algorithm, available online, to two toy ontologies to visualize its behaviour and facilitate later comparison with our own methods.

Figure 1 shows these two ontologies after a partitioning achieved with the control variable representing the maximum size of merged blocks fixed at 3 concepts, i.e. a block exceeding this size cannot be merged. So the blocks thus generated contain at most 6 concepts, and as $O_S$ has 13 concepts, this value ascertained we would get at least 3 blocks.

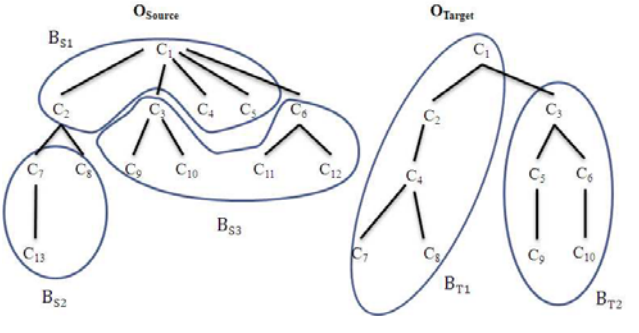



**Fig. 1** The blocks built by PBM

Figure 2 shows the anchors which are supposed to be shared between both ontologies. Block $B_{S1}$ contains 2 anchors, one of which is shared with $B_{T1}$ while the other is shared with $B_{T2}$. Block $B_{S2}$ only contains one anchor, shared with $B_{T1}$. Block $B_{S3}$ contains 3 anchors two of which are shared with $B_{T1}$ while the third is shared with $B_{T2}$.

Shared-anchors based proximity calculations must be performed on every possible pairs of blocks (6 pairs in this case). As the $(B_{S1}, B_{T1})$ pair only has one common

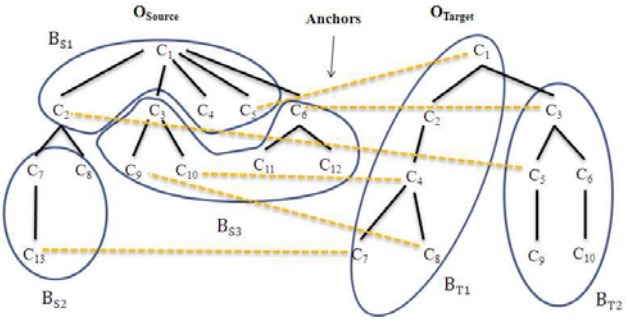



**Fig. 2** Anchors identification

anchor while the blocks have, in order, 2 and 4 common anchors, Proximity($B_{S1}$, $B_{T1}$) = 0.33. The other results are: Proximity($B_{S1}$, $B_{T2}$) = 0.5, Proximity($B_{S2}$, $B_{T1}$) = 0.4, Proximity($B_{S2}$, $B_{T2}$) = 0, Proximity($B_{S3}$, $B_{T1}$) = 0.57, Proximity($B_{S3}$, $B_{T2}$) = 0.4.

The number of pairs actually aligned varies according to the threshold value. Lowering the threshold multiplies alignments and the chances one has of finding mappings, but also increases runtime costs. With a high threshold, less time is spent aligning far blocks but this can result in the loss of potential mappings. When the threshold is set to 0.4, the ($B_{S1}$, $B_{T1}$) pair is not aligned and the common anchor is not discovered in the mappings. When the threshold is set to 0.33, all the anchors are discovered in the mappings, but every possible pair of blocks, except the pair without common anchor ($B_{S2}$, $B_{T2}$) has to be aligned.

This method allows PBM to decompose very large ontologies. Nevertheless this decomposition is made a priori, without taking into account the objective of alignment, because it is applied to each ontology independently from the other. Partitioning is done blindly, some anchors may not be in the blocks finally aligned and the resulting alignment does not necessarily include all desired mappings. Finally, the calculation of the relevant blocks to be aligned is expensive (in processing time).

Despite these criticisms, the decomposition algorithm PBM is, among all existing partitioning algorithms, the most adapted to the task of alignment since it allows control of the maximum size of the generated blocks.

We propose two methods that reuse this algorithm by modifying how it generates blocks. Our idea is to consider, as soon as possible during the partitioning, all the existing data relative to the alignment between the concepts of both ontologies and to try to simulate, at least in the second method, co-clustering.

## 3 Alignment Oriented Partitioning Methods

To take into account as soon as possible the objective of alignment, our methods are going to lean on two facts: on one hand the couples of concepts stemming from both ontologies which have exactly the same label and can be connected by a relation of equivalence and on the other hand the possible structural asymmetry of the two ontologies to be aligned.

Even on large ontologies, it is possible to identify, with a similarity measure strict and inexpensive to calculate, concepts which have a label in common across ontologies. As in PBM, we call these couples concept *anchors* but we will use them to generate partitions.

The structural asymmetry of both ontologies is used to order their partitioning and to choose the method to do it: if one ontology is more structured than the other, it will be easier to decompose it into blocks with a strong internal cohesiveness and its decomposition can serve as a guide for the decomposition of the other ontology. In what follows, the most structured ontology will be called the *target*, $O_T$ and the less structured, the *source*, $O_S$. The first method that we propose, called PAP (*Partition, Anchor, Partition*), consists in beginning by decomposing the target $O_T$, then by using identified anchors, to force the partitioning of $O_S$ to follow the pattern of $O_T$. In so doing, this first method partially breaks the structure of the source $O_S$. This is not a problem when the source is poorly structured.

However, if $O_S$ is well-structured, the PAP method is inadequate and we suggest another partitioning method, called APP (*Anchor, Partition, Partition*) which follows more closely the structure of both ontologies. The APP method partitions $O_T$ by favoring the fusion of blocks sharing anchors with $O_S$, and partitions $O_S$ by favoring the fusion of blocks sharing anchors with the same block generated from $O_T$.

### *3.1 The* PAP *Method*

The PAP method consists in beginning by decomposing the target $O_T$, then by forcing the partitioning of $O_S$ to follow the pattern of $O_T$. To achieve this, the method identifies for each block $B_{Ti}$ from $O_T$ all the anchors belonging to it. Each of these sets will constitute the kernel or *center* $CB_{Si}$ of a future block $B_{Si}$ to be generated from the source $O_S$. The alignment of the pairs of blocks allows to find, in the final step of alignment, all the equivalence relations between anchors. The PAP method consists of four steps besides the calculation of anchors:

**Partition $O_T$ into several blocks $B_{Ti}$.** Partitioning is done according to the PBM algorithm.

**Identify the centers $CB_{Si}$ of the future blocks of $O_S$.** The centers of $O_S$ are determined from two criteria: the pairs of anchors identified between $O_S$ and $O_T$, and the blocks $B_{Ti}$ built from the target ontology $O_T$.

Let the function $Anchor(E, E')$, whose arguments $E$ and $E'$ are each an ontology or a block, returns all concepts of $E$ which have the same label as the concepts of

$E'$. For each block $B_{Ti}$ built in the previous step, the centers of future blocks of $O_S$ are calculated as follows:

$$CB_{Si} = Anchor(O_S, B_{Ti})$$

**Partition the source $O_S$ around the centers $CB_{Si}$.** After identifying the centers of the future blocks $O_S$, we apply the PBM algorithm with the following difference. Instead of inputing the set of the $m$ concepts as $m$ blocks, each reduced to a single concept, we introduce the $n$ centers identified in the previous step, as distinct blocks but with several concepts and other concepts of $O_S$ that have no equivalents in $O_T$, each one in an individual block. The cohesiveness of the blocks representing the centers $O_S$ is initialized with the maximum value.

**Identifying the pairs of blocks to align.** Each block $B_{Si}$ built from a center is aligned with the corresponding block $B_{Ti}$. The algorithm can lead to the constitution of $B_{Sj}$ blocks containing no anchors and which, in the current state of our implementation, are not taken into account in the matching process. The treatment of these blocks without anchors is a perspective of this work, still under study.

**Example.** On the toy example presented earlier, Fig. 3 shows first the decomposition of $O_T$ achieved by the PBM algorithm, then the identification of the centers $CB_{Si}$ of the future blocks of $O_S$. These will be built from the blocks generated for target $B_{T1}$ and $B_{T2}$. $CB_{S1} = \{c_5, c_9, c_{10}, c_{13}\}$ and $CB_{S2} = \{c_2, c_6\}$.

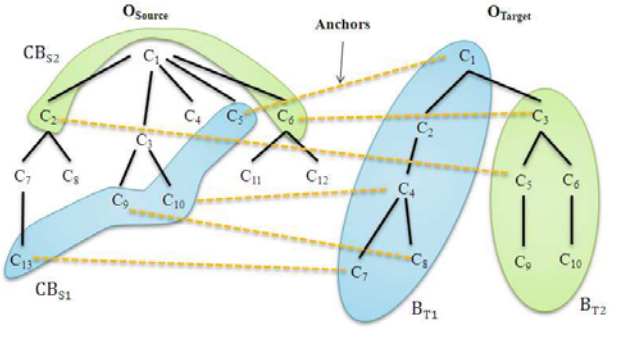



**Fig. 3** The centers $CB_{Si}$ identified from $B_{Ti}$

Figure 4 shows $O_S$ blocks resulting from the partition of $O_S$ around the centers. The test on the maximum size of constructed blocks being performed according to *PAP* method after the initial block grouping, block $B_{S1}$ becomes larger than the size limit so no other block can be grouped with it. It is the same for $B_{S2}$. Thus this partitioning reveals a block without anchor, $B_{S3}$, which will not be aligned. The aligned pairs are ($B_{S1}$, $B_{T1}$) and ($B_{S2}$, $B_{T2}$), immediately identifiable by construction.

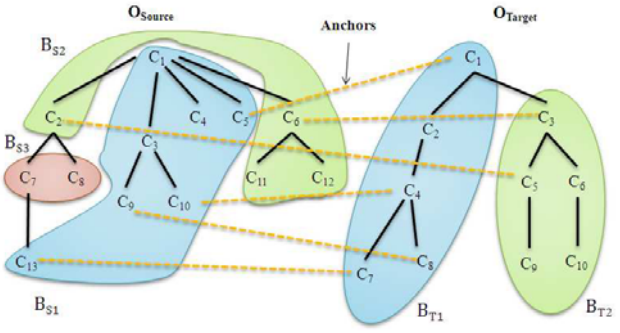



**Fig. 4** Partition of $O_S$ around the centers $CB_{Si}$ identified in precedent step

## 3.2 *The* APP *Method*

The idea of this method is to partition both ontologies at the same time, i.e. to co-cluster. The problem is that we cannot really treat these ontologies in parallel because of their large size. To simulate the parallelism, we partition the target ontology by favoring the fusion of blocks sharing anchors with the source, and we partition the source by favoring the fusion of blocks sharing anchors with the same block generated from the target. Then we take into account the equivalence relations between ontologies identified since the partitioning of $O_T$, which makes the search for resembling blocks easier and improves alignment results. Unlike the PBM algorithm and our PAP method, this partitioning method is alignment-oriented: it simplifies the subsequent task of aligning both ontologies. The APP method has three steps:

**Generate $O_T$ blocks.** To generate blocks of the target $O_T$, we use the PBM algorithm by modifying the definition of the *goodness* measure to take into account the equivalence relations between both ontologies. We add a coefficient representing the proportion of anchors that are shared in a block $B_j$ of $O_T$. The more anchors a block contains, the more this coefficient increases its cohesiveness or its coupling value respectively to other blocks. As a result, during the generation of blocks, the choice of the block that has the maximum value of cohesiveness or coupling depends not only upon relations between concepts inside or outside the blocks of $O_T$, but also upon the anchors shared with $O_S$.

Let $\alpha \in [0,1]$, $B_i$ and $B_j$ be two blocks of $O_T$, $|Anchor(B_j, O_S)|$ represents the number of anchors in $B_j$ and $|Anchor(O_T, O_S)|$ represents the total number of anchors. The *goodness* equation becomes:

$$goodness(B_i, B_j) = \alpha \left( \frac{\sum_{c_i \in B_i, c_j \in B_j} Link_S(c_i, c_j)}{sizeOf(B_i) \cdot sizeOf(B_j)} \right) + (1-\alpha) \frac{|Anchor(B_j, O_S)|}{|Anchor(O_T, O_S)|}$$

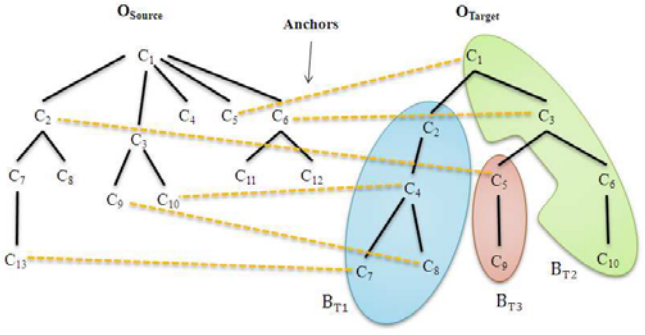



**Fig. 5** Built blocks from $O_T$ by the APP method

**Generate $O_S$ blocks.** Again we modify the *goodness* measure to take into account at the same time the values of links between $O_S$ concepts, the anchors shared between both ontologies and the blocks built for $O_T$. Let the block $B_i$ of $O_S$ be the block with the maximum value of cohesiveness and the block $B_k$ of $O_T$ be the block which shares the highest number of anchors with $B_i$. The new calculation of *goodness* favors the fusion of $B_i$ with $B_j$, which contains the highest number of anchors in common with $B_k$. This gathers in a single source block the anchors shared with one target block.

Let $\alpha \in [0,1]$, $B_i$ and $B_j$ be two distinct blocks of $O_S$. Let $B_k$ be the block of $O_T$ which shares the highest number of anchors with $B_i$. The *goodness* equation becomes:

$$goodness(B_i,B_j) = \alpha \left( \frac{\sum_{c_i \in B_i, c_j \in B_j} Link_S(c_i,c_j)}{sizeOf(B_i) \cdot sizeOf(B_j)} \right) + (1-\alpha) \frac{|Anchor(B_j,B_k)|}{|Anchor(O_T,O_S)|}$$

**Identification of blocks pairs.** The alignment is done between the blocks sharing the highest number of anchors; a block of $O_S$ can only align itself with a single block of $O_T$.

**Example.** Figures 5 and 6 also display results obtained upon our toy example. Fig. 5 shows the blocks from $O_T$ built according to the APP method, favoring anchor grouping. Fig. 6 shows the blocks built in $O_S$, favoring the construction of blocks sharing anchors with these of $O_T$ while taking the structure of $O_S$ into account.

Every source block is only aligned once with the block with which it has the greatest number of common anchors, identified by construction. So we align the pairs $(B_{S1}, B_{T1})$, $(B_{S3}, B_{T1})$ and $(B_{S2}, B_{T2})$.

$B_{T3}$ takes no part in the alignment process because it shares its single common anchor with $B_{S1}$ and $B_{S1}$ has more anchors in common with $B_{T1}$ than with $B_{T3}$.

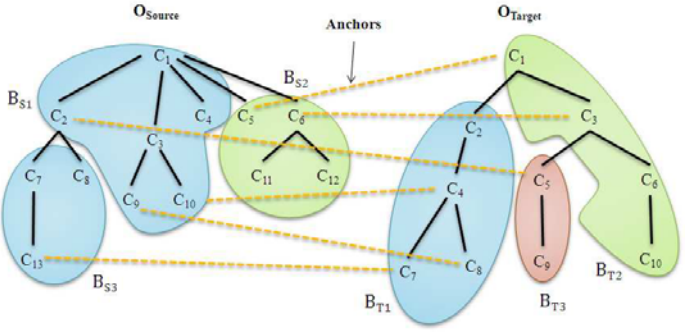



**Fig. 6** Built blocks from $O_S$ by the APP method

This results in the loss of an anchor match, $(c_2, c_5)$, but reduces alignment runtime. We can hope that the co-clustering building of the blocks takes inter-ontologies relationships more into account.

## 4 Experiments

We have implemented the two methods presented previously and experiments were made on various ontologies in order to compare partitioning methods through their suitability for alignment. Generated blocks were aligned by pairs using the alignment software developed within our team, *TaxoMap* (Hamdi *et al.*, 2008).

The experiments were first realized on ontologies in the geographical area, supplied by COGIT[6]. These ontology sizes are limited so it is possible to align them directly - without having to partition - and to obtain reference mappings. They are also well known in the team which enabled us to analyse the semantic relevance of the generated blocks. Other experiments were then made on two pairs of large ontologies which our tool fails to align because of scalability problems.

### *4.1 Experiments on Geographic Ontologies*

*Target* ontology BDTopo, is composed of 612 concepts related by subsumption links in a hierarchy seven levels deep. *Source* ontology BDCarto includes 505 concepts in a hierarchy of depth 4. The results of the direct alignment carried out without ontologies partitioning are presented in Table 1.

To make the partitions, the maximum size of merger blocks was fixed at 100 concepts, i.e. a block exceeding this size cannot be merged. So the blocks thus generated contain at most 200 concepts. Table 2 lists the number of blocks generated for each ontology.

[6] The COGIT laboratory (Conception Objet et Généralisation de l'Information Topographique), National Geographical Institute.

**Table 1** Relations identified by aligning BDCarto to BDTopo

| Ontologies | Target Size | Source Size | isEq | isClose | isA | $\Sigma$ |
|---|---|---|---|---|---|---|
| BDTopo-BDCarto | 612 | 505 | 197 | 13 | 95 | 305 |

**Table 2** Partitioning of BDTopo and BDCarto with the different methods

| | | Target Ontology BDTopo | | | Source Ontology BDCarto | | |
|---|---|---|---|---|---|---|---|
| Methods | Anchors | Generated blocks | Isolated concepts | Largest block | Generated blocks | Isolated concepts | Largest block |
| PBM | 191 | 5 | 0 | 151 | 25 | 22 | 105 |
| PAP | 191 | 5 | 0 | 151 | 10 | 16 | 143 |
| APP | 191 | 6 | 0 | 123 | 10 | 16 | 153 |

Target ontology BDTopo is the main ontology for COGIT. It is well constructed, compact and highly structured. The root is only linked to two direct children of depth 1, which are direct parents to a limited number of nodes. It is easy to partition into semantically relevant blocks, whether by the PBM method which is mainly based on the structural relations between concepts, by the PAP method, which uses the PBM algorithm for the partitioning of the target and so gives the same results for it, or by the APP method. Both possible decompositions, consisting of 5 or 6 blocks, are relevant.

On the opposite side, source ontology BDCarto is less structured and much dispersed. The root is linked to almost thirty direct children, and many sub-trees contain no more than about ten elements. Decomposition is more delicate. The PBM algorithm generates a big number of small blocks comprising no more than 5 or 6 concepts, 19 blocks do not contain anchors, and 22 blocks contain only one isolated concept. By using the information on the shared anchors, our methods allow to aggregate to larger blocks more than half of these small blocks and many isolated concepts, while maintaining its semantic consistency. The generated partition, less dispersed, is therefore more understandable for the humans and more efficient for the next phase of blocks alignment.

The choice of the pairs of blocks to align differs according to the method used:

**PBM:** among the 25 generated blocks only 6 source blocks contain anchors. Theses 6 blocks are aligned with the target blocks for which the ratio of shared anchors on the sum of anchors present in the two blocks is higher than a given threshold $\varepsilon$, fixed here at 0.1. This threshold is reached by 9 pairs of blocks, 9 alignments are made.

**PAP:** the 5 source blocks, built starting from the 5 blocks of the target which contain anchors, lead in all to 5 alignments.

**APP:** the 7 selected pairs are those which maximize the number of shared anchors of the 7 source blocks containing anchors and which each participates only in one alignment.

Table 3 shows the number of mappings we obtain by matching the different pairs of blocks chosen by our alignment tool, *TaxoMap*. The results presented show that even by matching fewer pairs of blocks than in the PBM method, matching blocks generated by our methods give better results in number of identified mappings.

**Table 3** Relations identified by the alignment of blocks generated by different methods

| Methods | Aligned Pairs | isEq | isClose | isA | Σ | Precision | Recall |
|---|---|---|---|---|---|---|---|
| PBM | 9 | 118 | 13 | 52 | 183 | 0.96 | 0.57 |
| PAP | 5 | 192 | 10 | 55 | 257 | 0.97 | 0.81 |
| APP | 7 | 147 | 11 | 61 | 219 | 0.97 | 0.69 |

If we analyse the results[7] of the two classical alignment measures to compare the relevance of the techniques, the precision (the number of correct mappings identified after partition compared to the full number of returned mappings after partition) and the recall (the number of correct mappings identified after partition compared to the number of reference mappings), we see that our methods have a much better recall. Indeed, these methods take into account the equivalence relations between the labels in the partitioning process, which brings together the concepts that have relations between them in blocks which will be considered thereafter as pairs to align, while the PBM method partitions ontologies independently from each other and makes only an a posteriori alignment. The PAP method allows in particular, by construction, to find all mappings corresponding to the anchors and thus has a higher recall. We are currently working upon heuristics which could be applied, after the partitioning step, on isolated blocks and which would increase the recall of our methods.

The fact that the different methods have a precision lower than 1. means that all three of them find mappings which had not been identified by the alignment of the unpartitioned ontologies. Although these mappings are here considered to be invalid, they are not necessarily wrong. Indeed, for every source concept, our tool produces a single mapping with one concept of the target ontology, that which it considers the best, even if several concepts of the target could be matched. If the two concepts involved in a reference mapping are no longer compared because they are divided into non-aligned blocks, another mapping, which will not necessarily be uninteresting, can be found for the source concept. The study of the quality of these

[7] These results were calculated automatically by the API of alignments evaluation available on the Web, http://oaei.ontologymatching.org/2008/align.html, by providing in reference the file generated by direct alignment by *TaxoMap* without partition.

new mappings, as well as more advanced analysis of the relative qualities of our two methods, will be carried out in complementary work.

## *4.2 Experiments on Large Ontologies*

We tested the two different methods on two pairs of large ontologies (Library and FAO). These pairs of ontologies are used as test in the evaluation campaign OAEI (*Ontology Alignment Evaluation Initiative*) in which alignment tools compete each year on ontologies of diverse sizes and domains.

For both tests (Library and FAO), the comparison between our methods and the PBM method is complex because the FALCON system was not a participant to the 2008 OAEI campaign and we did not participate to the FAO test in the 2007 campaign. Furthermore, as the FAO pair of ontologies was not a test case provided by the 2008 campaign, we did not access the reference mappings. Despite of this, we present in this section, two kinds of experiments. First, we provide a comparison between our results and those obtained by the participants having done the Library test in 2008. Second, we use the FAO test to compare the number of blocks generated by our methods and the PBM algorithm.

### 4.2.1 Library Test

The Library set of tests is made of two thesauri, GTT and Brinkman, in Dutch. These two thesauri are used by the National Library of the Netherlands to indexed the books of two large collections. GTT thesaurus contains 35,194 concepts and Brinkman contains 5,221 concepts. Each concept has (exactly) one preferred label, but also synonyms (961 for Brinkman, 14,607 for GTT). The organizers of the test in 2007 showed that both thesauri have similar coverage (2,895 concepts actually have exactly the same label) but differ in granularity and that the thesauri structural information was very poor. GTT (resp. Brinkman) contains only 15,746 (resp 4,572) hierarchical *broader links*. Its structure being particularly poor (it has 19,752 root concepts), GTT thesaurus was considered as the source in our experiments.

As both ontologies are very imbalanced and as the number of retrieved anchors was limited to 3,535, which is not much with respect to the size of the source, we only experimented with the PAP method. We set the maximum size for a block to be grouped to 500.

**Table 4** Partitioning of Brinkman and GTT

| | | Target Thesaurus Brinkman | | | Source Thesaurus GTT | | |
|---|---|---|---|---|---|---|---|
| Methods | Anchors | Generated blocks | Isolated concepts | Largest block | Generated blocks | Isolated concepts | Largest block |
| PAP | 3 535 | 227 | 0 | 703 | 2 041 | 16 265 | 517 |

The PAP method returned 227 blocks for Brinkman, the larger of which had 703 concepts, and 2,041 blocks for GTT, the larger of which had 517 concepts. 16,265 concepts of GTT remained isolated.

As over 1,800 blocks of GTT contained no anchors, we only aligned 212 pairs and identified 3,217 matches, only 1,872 of which were equivalence relations (ExactMatch).

**Table 5** Relations identified by the alignment of blocks generated by the PAP method

| Methods | Aligned Pairs | isEq | isGeneral | isClose | isA | $\Sigma$ | Precision | Recall |
|---|---|---|---|---|---|---|---|---|
| PAP | 212 | 1 872 | 40 | 274 | 1 031 | 3 217 | 0.88 | 0.41 |

The reason why so few equivalence relations were returned, with respect to the number of identified anchors, is that both thesauri contain a large number of synonyms. We identified 3,535 anchors while only 2,895 concepts were supposed to have the same label. This means that at least 640 anchors concern source concepts, among which at least 2 labels are considered equivalent to 2 other target labels, which are not necessarily associated to the same concept. The problem here is that if a source concept is anchored to 2 distinct target concepts, at best both these target concepts belong to the same block, and the target concept is linked by an ExactMatch relation to only one of these concepts. In the worst case, the 2 target concepts belong to distinct blocks and the PAP method does not know to which block the source concept should be linked. So the PAP method sets it to become an isolated concept.

**Table 6** Results of the systems taking part in the Library test

| Participant | ExactMatch | Precision | Coverage |
|---|---|---|---|
| DSSim | 2 930 | 0.93 | 0.68 |
| TAXOMAP | 1 872 | 0.88 | 0.41 |
| Lily | 2 797 | 0.53 | 0.37 |

Even though several anchors have disappeared, precision and coverage evaluated only upon equivalence relations (ExactMatch) by the organizers of the test, and presented in Table 6, place our system TAXOMAP running the PAP method, in reasonable position. Among the other two participants, DSSim [8] (Nagy *et al.*, 2008) got better results than us but Lily [9] (Wang and Xu, 2008) did worse.

[8] The authors of DSSim say they partition the ontologies but do not explain how.

[9] The authors of Lily say they process the ontologies according to a method which is not based upon partitioning but they refer to a yet unpublished article.

### 4.2.2 FAO Test

The FAO set of tests (2007) comprises two ontologies : AGROVOC and NALT, which consist respectively of 28 439 and 42 326 concepts. AGROVOC is a multilingual ontology built by FAO (Food and Agriculture Organization). It covers the fields of agriculture, forestry, fisheries, environment and food. NALT is the thesaurus of NAL (National Agricultural Library) on the same subject.

The most important ontology, NALT, is used as the target and AGROVOC is used as the source. The maximum size of merger blocks is fixed at 2 000 concepts.

**Table 7** Partitioning of AGROVOC and NALT

| | | Target Ontology NALT | | | Source Ontology AGROVOC | | |
|---|---|---|---|---|---|---|---|
| Methods | Anchors | Generated blocks | Isolated concepts | Largest block | Generated blocks | Isolated concepts | Largest block |
| PBM | 14 787 | 47 | 4 | 3 356 | 318 | 492 | 2 830 |
| PAP | 14 787 | 47 | 4 | 3 356 | 252 | 199 | 2 939 |
| APP | 14 787 | 47 | 4 | 3 118 | 95 | 199 | 3 534 |

Despite there are no reference mappings which make possible to analyse the quality of produced alignments, we nevertheless present the results of partitioning in Table 7 because they seem us relevant. Table 7 shows that in this experiment, as in the previous one, partitioning according to our methods minimized the number of isolated concepts, and in particular according to the APP method, minimized the number of generated blocks, leading to partitions that might be less dispersed.

Among the 47 blocks built for $O_T$ according to the PAP method, only 42 contain anchors. So 210 of the 252 blocks built for $O_S$ take no part in the alignment process which matches 42 pairs of blocks. The APP method matches 25 pairs of blocks.

## 5 Conclusion

As current tools for ontology alignment lose their effectiveness on large ontologies, the objective of this work was to study the techniques of ontology partitioning oriented towards the alignment task.

The two methods we propose take the PBM algorithm for ontology partitioning, developed for the alignment system, but instead of applying the algorithm, as PBM, successively and independently on each ontology, we try to take into account as soon as possible in the partitioning process the context of the alignment task.

Our methods are applied on two ontologies simultaneously, and use alignment-related data. These alignment-related data are easy to extract, even from large ontologies. They include pairs of concepts, one concept from each ontology, which have the same label, and structural information on the ontologies to align.

The PAP method is well suited for ontologies of a dissymmetrical structure. It starts by decomposing the best structured ontology and then forces the decomposition of the

second ontology following the same pattern. The APP method can be applied when both ontologies are well structured. It favors the generation of blocks of concepts, which are related, from one ontology to the other, by equivalence links.

The fact that the partitioning algorithms only use data easy to extract, in a light treatment, allows very large ontologies to be partitioned. It is thus a scalable approach.

Our methods were tested on different ontology couples. The results presented here show that they can build partitions less dispersed by limiting the number of generated blocks and isolated concepts. For the experiment where we have reference mappings, we have been able to see that our partitions lost fewer mappings.

We are currently working upon heuristics which could be applied, after the partitioning step, on isolated blocks and which would increase the recall of our methods.

We currently continue the experiments to analyse the qualities of our two methods when both ontologies are heavily unbalanced (in terms of size and structure) or when the number of concepts with identical labels is limited.

**Acknowledgements.** This research was supported by the French National Research Agency (ANR), through the GeOnto project ANR-O7-MDCO-005 on "Creation, Comparison and Exploitation of Heterogeneous Geographic Ontologies" (http://geonto.lri.fr/) and through the WebContent project ANR RNTL.

## References

Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. The Scientific American, 34–43 (2001)

Grau, B.C., Parsia, B., Sirin, E., Kalyanpur, A.: Automatic Partitioning of OWL Ontologies Using e-connections. In: DL 2005, Proceedings of the 18th International Workshop on Description Logics (2005)

Guha, S., Rastogi, R., Shim, K.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. Information Systems 25(5), 345–366 (2000)

Haase, P., Honavar, V., Kutz, O., Sure, Y., Tamilin, A. (eds.): Proceedings of the 1st International Workshop on Modular Ontologies, WoMO 2006, co-located with the International Semantic Web Conference, ISWC 2006. CEUR Workshop Proceedings, vol. 232 (2007), CEUR-WS.org

Hamdi, F., Zargayouna, H., Safar, B., Reynaud, C.: TaxoMap in the OAEI 2008 Alignment Contest. In: Proceedings of the 3th International Workshop on Ontology Matching, OM 2008 (2008)

Hu, W., Qu, Y., Cheng, G.: Matching large ontologies: A divide-and-conquer approach. Data Knowl. Eng. 67(1), 140–160 (2008)

Hu, W., Zhao, Y., Qu, Y.: Partition-Based Block Matching of Large Class Hierarchies. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 72–83. Springer, Heidelberg (2006)

Nagy, M., Vargas-Vera, M., Stolarski, P., Motta, E.: DSSim Results for OAEI. In: Proceedings of the 3th International Workshop on Ontology Matching, OM 2008 (2008)

Noy, N.F., Musen, M.A.: PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In: AAAI/IAAI, pp. 450–455 (2000)

Shvaiko, P., Euzenat, J.: A Survey of Schema-Based Matching Approaches. Journal on Data Semantics IV, 146–171

Rahm, E., Bernstein, P.A.: A survey of approaches to automatic schema matching. VLDB Journal: Very Large Data Bases 10(4), 334–350 (2001)

Reynaud, C., Safar, B.: Techniques structurelles d'alignement pour portails web. RNTI, Revue des Nouvelles Technologies de l'Information (2007)

Stuckenschmidt, H., Klein, M.: Structured-Based Partitioning of Large Concept Hierarchies. In: McIlraith, S.A., Plexousakis, D., van Harmelen, F. (eds.) ISWC 2004. LNCS, vol. 3298, pp. 289–303. Springer, Heidelberg (2004)

Wang, P., Xu, B.: Lily: Ontology Alignment Results for OAEI 2008. In: Proceedings of the 3th International Workshop on Ontology Matching, OM 2008 (2008)

Wang, Z., Wang, Y., Zhang, S., Shen, G., Du, T.: Matching Large Scale Ontology Effectively. In: Mizoguchi, R., Shi, Z.-Z., Giunchiglia, F. (eds.) ASWC 2006. LNCS, vol. 4185, pp. 99–105. Springer, Heidelberg (2006)

Wu, Z., Palmer, M.: Verb semantics and lexical selection. In: Proc. 32nd Annual Meeting of the Association for Computational Linguistics (ACL), pp. 133–138 (1994)

# Learning Ontologies with Deep Class Hierarchies by Mining the Content of Relational Databases

Farid Cerbah

**Abstract.** Relational databases are valuable sources for ontology learning. Previous work showed how precise ontologies can be learned and be fruitfully exploited to solve practical problems, such as ensuring integration and interoperation of heterogeneous databases. However, a major persisting limitation of the existing approaches is the derivation of ontologies with flat structure that simply mirror the schema of the source databases. In this paper, we present the RTAXON learning method that shows how the content of the databases can be exploited to identify categorization patterns from which class hierarchies can be generated. This fully formalized method combines a classical database schema analysis with hierarchy mining in the stored data. RTAXON is one of the methods implemented in RDBToOnto, a comprehensive tool that support the transitioning process from access to the data to generation of populated ontologies.

**Keywords:** Ontology Learning, Data Mining, Maximum Entropy, Relational Databases, Reverse Engineering.

## 1 Introduction

In companies that need to produce and manage technical knowledge on complex engineering assets, as in aerospace and automotive industries, a large proportion of technical corporate repositories are built upon relational databases. These repositories are without doubt among the most valuable sources for building highly accurate and effective domain ontologies. However, undertaking such a transitioning process to ontologies without adequate software support might be deemed too tedious and costly by many practitioners. In this context, the availability of robust learning tools

Farid Cerbah
Dassault Aviation, 78, quai Marcel Dassault, 92552 Saint-Cloud Cedex 300 France
e-mail: farid.cerbah@dassault-aviation.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 271–286.
springerlink.com 

to derive ontologies from relational databases can be a strong argument to convince potential adopters.

Ontology learning from relational databases is not a new research issue. Several approaches and tools have been developed to deal with such structured input. Past contributions showed how precise ontologies can be learned and be fruitfully exploited to solve practical problems, such as ensuring integration and interoperation of heterogeneous databases. However, a major persisting limitation of the existing methods is the derivation of ontologies with flat structure that simply mirror the schema of the source databases. Such results do not fully meet the expectations of users that are primarily attracted by the rich expressive power of semantic web formalisms and that could hardly be satisfied with target knowledge repositories that look likes their source relational databases. A natural expectation is to get at the end of the learning process ontologies that better capture the underlying conceptual structure of the stored data.

Ontologies with flat structure is the typical result of learning techniques that exclusively exploit information from the database schema without (or just marginally) considering the data. A careful analysis of existing databases shows that additional definition patterns can be learned from the data to significantly enrich the base structure. More particularly, class hierarchies can be induced from the data to refine classes derived from the relational schema.

In this paper, we define a comprehensive approach to ontology learning from relational databases that combines two complementary information sources: the schema definition and the stored data. We show how the content of the databases can be exploited to identify categorization patterns from which class hierarchies can be generated.

The remainder of the paper is organized as follows. We introduce in Sect. 2 a motivating example to illustrate the idea of combining the two sources in the learning process. In Sect. 3, we review previous work on ontology learning applied to relational databases and the related issue of database reverse engineering. Sect. 4 is the core of this contribution. We give an extensive description of our formalized approach. Sect. 5 describes the experiments we conducted to validate the implemented method. Section 6 provides an overview of RDBToOnto, the tool in which the learning method is implemented. Then, we conclude with some directions for further research.

## 2 A Motivating Example

We start by depicting the typical transitioning process on an academic example. Figure 1 shows the input and the potential output of such of a process when applied on an extract of a database from a food delivery service application.

The derivations applied to get the target ontology can be divided in two interrelated parts. The first part, named **(a)** in the figure, includes derivations that are motivated by the identification of patterns from the database schema. In this example, each relation (or table) definition from the schema is the source of a class

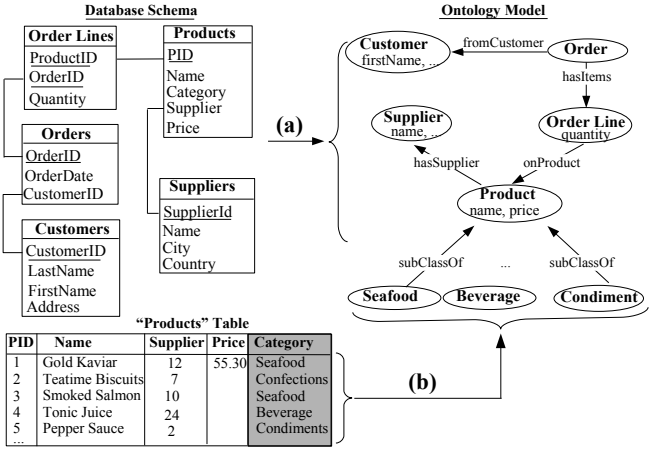



**Fig. 1** An example of ontology building by exploiting both the schema and the data

in the ontology. Such simple mappings from relations to classes are often relevant though many exceptions need to be handled (for instance, as we will see later on, some relations are more likely to be translated as class-to-class associations). To complete the class definitions, data type properties are derived from some of the relation attributes. The links between tables in the schema show the external references made through foreign key relationships (the primary keys are the underlined attributes in the relation definitions). These binary key-based associations are the most reliable input for linking classes and, in our example, each of the four foreign key relationships is translated into an object property.

The derivations applied to obtain this upper part of the ontology are well covered by current methods and, if applied on this database example, most of the methods would actually provide this result as output. However, by looking closer at the data, it can be noticed that the process could go further. In the Products table, we can see that additional structuring patterns could be exploited to make the ontology more accurate. More particularly, the **(b)** part of the derivations shows how the Product class can be refined with subclasses derived from the values of Category column in the source Products table. In the same vein, the Supplier class could be extended with a two-level hierarchy by interpreting the values in both City and Country columns of the corresponding table (resulting in subclasses Sweden Supplier $\longrightarrow$ Stockholm Supplier, Göteborg Supplier, etc).

These are typical examples of subsumption relations that can be discovered by mining the database content. One of the key issues addressed in this work is the identification of relation attributes that may serve as good *categorization sources* and we show how these specific learning mechanisms can be consistently integrated in a comprehensive learning approach to the global ontology construction problem.

## 3 Related Work

Ontology learning from relational databases is a relatively recent issue. However, it can benefit from early work in the domain of database reverse engineering whose goal was to extract object-oriented models from relational models (Behm *et al.*, 1997; Premerlani and Blaha, 1994; Ramanathan and Hodges, 1997). The core of the transformation rules for database reverse engineering purposes are still relevant in the context of ontology learning. The most reliable rules have been reused as a starting point and extended in several approaches that have ontologies as target models (Stojanovic *et al.*, 2002; Astrova, 2004; Li *et al.*, 2005).

Most approaches are based on an analysis of the relational schemas. However, to some extent, analysis of the database content has been investigated yet both in reverse engineering and ontology learning. In Tari *et al.* (1997), the reverse engineering method includes a thorough analysis of *data correlations* between tuples in order to identify additional relations not made explicit in the schema. The impact of these data exploration mechanisms are limited since *data correlation* in this context is to be interpreted as correlations between key values, i.e. between identifiers of the tuples[1]. However, the non-key attributes are not considered. In Astrova and Stantic (2004), the same key value correlations are reinterpreted in the context of ontology learning. In addition, the analysis is extended by considering equality and overlap between attribute *names*, but attribute extensions are not involved in the process.

Several approaches are including in their transitioning process specific rules for identifying inheritance relationships. However, efforts are also concentrated on relationships based on key value correlations, and more particularly on key value inclusion. For example, in the schema of figure 1, the Customers table might have been refined by the designers with the additional tables PrivateCustomer and CompanyCustomer. A way to link these specialized tables to the main Customers table would be to share primary key values. In practice, the rules based on the identification of such key-based constructs are not the most productive since these modelling schemes are only found in carefully designed databases.

In Lammari *et al.* (2007), the identification of "Generalization/Specialization" relations is based on a precise interpretation of null value semantics. This approach somewhat takes advantage of the unsuitability of the relational model to properly express inheritance. Typically, when all instances (i.e tuples) of a generic concept and its sub-concepts are gathered into a single table, some attributes may only be relevant for some subconcepts, and thus filled for instances of these sub-concepts but left empty for the others. For example, in an Employees relation that includes all employees of a flight company, attributes FlightHours and LicenceNumber would be filled with null values in entries corresponding to non-members of the flight staff. Partitioning of a relation on the basis of null values and their cooccurrences may reveal the underlying concept hierarchy. This hierarchy identification approach

[1] The basic idea is to exploit the fact that "objects" unambiguously identified by unique keys all over the database may have their descriptions spread in different tables. The key value correlations may reveal the implicit links between the involved tables.

involves an analysis of non-key attribute extensions. On this respect, this approach and the one we propose in this paper fall in the same category.

Some approaches are using forms, often automatically generated from the source database (Astrova, 2004; Benslimane *et al.*, 2007). Though forms can hardly be considered as the primary source compared to the database schema and data, such more or less structured input can be used as a complementary source. More particularly, to name elements of the resulting ontology model, user-oriented concept names found in the forms can be more explicit than those given to relations and attributes in the relational schema.

As a related issue, mapping languages (Bizer, 2003; de Laborda and Conrad, 2005; Barrasa *et al.*, 2004) are declarative means that provide convenient ways to map relational models to pre-existing ontologies and to automatically generate instances from the data. For database integration needs, *Relational.OWL* (de Laborda and Conrad, 2005) is an ontology of the relational model that can be used as a neutral representation to ensure interoperability of database systems.

## 4 Combining Schema and Data Analysis

The primary motivation in the design of the RTAXON method was to combine the most robust rules for exploiting relational schemas with data mining focused on the specific problem of concept hierarchy identification. One of the key issues addressed in this work is the identification of relation attributes that may serve as good categorization sources and we show how these specific learning mechanisms can be coherently integrated into a comprehensive learning approach to ontology construction. In this prominent part of the paper, we start by introducing some basic notations and definitions that will be used to describe our approach. Then, we outline the steps of the overall ontology learning process (Sect. 4.2) before providing a detailed description of the data mining step for hierarchy identification (Sect. 4.3).

### *4.1 Preliminary Definitions*

A *relational database schema* $D$ is defined as a finite set of relation schemas $D = \{R_1, \ldots, R_n\}$ where each *relation schema* $R_i$ is characterized by its finite set of attributes $\{A_{i1}, \ldots, A_{im}\}$. A function *pkey* associates to each relation its primary key which is a set of attributes $K \subseteq R$.

A relation $r$ on a relation schema $R$ (i.e. an *instance* of $R$) is a set of tuples which are sequences of $|R|$ values. Similarly, a database $d$ on $D$ is defined as a set of relations $d = \{r_1, \ldots, r_n\}$. By convention, if a relation schema is represented by a capital letter, the corresponding lower-case letter will denote an instance of the relation schema.

A *projection* of a tuple $t$ on a set of attributes $X \subseteq R$, denoted $t[X]$, is defined as a restriction on $t$, resulting in the subsequence with values corresponding to attributes of $X$. The projection of a relation $r$ on $X$, denoted $\pi_X(r)$, is defined by $\pi_X(r) = \{t[X] \mid t \in r\}$.

The concept of *inclusion dependency* (e.g. Abiteboul *et al.* 1995) is used to account for correlations between relations. An inclusion dependency is an expression $R[X] \subseteq S[Y]$ where $X$ and $Y$ are respectively attribute sequences of $R$ and $S$ relation schemas, with the restriction $|X| = |Y|$. The dependency holds between two instances $r$ and $s$ of the relation schemas if for each tuple $u$ in $r$ there is a tuple $v$ in $s$ such that $u[X] = v[Y]$. Informally, an inclusion dependency is a convenient way to state that data items are just copied from another relation.

*Foreign key relationships* can be defined as inclusion dependencies satisfying the additional property: $Y = pkey(S)$. The notation $R[X] \subseteq S[pkey(S)]$ will be used for these specific dependencies.

Formal descriptions of ontology fragments will be expressed in OWL abstract syntax.

## *4.2 The Overall Process*

The transformation process is basically a composition of automated steps. The main steps of this process are: database normalization, class and property learning and ontology population.

It should be mentioned that some of the features and processing steps involved in this specific method are reused from the RDBToOnto framework (see Sect. 6) and can be exploited in the design and implementation of other methods.

**- Database Normalization**

In early approaches, this stage is not integrated in the learning process. It is quite common to consider as input relational databases that are in some normal form, often 2NF or 3NF (e.g. Abiteboul *et al.* 1995). It is assumed that the transformation process can be easily extended to cope with ill-formed input by incorporating at the early stages of the process a normalization step based on existing algorithms. Though theoretically acceptable, this assumption has some drawbacks in practice. Many databases we used to experiment the process had redundancy problems and substantial normalization efforts were often required to build up acceptable input for ontology construction. More particularly, data duplication between relations is a recurring problem that might have a negative impact on the resulting ontologies. Such data duplications can be formalized as inclusion dependencies where the set of attributes from the source relation are not restricted to the primary key (i.e inclusion dependencies that are not foreign key relationships). To eliminate these duplications, the database need to be transformed by turning all these inclusion dependencies into foreign key relationships. More formally, each attested dependency $R[X] \subseteq S[Y]$ with $Y \neq pkey(S)$ is replaced by the foreign key relationship $R[A] \subseteq S[pkey(S)]$ where $A$ is a newly introduced foreing key attribute, and all non key attributes in $X$ together with corresponding data in $r$ are deleted from the relation.

This preliminary step is semi-automated as the inclusion dependencies to be processed are defined manually and the database transformation is performed automatically.

**- Class and Property Identification**

This is the core step of the ontology learning process where relations of the database are explored to derive parts of the target ontology model. The database schema is the first information source exploited through the application of prioritized rules that define typical mappings between schema patterns and ontology elements, namely classes, datatype and object properties. We give in table 1 three of the most reliable rules which are also employed in several existing approaches. The first trivial rule states that every relation can potentially be translated as a class though relations can be consumed by more specific rules with higher priority, such as the third rule. The second rule is also a simple mapping from a foreign key relationship to a functional object property. The third rule is intended to match a relation with a composite primary key and two key-based attributes. Such bridging relations are only introduced in the database to link two other relations through key associations. They are turned into many-to-many object properties.

Content of the relations is the second information source allowing to refine with subclasses some of the classes obtained by applying schema-based mapping rules. This additional part is central in the RTAXON method (see Sect. 4.3).

**- Ontology Population**

Final step aims at generating instances of classes and properties from the database content. For a given class, an instance is derived from each tuple of the source relation. Moreover, if refinement into subclasses has been successfully applied on the class, the instances need to be further dispatched into the subclasses.

## *4.3 Extracting Hierarchies from the Data*

In Sect. 2, we introduced through an example the issue of hierarchy mining from database content, showing how classes derived from the schema can be refined with subclasses extracted from the stored data.

More specifically, our motivating example provides illustration of some modelling patterns attested in many databases where specific attributes are used to assign categories to tuples. These frequently-used patterns are highly useful for hierarchy mining as values of *categorizing attributes* can be exploited to derive subclasses.

Our method for hierarchy mining is focused on exploiting the patterns based on such categorizing attributes. Finding these patterns is a difficult task compared to the matching operations required to identify patterns which are based solely on the schema definition, as in rules of table 1.

We describe below the pattern identification procedure. Then, we discuss the generation of the subclasses from the identified patterns.

### 4.3.1 Identification of the categorizing attributes

Two sources are involved in the identification of categorizing attributes: the names of attributes and the redundancy in attribute extensions (i.e. in column data). These

**Table 1** Three reliable rules that match patterns in the database schema. In the **Target** part, the variable in bold holds the Uri of the generated ontology fragment. *sourceOf* assertions provide traceability to control the process

**Relation to Class**

| Source | Preconditions | Target |
|---|---|---|
| $R \in D$ | $\neg \exists\, C \mid R = sourceOf(C)$ | class($\boldsymbol{C_R}$) |

**Foreign key Relationship to Functional Object Property**

| Source | Preconditions | Target |
|---|---|---|
| $R_0[A] \subseteq R_1[pkey(R_1)]$ | $R_0 = sourceOf(C_0)$<br>$R_1 = sourceOf(C_1)$ | ObjectProperty($\boldsymbol{P_A}$<br>domain($C_0$)<br>range($C_1$)<br>Functional) |

**Composite Key Relation to Object Property**

| Source | Preconditions | Target |
|---|---|---|
| $R_0 \in D$<br>$\lvert R_0 \rvert = 2$<br>$pkey(R_0) = \{K_1, K_2\}$<br>$R_0[K_1] \subseteq R_1[pkey(R_1)]$<br>$R_0[K_2] \subseteq R_2[pkey(R_2)]$ | $R_1 = sourceOf(C_1)$<br>$R_2 = sourceOf(C_2)$ | ObjectProperty($\boldsymbol{P_R}$<br>domain($C_1$)<br>range($C_2$)) |

two sources are indicators that allow to find attribute candidates and select the most plausible one.

**- Identification of lexical clues in attribute names**

Categorizing attributes are often *lexically marked*. When used for the purpose of categorization, the attributes may bear names that reveal their specific role in the relation (i.e. classifying the tuples). In example of figure 1, the categorizing attribute in the Products relation is clearly identified by its name (Category). The lexical clue that indicates the role of the attribute can just be a part of a compound noun or of an abbreviated form, as in the attribute names CategoryId or CatId. Our candidate filtering method relies on a simple segmentation procedure that aims at identifying clues from a predefined list of frequently used lexical items. We further discuss in Sect. 5 on evaluation the experimental setting of this predefined list of clues.

With an extensive list of lexical clues, the filtering step based on lexical clues can be effective (see Sect. 5). However, experiments on complex databases showed that this step often ends up with several candidates. Furthermore, attributes that can play a categorization role are not necessarily defined with lexically marked names. In the example of figure 2, the attributes Country and City can be seen in some application contexts as good categorization sources even though no lexical clues can be found in the attribute names. These facts motivate the need for complementary ways to characterize the potentially relevant categorizing attributes. Additional filtering

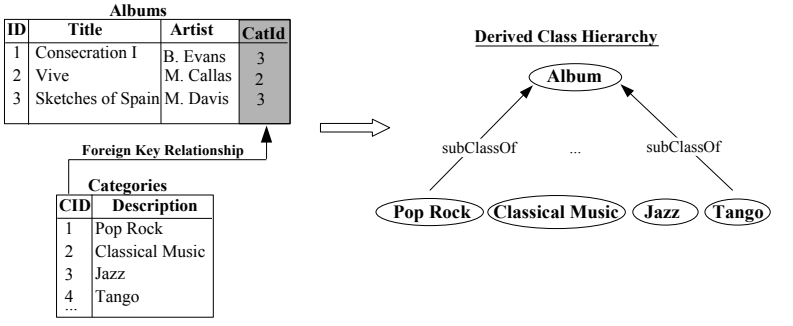



**Fig. 2** An example of a categorization pattern where the categories to be exploited for hierarchy generation are further defined in an external relation

mechanisms can help to make a decision even when no lexical clues can be found or to choose between lexically pre-filtered attributes. Information diversity in the attribute extension appears to be a good complementary source in this selection process.

**- Filtering though entropy-based estimation of data diversity**

We make the assumption that a good candidate for tuple categorization might exhibit some typical degree of diversity that can be formally characterized using the concept of entropy from information theory.

Entropy is a measure of the uncertainty of a data source. In our context, attributes with highly repetitive values will be characterized by a low entropy. Conversely, among attributes of a given relation, the primary key attribute will have the highest entropy since all values in its extension are distinct.

Informally, the rationale behind this selection step is to favor the candidate that would provide *the most balanced distribution of instances within the subclasses*.

We give in what follows a formal definition of this step.

If $A$ is an attribute of a relation schema $R$ instantiated with relation $r$, the diversity in $A$ is estimated by:

$$H(A) = -\sum_{v \in \pi_A(r)} P_A(v) \,.\, \log P_A(v) \tag{1}$$

$$P_A(v) = \frac{|\sigma_{A=v}(r)|}{|r|} \tag{2}$$

- $\pi_A(r)$ is the projection of $r$ on $A$ defined as $\pi_A(r) = \{t[A] \mid t \in r\}$. This set is the *active domain* of $A$. In other words, $\pi_A(r)$ is the set of values attested in the extension of $A$. Each value $v$ of the set $\pi_A(r)$ is a potential category (to be mapped to a subclass in the ontology).

- $\sigma_{A=v}(r)$ is a selection on $r$ defined as $\sigma_{A=v}(r) = \{t \in r \mid t[A] = v\}$. This selection extracts from the relation $r$ the subset of tuples with $A$ attribute equal to $v$. In this specific context, the selection extracts from the relation all entries with (potential) category $v$.
- $P_A(v)$ is the probability of having a tuple with $A$ attribute equal to $v$. This parameter accounts for the weight of $v$ in $A$. It can be estimated by the relative frequency of $v$ (i.e. maximum-likehood estimation).

Let now $C \in R$ denote the subset of preselected attributes using lexical clues. A first pruning operation is applied to rule out candidates with entropy at marginal values:

$$C' = \{ A \in C \mid H(A) \in [\, \alpha,\, H_{max}(R) \,.\, (1-\beta) \,] \} \tag{3}$$

- $H_{max}(R)$ is the highest entropy found among attributes of the relations ($H_{max}(R) = \max_{A \in R} H(A)$)
- $\alpha$ and $\beta$ are parameters such that $\alpha, \beta \in [0,1]$.

As said earlier, $H_{max}(R)$ is often the entropy of the primary key attribute.

If, after this pruning step, several candidates still remain[2], we ultimately select the attribute that would provide the most balanced organization of the instances. This amounts to look for the attribute whose entropy is the closest to the maximum entropy for the number of potential categories involved. This maximum entropy is given by :

$$\tilde{H}_{max}(A) = \log |\pi_A(r)| \tag{4}$$

This reference value, which is derived from the entropy expression (1), is representative of a *perfectly* balanced structure of $|\pi_A(r)|$ categories with the same number of tuples in each category. Note that this value is independent of the total number of tuples ($|r|$).

The final decision aims at selecting the attribute $A^*$ whose entropy is the closest to this reference value:

$$A^* = arg \min_{A \in C'} \delta(A) \tag{5}$$

Where

$$\delta(A) = \frac{|H(A) - \tilde{H}_{max}(A)|}{\tilde{H}_{max}(A)} \tag{6}$$

#### 4.3.2 Generation and population of the subclasses

As shown in first rule of table 2, the generation of subclasses from an identified categorizing attribute can be straightforward. A subclass is derived from each value type of the attribute extension (i.e. for each element of the attribute active domain).

However, proper handling of the categorization source may require more complex mappings. The second rule in table 2 matches a more specific pattern where

[2] Note that all candidates can be eliminated. In this case, the first candidate is arbitrarily chosen.

**Table 2** Complex rules for hierarchy generation based on identification of categorizing attributes ($A = catAtt(r)$). Within the target part of the rule, the variable in bold holds the Uri of the generated fragment in the ontology.

**Categorizing Attribute Values to Subclasses**

| Source | Preconditions | Target |
|---|---|---|
| $r \in d$<br>$A = catAtt(r)$ | $R = sourceOf(C)$ | $\forall v \in \pi_A(r)$<br>class($\boldsymbol{C}_v$ partial $C$) |

**Categorizing Attribute (Indirect) Values to Subclasses**

| Source | Preconditions | Target |
|---|---|---|
| $r \in d$<br>$A = catAtt(r)$<br>$R[A] \subseteq S[pkey(S)]$<br>$pkey(S) = \{B_0\}$<br>$S = \{B_0, B_1\}$<br>$\lvert\pi_{B_0}(r)\rvert = \lvert\pi_{B_1}(r)\rvert$ | $R = sourceOf(C)$ | $\forall v \in \pi_{B_1}(r)$<br>class($\boldsymbol{C}_v$ partial $C$) |

values to be used for subclass generation are issued from another relation. The structuring scheme handled by this rule is encountered in many databases. We give in figure 2 an example where this scheme is applied. In this example, the categorizing attribute CatId in Albums relation is linked through a foreign key relationship to a relation Categories in which all allowed categories are compiled. More suitable names can be assigned by using the values from the second attribute Description of the Categories relation instead of the numerical key values. In addition, a more exhaustive hierarchy can be derived by considering also the categories that have no corresponding tuples in the Albums relation, such as Tango category.

Classes of the resulting hierarchy are populated by exploiting the tuples from the same source relation. An instance is generated from each tuple. The extra task of dispatching the instances into subclasses is based on a partitioning of the tuples according to values of the categorizing attribute.

Formally, for each value $v$ of $A^*$, the corresponding class is populated with the instances derived from the tuples of the set $\sigma_{A^*=v}(r) = \{t \in r \mid t[A] = v\}$.

## 5 Evaluation

To evaluate RTAXON, we gathered a set of 30 databases from different domains. Several technical databases with complex structure were included in this set. We carried out a thorough analysis of these databases to set up a reference list of categorizing attributes. 127 categorizing attributes have been identified from 340 tables. In the selected attributes, the categorisation role is lexically marked through different types of clues.

**Table 3** Performance of the RTAXON selection procedure. In 35% of the decisions made, the second entropy-based filtering step was required to resolve conflicts between several candidates which resulted from the first filtering step based on lexical clues. 78% accuracy was achieved by this second step.

| Precision | Recall | F-Measure | w/ conflict res. | Acc. conflict res. |
|---|---|---|---|---|
| 75% | 72% | 73,5% | 35% | 78% |

As attested by the evaluation results outlined in table 3, good performance is achieved on our representative test set. It is worth noting that this performance level was obtained by carefully fitting the list of lexical clues exploited in the first filtering step. In this context, finding the right balance between precision and recall amounts to find the proportion of word stems (*vs* full clue words) to be included in this clue list. Because of the large proportion of attributes with abbreviated names, good recall cannot be obtained without exploiting clue stems. For example, the stem cat is required to identify the categorizing attributes Catname, CatId and SubcatItemId, all encountered in our test set. The counterpart is that such short clues have a negative impact on the precision. In our evaluation scheme, we only included the stems of the most frequently used clue words. For instance, the clue word Family, which is identified as a relevant but not very frequent indicator, is included in the list without its stem (fam). Consequently, the selection process fails to identify some relevant attributes (e.g. Productfam) of our test set.

In 35% of the attribute selections performed, more than one candidate resulted from the first filtering step based on the lexical clues. The conflicts are solved by invoking the complementary step based on data diversity estimation. 78% accuracy was achieved by this conflict resolution step which has the expected effect of properly excluding the non-relevant attributes. In the definition of the selection procedure (Sect. 4.3), we made the simplifying assumption that only one categorizing attribute can be found in a given table. However, the fact is that several candidates filtered on a lexical basis can be relevant. This is often the case in complex databases where tables may include several categorizing attributes to classify the data from different dimensions. Being able to keep in the end two or even more attributes can lead to better modelling options, especially within an ontology-based representation framework where multiple class hierarchies are allowed. Further investigations are needed in this respect to extend the RTAXON method while keeping the same level of performance.

To better assess the relevance of the entropy-based filtering, we compared maximum entropy, which is taken as the reference value in our approach, with three other basic measures. The selection process was unchanged (i.e. the selection of the candidate whose score is the closest to the reference value). The three basic measures considered were the mean, the minimum and the maximum of the active domain cardinalities over the table attributes. Typically, the aim of the selection based on the mean reference value is to favor attributes with an "intermediate" number of distinct values. In the four settings, the process starts by excluding attributes with

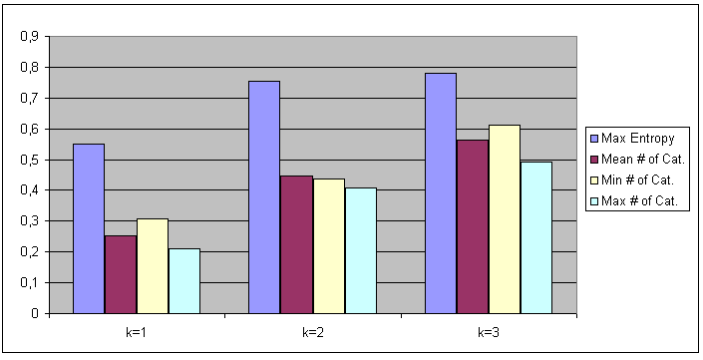



**Fig. 3** Comparison of the maximum entropy value with alternative reference values (mean, minimum and maximum of the active domain cardinalities over the table attributes). The selection is considered successful if a categorizing attribute is in the first $k$ ranked attributes

marginal scores (see expression (3) in section 4.3). This first operation has the effect of ruling out columns with many duplicated values and, conversely, columns with many different values (such as primary keys).

The comparison was performed on a set of 207 attributes obtained by extending the original set of lexically marked categorizing attributes used for the global evaluation. We complemented this set with potentially relevant attributes identified in our database collection but which are not lexically marked. The results of this comparison are given in figure 3. In the defined evaluation scheme, the selection is considered successful if at least one of the categorizing attributes is in the first $k$ ranked attributes. The selection based on maximum entropy clearly outperforms the three others. These figures are strong arguments in favor of a fine characterization of the attribute information diversity to find good structuring patterns. Maximum entropy principles ensures the good ranking of the attributes that would provide balanced distributions of instances in subclasses. However, a side effect that needs to be better controlled is the positive influence on attributes with a larger number of potential classes.

## 6 The RDBToOnto Tool

The RTAXON method is implemented in RDBToOnto[3], see also Cerbah (2008), a highly configurable tool that eases the design and implementation of methods for ontology learning from relational databases. It is also a user oriented tool that supports the complete transitioning process from access to the input databases to generation of populated ontologies. The settings of the learning parameters and control of the process are performed through a full-fledged dedicated interface.

[3] http://www.tao-project.eu/researchanddevelopment/demosanddownloads/RDBToOnto.html

A basic principle in the design of RDBToOnto is to allow the derivation of an exploitable ontology in a fully automated way. By using the tool with its default configuration, a user can get a populated ontology by simply providing as input the Uri of the input database. However, it also allows the user to iteratively refine the result by adding *local constraints*. Several types of constraints are pre-defined while allowing (experienced) users to define new ones. More particularly, local constraints can be included to complement the categorization work performed by RTAXON. To further refine the ontology structure, the user can specify categorization patterns that have been missed by the automated mechanisms (i.e. by selecting relevant categorization attributes through the interface). The generation stage is still fully automated through application of table 2 rules.

An additional benefit of the semi-automated approach is the ability to handle more complex categorization patterns. In this perspective, it is possible to deal with patterns where two categorizing attributes are combined and from which two-level hierarchies can be derived. We discussed a typical illustration of such a pattern in our initial example (Sect. 2 and figure 1). We showed how a geography-based hierarchy of suppliers could be constructed from Country and City attributes in the Suppliers relation. Automated identification of such patterns is not covered by RTAXON. In the current state of the method, these patterns based on two categorizing attributes are signaled by the user through local constraints. However, the automated generation of the two-level hierarchy and its population are supported by the method[4].

Local constraints on instance naming are also highly useful when building fine-tuned ontologies. Instead of letting the system assign arbitrary names to instances, it is possible to specify through local constraints attached to source relations how names should be derived from attribute values (e.g., to an Employees source relation, it is possible to attach a constraint specifying that instance names should be formed by combining values of FirstName and LastName attributes).

A set of reusable components can be directly exploited to implement new learning methods and the user interface can be extended to handle the specific local constraints of the new methods.

Additionally, database readers for some of the most common database formats are included in the tool and new ones can be integrated. The database normalization task described in Sect. 4.2 is also supported by a reusable component.

## 7 Conclusion and Further Work

We presented a novel approach to ontology learning from relational databases that shows how well-structured ontologies can be learned by combining a classical analysis of the database schema with a task specifically dedicated to the identification of

[4] The rules are based on the same principle as those defined in table 2. If $A$ and $B$ are the attributes that stand respectively for first and second levels of the hierarchy, classes of the first level are generated from the set $\pi_A(r)$, classes of the second level from the set $\pi_B(r)$ and the subsumption relation are established with respect to couples of the projection $\pi_{AB}(r)$.

categorization patterns in the data. The formalized method is fully implemented and included in the RDBToOnto platform as the main learning component. The method was validated on a representative set of databases.

A major direction for improvement is the extension of the method to deal with the identification of more complex categorization patterns. We showed that the method has been extended to cover two-level hierarchies. However, the automated part is restricted to the generation of the hierarchies whereas the patterns are supposed to be given as input to the process. Such patterns that are based on several categorizing attributes might be identified by analyzing cooccurrences between values of the involved attributes.

The pattern identification step of the method critically depends on the two exploited information sources (occurrences of lexical clues in attribute names and data diversity in attribute extensions). Diversifying the sources is a way to reduce this dependency. Typically, the use of prior knowledge can help to recognize categorizing attributes. For example, geographical (e.g. country and city names) and temporal data are often used in databases to structure information. Null values as explored in Lammari *et al.* (2007) may also provide a complementary source that could be combined with the sources already exploited in the RTAXON method.

## References

Abiteboul, S., Hull, R., Vianu, V. (eds.): Foundations of databases. John Benjamins, Amsterdam (1995)

Astrova, I.: Reverse Engineering of Relational Databases to Ontologies. In: Bussler, C.J., Davies, J., Fensel, D., Studer, R. (eds.) ESWS 2004. LNCS, vol. 3053, pp. 327–341. Springer, Heidelberg (2004)

Astrova, I., Stantic, B.: Reverse Engineering of Relational Databases to Ontologies: An Approach Based on an Analysis of HTML Forms. In: Workshop on Knowledge Discovery and Ontologies (KDO) at ECML/PKDD 2004, Pisa (2004)

Barrasa, J., Corcho, O., Gómez-Pérez, A.: R2O, an Extensible and Semantically Based Database-to-Ontology Mapping Language. In: Second Workshop on Semantic Web and Databases (SWDB 2004), Toronto, Canada (2004)

Behm, A., Geppert, A., Dittrich, K.R.: On the Migration of Relational Schemas and Data to Object-Oriented Database Systems. In: Györkös, J., Krisper, M., Mayr, H.C. (eds.) Proc. 5th International Conference on Re-Technologies for Information Systems, Oesterreichische Computer Gesellschaft, Klagenfurt, Austria, pp. 13–33 (1997), `http://citeseer.ist.psu.edu/behm97migration.html`

Benslimane, S.M., Benslimane, D., Malki, M., Maamar, Z., Thiran, P., Amghar, Y., Hacid, M.-S.: Ontology development for the Semantic Web: An HTML form-based reverse engineering approach. International Journal of Web Engineering 6(2), 143–164 (2007)

Bizer, C.: D2R MAP - A Database to RDF Mapping Language. In: Proceedings of WWW 2003, Budapest (2003)

Cerbah, F.: Learning highly structured semantic repositories from relational databases – The RDBToOnto tool. In: Bechhofer, S., Hauswirth, M., Hoffmann, J., Koubarakis, M. (eds.) ESWC 2008. LNCS, vol. 5021, pp. 777–781. Springer, Heidelberg (2008)

de Laborda, C.P., Conrad, S.: Relational.OWL: a data and schema representation format based on OWL. In: APCCM 2005: Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling, pp. 89–96. Australian Computer Society, Inc., Darlinghurst (2005)

Lammari, N., Comyn-Wattiau, I., Akoka, J.: Extracting Generalization Hierarchies from Relational Databases. A Reverse Engineering Approach. Data and Knowledge Engineering 63, 568–589 (2007)

Li, M., Du, X., Wang, S.: Learning ontology from relational database. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3410–3415. IEEE, Los Alamitos (2005)

Premerlani, W., Blaha, M.: An approach for reverse engineering of relational databases. Communications of the ACM 37(5) (1994)

Ramanathan, S., Hodges, J.: Extraction of object-oriented structures from existing relational databases. ACM SIGMOD 26(1) (1997)

Stojanovic, L., Stojanovic, N., Volz, R.: Migrating data-intensive Web Sites into the Semantic Web. In: Proceedings of the ACM Symposium on Applied Computing (SAC 2002), Madrid (2002)

Tari, Z., Bukhres, O.A., Stokes, J., Hammoudi, S.: The Reengineering of Relational Databases Based on Key and Data Correlations. In: DS-7, p. 184 (1997)

# Semantic Analysis for the Geospatial Semantic Web

Alina Dia Miron, Jérôme Gensel, and Marlène Villanova-Oliver

**Abstract.** *Semantic analysis* is a new search paradigm for the Semantic Web, which aims the automatic extraction of *semantic associations* existing between individuals described in RDF(S) graphs. In order to infer additional *semantic associations* and to increase the accuracy of the analysis, we propose here, to adapt semantic analysis for OWL-DL ontologies. We also show that by taking into account spatio–temporal information which is usually attached to resources, new and possibly interesting semantic associations can be discovered. Moreover, we propose to handle spatial and temporal contexts in order to limit the scope of the analysis to a given region of space and a given period of time, considered interesting from the user's point of view. For reasoning with spatial and temporal information and relations we use ONTOAST, a spatio–temporal representation and querying system, which is compatible with OWL-DL.

**Keywords:** Geospatial Semantic Web, Semantic Analysis, Spatio-temporal Annotations, OWL.

## 1 Introduction

Nowadays, one of the most popular Web usages is information search (de Kunder, 2008). Current query systems and search engines retrieve relevant Web documents by applying syntactic matching between given keywords and textual content of Web documents. However, with the tremendous amount of digital data available on the Web, problems of data relevance and information overload become acute.

Alina Dia Miron · Jérôme Gensel · Marlène Villanova-Oliver
Laboratoire d'Informatique de Grenoble, 681 rue de la Passerelle,
BP 72, 38402 Saint Martin d'Hères Cedex, France
e-mail: Alina-Dia.Miron@imag.fr, Jerome.Gensel@imag.fr,
Marlene.Villanova-Oliver@imag.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 287–306.
springerlink.com 

The Semantic Web (Berners-Lee *et al.*, 2001) addresses those issues and promises to increase the performances and the relevance of search engines, by annotating the content of any Web resource with machine understandable ontological terms. By adding a descriptive layer to conventional Web pages, the Semantic Web supports the evolution of data towards knowledge and marks the beginning of a new stage in the exploration of Internet. This new stage requires the development of new search paradigms. *Semantic analysis* (Sheth *et al.*, 2002) is one of them. It aims at answering semantic queries like "*Is instance x in any way connected to instance y?*", by retrieving all the paths that connect individual[1] *x* to individual *y* within the considered ontology graphs.

## *1.1 Semantic Analysis*

Different types of real world *objects* are often connected in complex and unexpected ways. There lies the interest of using *semantic analysis* techniques, which offer new perspectives in the discovery of connections between seemingly unrelated *individuals*. This new search paradigm also offers the means to associate a context with each query, in order to handle the user's interests as well as for eliminating the irrelevant results. Semantic analysis has been successfully used in domains such as homeland security, biomedical patents discovery, detection of conflict of interests.

However, so far, semantic analysis has only been defined for RDF(S) graphs and suffers from the limited expressive power of RDF(S). Namely, it only exploits explicit knowledge, as RDF(S) does not provide other axioms beside `subClassOf` and `subPropertyOf`. Also, it is impossible to define in RDF(S) the equivalence and equality relations between resources, so the scope of the *semantic analysis* research is limited considerably.

So far, *semantic analysis* has mainly focused on the thematic dimension of metadata, analyzing, for instance, the collaboration relations existing between two persons, members of an organization. Thus, *semantic analysis* completely ignores the implicit connections resulting from a spatial and temporal proximity between individuals, which reveal to be very important in domains such as homeland security. In this context, the study of the spatial proximity between known members of different terrorist cells at some given instants and/or intervals of time, can result in the identification of hidden collaboration relations.

Moreover, if we study the current developments on the Web, with the growing popularity of spatial Web applications such as *Google Earth, Mappy, ViaMichelin, Geoportail, Virtual Earth 3D*, to name a few, arises the idea of a future Géospatial Web, where resources are geo-tagged or, in other words, annotated using spatial informations. Egenhofer pushes the argument even further and imagines a Geospatial Semantic Web, in which the spatial and temporal annotations are well formalized using ontologies (Egenhofer, 2002), and automatically exploited by agents and query

[1] The terms individual and object are used in this paper as synonyms, for designating an instance of a concept.

engines. In this context, spatial and temporal annotations on Web resources become publicly available and could be exploited by *semantic analysis* for two purposes:

1. the inference of additional *semantic associations* and
2. the rejection of those which are incompatible with some given spatio–temporal context(s).

For instance, given the ontological description of two persons who lived in the same residential area ten years ago, a *spatio-temporal semantic analysis* may deduce that these two persons are very likely to know each other. So, the inferred proximity of their houses can be automatically taken into account for suggesting that there might be a link between those two persons.

## *1.2 Spatial and Temporal Reasoning for the Semantic Web*

While many of the requirements for capturing semantics and expressing ontologies are successfully addressed by the Semantic Web initiative, there is still a fundamental lack when considering the existing standard descriptions (Agarwal, 2005) and reasoning mechanisms (Egenhofer, 2002; O'Dea *et al.*, 2005) dealing with spatial and temporal information. Since geospatial reasoning is mainly based on mathematical computations, logical formalisms upon which rely the ontology languages recommended by W3C (RDF(S), OWL) prove to be unsuited for handling such data. As a consequence, the processing of spatial and temporal information needs to be performed using external formalisms and tools. Once obtained, the results of such computations should be integrated back into the ontological space, becoming available for future reasoning activities.

In (Miron *et al.*, 2007b), we have proposed the use of a system called ONTOAST as an answer to the lack of specialized spatial and temporal inference engines defined on top of OWL or RDF(S) ontologies. ONTOAST is a spatio–temporal ontology modeling and query environment. ONTOAST supports reasoning on spatial, temporal and thematic knowledge and can be used for the Semantic Web thanks to its compatibility with OWL-DL (Miron *et al.*, 2007a).

In this paper, we illustrate the use of ONTOAST for the discovery of *semantic associations* (Anyanwu and Sheth, 2003; Halaschek *et al.*, 2004; Sheth *et al.*, 2002). We propose here a *Semantic Association Discovery Framework* that uses the spatial and temporal reasoning capabilities of ONTOAST for limiting the semantic search space and for inferring new semantic associations. In the first case, the ONTOAST reasoner filters the ontological knowledge with respect to some specified thematic, spatial and temporal query contexts. In the second case, by exploiting spatial information in the description of OWL individuals and object properties, ONTOAST renders explicit some implicit qualitative spatial relations. Those qualitative relations are then used for inferring new and possibly interesting semantic associations, such as "individual *x* lived *veryClose* to individual *y*, for two years", "individual *x* worked *inside* the same building as individual *y*", *etc.* as showed in section 4.5.

We also describe in this paper the spatial ontology GeoRSS-Simple, that we have used for attaching spatial information to Web resources. We have extended this

ontology by a set of qualitative spatial relations. We equally present here the OWL-Time ontology that we use for describing qualitative and quantitative temporal information.

The paper is organized as follows. In section 2 the definition of *semantic associations* in the context of OWL-DL ontologies is presented. Section 3 illustrates the architecture of the semantic framework we propose for semantic association discovery. The semantic association discovery approach as well as the context definitions are discussed in section 4. Section 5 presents some related work and section 6 concludes.

## 2 Semantic Analysis for OWL-DL Ontologies

The concept of *semantic association* was first introduced by (Sheth *et al.*, 2002) for describing the *indirect relations*[2] that link two individuals $x$ and $y$, within a considered RDF(S) graph, $G$. A *semantic association* (also called *ontology-path*) is formally defined in (Anyanwu and Sheth, 2003; Halaschek *et al.*, 2004) as a sequences $x(P_1 \bullet P_2 \bullet \ldots \bullet P_n)y$ of RDF entities so that there exists a set of *objects* $o_1, o_2, \ldots, o_{n-1}$ defined in $G$ which respect the following constraint: $xP_1o_1 \wedge o_1P_2o_2 \wedge \ldots o_{n-1}P_ny$, where $\forall i \in [1,n], P_i$ is an *object property* defined in $G$ whose extension includes the object pairs $(o_{i-1}, o_i), o_0 = x$ and $o_n = y$. For example, in the simple ontology illustrated in Fig. 1, a semantic association can be identified between individuals $pers_3$ and $pers_5$:

$$pers_3\,(studentOf \bullet collaboratesWith)\,pers_5.$$

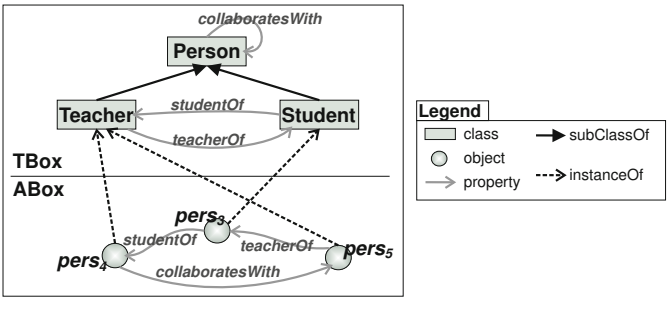



**Fig. 1** A simple RDF(S) graph. The upper part represents the terminological knowledge (TBox) and the lower part represents the assertional knowledge (ABox).

### *2.1 Using OWL-DL Instead of RDF(S)*

The choice of using RDF(S) as an ontology modeling language can be considered as too restrictive considering its limited expressive power when compared to more

[2] That implies a set of intermediary *individuals* and *relations*.

powerful formalisms such as Description Logics or Object Knowledge Representation Languages. Reasons for this are the limited number of axioms supported by RDF(S), only `subClassOf` and `subPropertyOf`, and the absence of axioms to support the definition of RDF(S) graphs alignment. Moreover, given their XML based syntax, RDF(S) graphs are difficult to understand and manipulate by non-expert users. Those observations motivate our work for adapting semantic analysis techniques to OWL-DL ontologies. Our idea is, on the one hand, to obtain more expressive power for defining complex ontologies and, on the other hand, to perform extended inferences that explore, for example, OWL-DL axioms or ontology alignments.

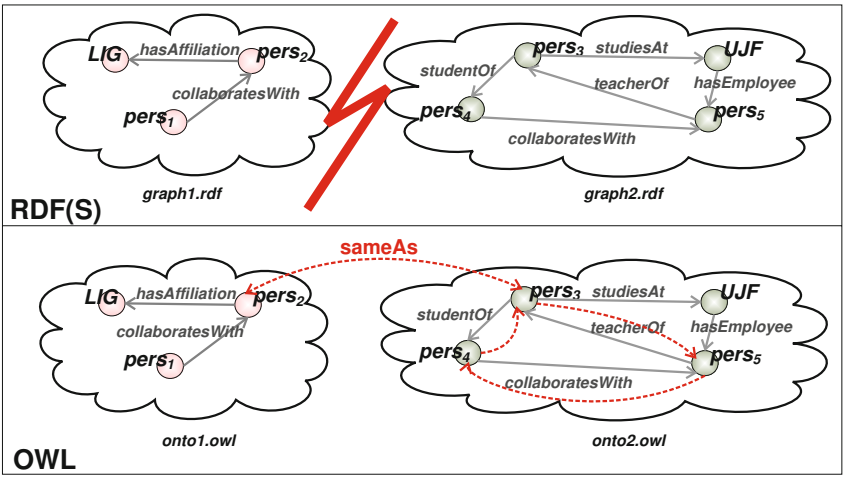



**Fig. 2** Simple examples of ontologies modeled using RDF(S) and OWL-DL

In order to illustrate the advantages of using OWL-DL ontologies instead of RDF(S) graphs as a basis for *semantic analysis*, let us study the situation illustrated in Fig. 2, where $pers_2$ in `graph1.rdf` is indeed the same individual as $pers_3$ in `graph2.rdf`, and the query for *semantic associations* linking individuals $pers_1$ and $pers_4$. In RDF(S), one cannot define two resources as being equal, so the two graphs `graph1.rdf` and `graph2.rdf` remain disconnected. In this situation, no association can be discovered between $pers_1$ and $pers_4$. In the second case, when the same information is modeled in OWL-DL, one can define an equality relation between $pers_2$ and $pers_3$. This creates a bridge between ontologies `onto1.owl` and `onto2.owl`. In this case, at least one semantic association is found between $pers_1$ and $pers_4$ :

$$pers_1\,(collaboratesWith \bullet studentOf)\,pers_4.$$

Moreover, in OWL-DL one can also define object properties as being inverse, functional, inverse functional or symmetrical. For instance, by defining

studentOf and teacherOf as being inverse properties, the implicite relations $\texttt{teacherOf}(\texttt{pers}_4, \texttt{pers}_3)$ and $\texttt{studentOf}(\texttt{pers}_3, \texttt{pers}_5)$ can be inferred using a reasoner such as Pellet or RacerPro. If we also define the collaboratesWith relation as being symmetrical, two new semantic associations can be discovered between $pers_1$ and $pers_4$:

$pers_1\,(collaboratesWith \bullet studiesAt \bullet hasEmployee \bullet collaboratesWith)\,pers_4$,
$pers_1\,(collaboratesWith \bullet studentOf \bullet collaboratesWith)\,pers_4$.

## 2.2 A Formal Definition of Semantic Associations

### 2.2.1 OWL Vocabulary

We consider an OWL vocabulary $V$, formally defined by (Patel-Schneider *et al.*, 2004) as consisting of a set of literals $V_L$ and seven sets of URI references: $V_C$, $V_D$, $V_I$, $V_{DP}$, $V_{IP}$, $V_{AP}$, and $V_O$, where $V_C$ is the set of class names and $V_D$ is the set of datatype names of a vocabulary. $V_{AP}$ represents the annotation property names, $V_{IP}$, the individual-valued property names, $V_{DP}$, the data-valued property names, $V_I$ the individual names, and $V_O$ the ontology names of a vocabulary. The same source (Patel-Schneider *et al.*, 2004) defines an OWL interpretation as a tuple of the form: $I = < R, EC, ER, L, S, LV >$, where $R$ is the set of resources of $I$, $LV$ represents the literal values of $R$, and $L$ is an interpretation that provides meaning to typed literals. The mapping $S$ provides meaning for URI references that are used to denote OWL individuals, and helps provide meaning for annotations. $EC$ provides meaning for URI references that are used as OWL classes and datatypes while $ER$ provides meaning for URI references that are used as OWL properties.

We also consider $O$ ($O \subseteq R$, $O \bigcap LV = \emptyset$ ) as being a non empty set of class instances (objects), the set of object property instances $\Gamma$ ($\Gamma \subseteq 2^{O \times O}$), and the set $\Lambda$ containing the datatype property instances ( $\Lambda \subseteq 2^{O \times LV}$). Considering the fact that we search for links between individuals in an OWL concrete model, we are only interested in a subset of the mappings $EC$ and $ER$ defined for the interpretation model $I$. Thus, we define the function $Ext_C$ ($ExtC : V_C \rightarrow 2^O$) as being a specialization of the $EC$ mapping for providing meaning for URI references that are used as OWL *classes* ($Ext_C(Cl) \subseteq 2^O, Cl \in V_C$). We also consider $Ext_R : V_{IP} \rightarrow \Gamma$, ($Ext_R(Rel) \subseteq \Gamma \subseteq 2^{O \times O}, Rel \in V_{IP}$) and $Ext_D : V_{DP} \rightarrow \Lambda$ , ($Ext_D(Att) \subseteq \Lambda, Att \in V_{DP}$), as two sub mappings of $ER$ which provide meaning for URI references that are used as OWL *object properties* respectively OWL *datatype properties*.

### 2.2.2 Ontology Graph

For any OWL-DL ontology, $\Omega$, it is possible to build an oriented graph $G$, as defined by equation 1, whose set of vertices, $V_G$, contains the individuals and literal values defined in the declarative part of $\Omega$ ($ABox$).

$$G = (V_G, E_G), where\ V_G = O \cup LV\ and\ E_G = \Gamma \cup \Lambda,$$
$$E_G \subseteq \{(i,j) | i \in O, j \in O \cup LV, \exists Rel \in V_{IP} : (S(i), S(j)) \in Ext_R(Rel) \vee \exists Att \in V_{DP} : (S(i), S(j)) \in Ext_D(Att)\} \quad (1)$$

The set of directed edges, $E_G$, corresponds to the *object properties* (or *tuples*[3]), introduced in the declarative part of $\Omega$ (*ABox*) or inferred using the property axioms defined in the terminological part of $\Omega$ (*TBox*).

### 2.2.3 Semantic Associations - Definition

In this context, we say that two individuals $x$ and $y$ $(x, y \in V_G)$ are *semantically associated* with respect to the graph $G$, if the latter contains at least one path that starts with the vertex $x$, passes through a series of intermediate objects $(o_i \in V_G, i \in [1, n-1])$ connected by *tuples* $(e_i \in E_G, i \in [1,n])$ and reaches the vertex $y$. For describing the *ontology-paths* in an unambiguous way, in this paper, we use the following notation:

$$ontology - path(x,y) = x \xrightarrow{e_1} o_1 \xrightarrow{e_2} \ldots \xrightarrow{e_{n-1}} o_{n-1} \xrightarrow{e_n} y,$$
$$o_i \in V_G, 0 \leq i \leq n, x = o_0, y = o_n, e_j \in E_G, 1 \leq j \leq n. \quad (2)$$

Our work focuses on inference techniques that are able to deduce temporal and spatial relationships between *objects* described in an ontology graph $G$, and that can lead to the discovery of new *ontology-paths*. To this end, we propose the use of a semantic association discovery framework whose architecture is presented in the next section.

## 3 Geospatial and Temporal Semantic Analysis Framework

Our proposal is based on the use of ONTOAST (which stands for ONTOlogies in AROM-ST[4]) as a spatio-temporal ontology modeling and semantic query environment. ONTOAST (Miron *et al.*, 2007b) is an OWL-DL compatible extension of the Object Based Representation System AROM (Page *et al.*, 2001), a generic tool designed for knowledge modeling and inference. The originality of this system stands in its powerful and extensible typing system and in its Algebraic Modeling Language (Moisuc *et al.*, 2005) used for expressing operational knowledge in a declarative way. This language allows one to specify the value of a variable using numerical and symbolic equations involving various elements of the knowledge base. ONTOAST is built upon the spatio-temporal module AROM-ST. The interest of using ONTOAST lies in its predefined set of qualitative spatial and temporal associations (presented in details in (Miron *et al.*, 2007b)). Those *associations* can be

[3] A *tuple* is an instance of an *object property*.

[4] AROM-ST is a spatio-temporal extension of the AROM Object Knowledge Representation System.

used in order to complete data on the modeled *objects* as well as to allow for a more flexible query formulation.

Thus, ONTOAST ontologies are flexible enough to handle the coexistence of, on the one hand, quantitative spatial and temporal data in the form of exact geometries and time intervals or instants and, on the other hand, imprecise data in the form of qualitative spatial and temporal relations. These two kinds of information complement one another and offer advanced reasoning capabilities. ONTOAST takes into account three categories of qualitative spatial relations: *topology, orientation* and *distance*. They can be automatically inferred from existing knowledge when they are needed, or explicitly defined by users. In order to perform similar inferences on temporal data, ONTOAST manages a set of qualitative temporal relations *(before, after, starts/started-by, finishes/finished-by, during/contains, equals, meets/met-by, overlaps/overlapped-by)*.

This paper presents the use of ONTOAST in the semantic association discovery process. Fig. 3 illustrates the Semantic Analysis Framework that we propose. It contains five main modules in charge respectively of the Knowledge Acquisition, the Query Interface, the Ontology-path Discovery, the Result Classification and the Result Visualization. They are organized as distinct modules built on top of the ONTOAST System. In order to use the reasoning facilities provided by ONTOAST, ontological knowledge has to be translated into the AROM formalism and stored into a local object-oriented Knowledge Repository (see step 1 in Fig. 3).

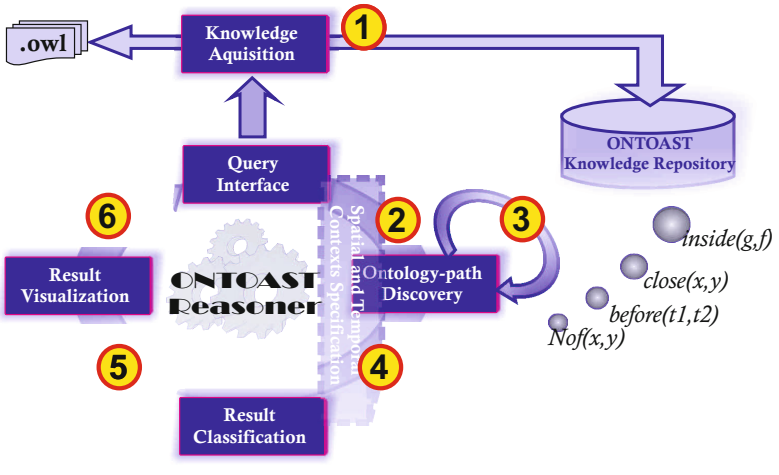



**Fig. 3** Overview of the ONTOAST Framework for the discovery of semantic associations

Once ontological knowledge is imported into the ONTOAST Knowledge Repository, spatial and temporal semantic inferences can be activated. During the *ontology-path* discovery process, Knowledge Repository data are filtered using the Spatial and Temporal Contexts Specification, in order to reduce the search space (step 2 in

Fig. 3). Filtered ontological knowledge is exploited by the Ontology-path Discovery module which searches for the semantic associations between two individuals (step 3 in Fig. 3). The Spatial and Temporal ONTOAST Reasoner is used both in the filtering phase (for inferring spatial characteristics and temporal validity for individuals) and in the Ontology-path Discovery phase (for inferring spatial connections between individuals). The obtained *ontology-paths* are then transfered to the Result Classification module (step 4 in Fig. 3) which calculates their relevance using the context specifications (step 5 in Fig. 3). Finally the Result Visualisation module presents the results to the user (step 6 in Fig. 3).

Obviously the handling of spatial and temporal information increases the scope of the semantic analysis, but raises at the same time new representation and reasoning challenges. For instance, the limited typing system adopted by RDF(S) and OWL does not offer support for spatial extensions. As a consequence, in order to model spatial data in OWL, dedicated concepts which simulate spatial datatypes have to be used. For example, a polygon (instance of a *Polygon* class) will be represented by a list of *Points* $(x,y)$ objects connected to a given Coordinate System. Nevertheless, this solution has several disadvantages. First of all it is not very easy to use by non-expert users as the handling of complex ontological concepts and properties is required. Second of all, at present, there is no standard spatial ontology for describing spatial information. Thus the automatic reconciliation of spatial descriptions defined with respect to distinct spatial ontologies can be extremely difficult to realize. Moreover, the exploitation of spatial informations in the context of the Semantic Web is very difficult due to the current absence of spatial reasoners capable of handling RDF(S) graphs and/or OWL ontologies.

## *3.1 Spatial Information*

Several geospatial ontologies which model geometric features have been proposed up to now. An assessment study which compares 45 geospatial and temporal ontologies relevant to geospatial intelligence, from the annotation, qualitative reasoning and information integration points of view, has recently been published (Ressler and Dean, 2007). Following the recommendation of the authors, we have chosen the GeoRSS-Simple[5] OWL encoding as a reference spatial ontology. GeoRSS-Simple is a very lightweight format for spatial data that developers and users can quickly and easily integrate into their applications and use for expressing semantic annotations with little effort. It supports basic geometries (*point, line, box, polygon...*) and covers the typical use cases when encoding locations.

The geospatial descriptions are centered on two spatial concepts: *_Geometry* and *_Feature*. The *_Geometry* class, together with its five specializations (see Fig.4) — *Envelope, LinearRing, LineString, Point* and *Polygon*— can be used for modeling spatial locations as individuals. This can reveal useful when exact geometrical information is not available for a spatial object, say *l*, which can instead be spatially characterized using the spatial relations that hold between *l* and other spatial objects

[5] http://georss.org/simple

in the ontology. One can, for instance, simply define a spatial location *l*, as being a *Polygon* instance, without specifying its border. Then, for situating *l* in space, one can link it to an address *a*, by a relation like the *NOf* direction relation. When using the *_Feature* concept, one has a simpler model, which describes spatial information by means of well formed string or double values, designated using the data properties *line, point, pos, etc.* (see Fig. 4). The ONTOAST spatio-temporal reasoner is designed to recognize spatial data types modeled using the ontological concepts defined in the GeoRSS-Simple ontology.

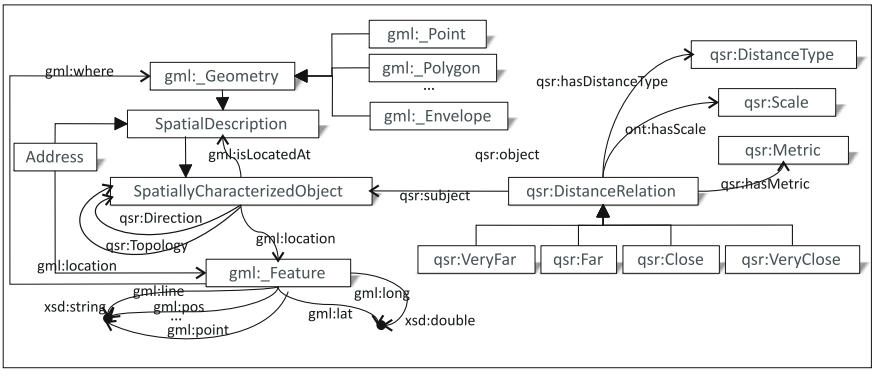



**Fig. 4** General OWL ontology defining Spatial Objects. It integrates the QualitativeSpatialRelations ontology (ont prefix), a reference ontology we have chosen for modeling qualitative spatial relations compatible with ONTOAST, and the GeoRSS ontology (gml prefix), the reference ontology for modeling geometric features.

In order to represent in OWL qualitative spatial relations that can be handled by the ONTOAST reasoner, we propose the use of a *QualitativeSpatialRelations* ontology. This ontology contains two generic types of spatial relations, *Direction* and *Topology*, modeled using object properties. The *Direction* relation has nine specializations which allow the expression of cardinality directions (*Nof, Sof, Eof, Wof, CenterOf, NEof, NWof, SEof, SWof*) existing between spatial objects. Specializations of the *Topology* relation have also been defined: *Disjoint, Contains, Crosses, Touches, Equals, Intersects, Overlaps* and *Within*.

*Distance* relations are more difficult to represent since they come with attributes specifying, for example, the metric system employed when calculating the distance (Euclidian distance, shortest road, drive time, etc.), the scale, or the distance type when regions of space are described (average border points, gravity centers, administrative centers, etc.). Since OWL-DL does not allow object properties to have some attributes, distances between spatial objects are reified as objects of the *DistanceRelation* (see Fig.4 ). Four specializations (*VeryFar, Far, Close*, and *VeryClose*) of the *DistanceRelation* allow the specification of absolute distances between concepts. Fig.4 illustrates the general OWL ontology we have used for annotating spatial

objects. With this ontology, spatial information can be attached to *SpatiallyCharacterizedObject*'s instances in four ways: a) using the *isLocatedAt* object property, which designates a concrete geometry (instance of *_Geometry*) for the specified geographic object, b) using the *isLocatedAt* object property that refers to an *Address*, c) using the *location* object property that designates a concrete spatial feature (instance of *_Feature*) and d) through the use of a qualitative spatial relation with another *SpatiallyCharacterizedObject*.

## *3.2 Temporal Information*

While the Web Ontology Language offers no data type support for modeling spatial information, things are different for temporal information. OWL offers several dedicated data types: *xsd* : *dateTime*, *xsd* : *date*, *xsd* : *gYearMonth*, *xsd* : *gMonthDay*, *xsd* : *gDay*, *xsd* : *gMonth*, that can be used for associating temporal characteristics with individuals. However, complex temporal configurations cannot be expressed by exclusively using those time data types. For example, if one is interested in expressing that the exact temporal extent of the event $e_1$, is unknown and that $e_1$ happened before a certain *instant*, one has to use temporal concepts which model temporal *instants* or/and *intervals* and adapted temporal relations defined in a reference temporal ontology.

Standardization efforts have been made so far (Hobbs and Pan, 2005, 2006), which resulted in the definition of an expressive ontology of time: OWL-time. In Fig.5, we present the fragment of the OWL-time ontology used in this work. The ontology considers two types of temporal entities: *instants* and *intervals*. *Instants* represent punctual moments of time. One *instant* can be defined using i) the *inDateTime* object property which specifies its concrete date-time description, ii) the *inXSDDateTime* data property, or iii) one or several qualitative temporal relations (*after, before, hasEnd, hasBeginning*) the *instant* satisfies with respect to another *instant* and/or *interval*. *Intervals* are defined by specifying the earliest and the latest *instants* they include (through the *hasBeginning*, *hasEnd* object properties). *Proper intervals* are special *intervals* whose beginning and end *instants* are different.

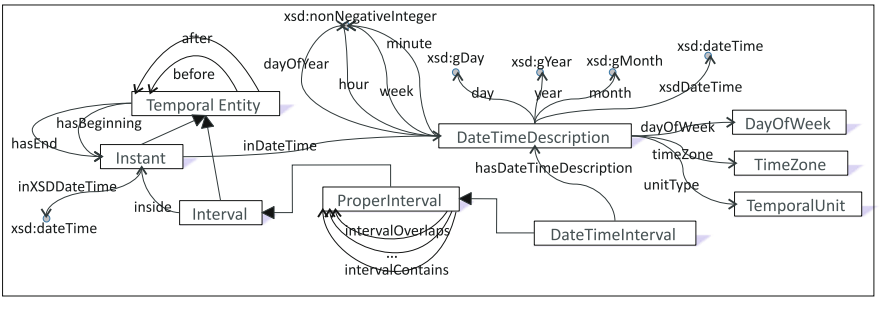



**Fig. 5** Graphical representation of a fragment of the OWL-time ontology

Thirteen qualitative temporal relations are defined between *proper intervals*, inspired by Allen's temporal calculus (Allen, 1983): *intervalEquals, intervalBefore, intervalMeets, intervalOverlaps, intervalStarts, intervalDuring, intervalFinishes, intervalAfter, intervalMetBy, intervalOverlappedBy, intervalStartedBy, intervalContains, intervalFinishedBy*. We consider that temporal characteristics can be attached to temporal objects through the use of the *holds* object property which has two specializations: *atTime*, and *during*, as illustrated in Fig. 6. We have added a generic object property, *qualTempRel*, to our reference temporal ontology, which is a generalization of the thirteen qualitative temporal relations defined between proper intervals in the OWL-time ontology.

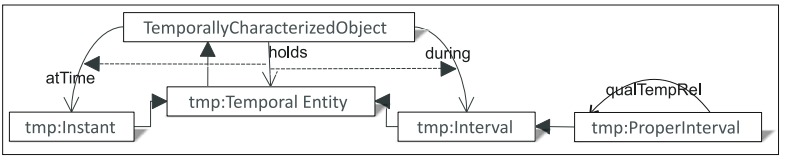



**Fig. 6** Reference ontology defining *Temporally Characterized Objects*. It specifies utility relations for attaching temporal validity to objects.

Nonetheless, one cannot use the temporal datatypes, nor the temporal concepts defined by the OWL-time ontology for expressing object property validity within the decidable frontiers of OWL-DL. This is because, in OWL-DL, the domain and range of data properties and object properties cannot contain object property based expressions. An interesting approach is presented in (Gutierrez *et al.*, 2005), and consists in stamping RDF triples with instants or intervals of time. Since current OWL recommendations do not support temporal stamps, we propose to simulate them through special annotations handled by the ONTOAST parser. Three examples of temporal annotations are shown in Fig.7. The first two assign temporal intervals to tuples (instances of object properties) and the last one stamps the corresponding tuple with a temporal instant.

```
ObjectPropertyAssertion(Comment("temporalValidity(2005-10-10,T10:30:00 -)")  knows7 pers1 pers2)
ObjectPropertyAssertion(Comment("temporalValidity(2003-01-10,2006-30-06)")  studiedAt  pers1 UJF)
ObjectPropertyAssertion(Comment("temporalValidity(2008-08-10)")  colaboratesWith pers1 pers2)
```

**Fig. 7** Examples of temporal annotations on OWL DL tuples, described using the Functional-Style Syntax defined for OWL 2.

# 4 Query Approach

Searching for *semantic associations* between two individuals can be achieved by applying classical algorithms of graph path discovery between two given vertices. In the open Geospatial Semantic Web environment, those algorithms can return a high number of results. It is therefore necessary to establish some filter criteria that limit the search space according to the user's interests. In the following sections three such filters are presented.

In order to illustrate our approach for the discovery of geospatial and temporal *semantic association*, in the reminder of this section, we refer to an ontology describing collaboration relationships existing between researchers (Fig.9). Let us consider a query that searches for the existing links between two persons, $pers_2$ and $pers_4$.

## *4.1 Semantic Association Discovery Approach*

The Ontology-path Discovery module of the Semantic Analysis Framework presented in section 3 integrates the depth first discovery algorithm presented in Fig. 21. Ontology-paths between two given individuals *x* and *y* are inferred using a progress stack (called *stack*) and a result stack, *pPath*, both containing intermediate tuples. They are initialized with a virtual tuple, having as objects the start individual (see lines 2-3). At each step of the algorithm, if advance in the graph if possible, all the tuples having as subject the current individual (*source*) and which satisfy the context specifications are added to the stack (see lines 5-9). If, on the contrary, there is no tuple having as subject the current source, the last considered tuple is eliminated from the *pPath* and from the work stack as well. When adding an intermediate tuple to the *pPath*, the absence of cycles and the length constraint (*LMax*) are checked (line 15). If the current tuple *t*, blocks the construction of a valid ontology-path, *t* is eliminated (lines 24-28) both from the *stack* and from the *pPath*. The algorithm is executed as long as the *stack* contains possible path alternatives (line 29). In order

**pPathDiscovery(start, stop, Lmax)**

```
1.source=start
2.push(pPath, create Tuple (null, null, source))
3.push(stack, create Tuple (null, null, source))
4.do
5.  if hasTuples(source)
6.    if getSubject(top(stack))=source
7.      for each tpl in tuples(source)
8.        if inCtx(tpl)
9.          push(stack, tpl)
10.   else
11.    pop(pPath)
12.    pop(stack)
13.    source=getObject(top(stack))
14.t=top(stack)
15.if size(pPath) Lmax and not contains(pPath,t)
 and not cycle(pPath,getObject(t))
16.  push(pPath, t)
17.  if getObject(t)=stop
18.    add(resultSet, pPath)
19.    pop(pPath)
20.    pop(stack)
21.    source=getSubject(top(stack))
22.  else source=getObject(t)
23.else
24.  if contains(pPath,t)
25.    pop(pPath)
26.  if size(stack) >1
27.    pop(stack)
28.    source= getSubject(top(stack))
29.while size(stack)>1
30.return resultSet
```

**Fig. 8** The ontology-path discovery algorithm

to add a tuple $t$ to the current ontology-path, its elements must satisfy the Temporal and Spatial Context specifications, defined in sections 4.3 and subsec:4.2. Those tests are performed by the *inCtx* function (see line 9). The set of inferred ontology-paths is kept by the variable *resultSet* which is a list of lists (see lines 18, 30).

In the worst case scenario, when all objects in $G$ are connected to each other by a maximum of $d_{max}$ direct tuples, the maximum complexity of this algorithm is $O(d_{max}^{L_{max+1}})$ . This case is highly improbable since the instances of object properties described in ontologies rarely hold between all pairs of objects. Moreover, the use of contexts considerably limits the computations. Nevertheless, we intend to optimize this first version of algorithm, following the approach presented in (Tarjan, 1981), which provides promising fast algorithms for solving path problems.

In the case when no temporal and spatial filters are applied, the algorithm exhaustively discovers all paths linking $pers_2$ to $pers_4$ in the ontology of Fig.9. Results obtained in this case are illustrated in table 1. These results can be further refined using the contextual information attached to the query, as shown in the following sections.

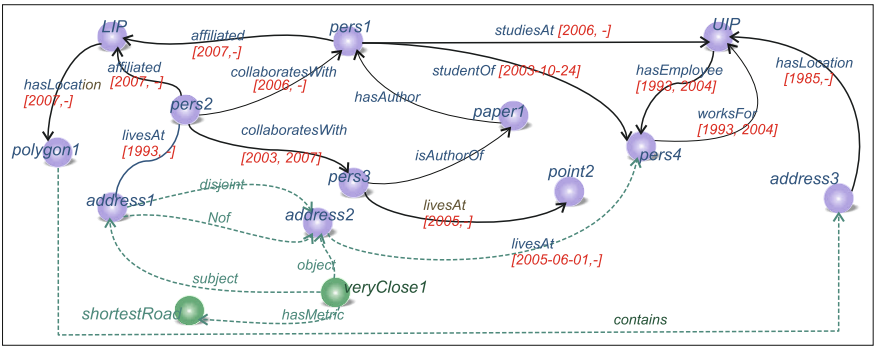



**Fig. 9** Graphical representation of an OWL *ABox* describing a small social network in the research domain

**Table 1** *Ontology-paths* linking $pers_2$ and $pers_4$ in the ontology illustrated by Fig.9

| |
|---|
| $p_1 = pers_2 \overset{collaboratesWith}{\rightarrow} pers_1 \overset{studentOf}{\rightarrow} pers_4$ |
| $p_2 = pers_2 \overset{collaboratesWith}{\rightarrow} pers_1 \overset{studiesAt}{\rightarrow} UIP \overset{hasEmployee}{\rightarrow} pers_4$ |
| $p_3 = pers_2 \overset{collaboratesWith}{\rightarrow} pers_3 \overset{isAuthorOf}{\rightarrow} paper_1 \overset{hasAuthor}{\rightarrow} pers_1 \overset{studentOf}{\rightarrow} pers_4$ |
| $p_4 = pers_2 \overset{collaboratesWith}{\rightarrow} pers_3 \overset{isAuthorOf}{\rightarrow} paper_1 \overset{hasAuthor}{\rightarrow} pers_1 \overset{studiesAt}{\rightarrow} UIP \overset{hasEmployee}{\rightarrow} pers_4$ |

## 4.2 Thematic Context

The thematic context for a semantic query captures the user's thematic interest in order to exclusively present her/him the semantic associations which may be relevant

from her/his point of view. Concretely, within the target ontologies, one or more regions of interest $Ct_1, Ct_2, \ldots, Ct_l$ can be defined, in the form of sets of classes and object properties not necessarily disjoint. Once a class/object property is included into a region of interest $Ct_i$, all of its subclasses/sub properties are also considered as being part of the context $Ct_i$. The user can assign different weights to those regions $(w_1, \ldots w_l)$, for quantifying their importance. The weights are then used to calculate the rank of the discovered semantic associations. The thematic context also assumes the definition of a context parameter: *the context deviation*. It represents an integer value which specifies the maximum number of successive elements (individuals and object properties instances) included by an ontology path that can be outside of all regions of interest. In other words, a *context deviation* equal to 3 tell us that the valid ontology paths cannot contain more than 3 successive elements which are not included in any region of interest. The thematic context can be seen as a filter that limits the discovery of semantic associations to the regions of interest as well as to their immediate neighborhood.

For instance, in the ontology illustrated in Fig.9, if the user is exclusively interested in direct *collaboration* and *student/teacher* relations between *persons*, then he/she defines the thematic context $Ct_1$={*Person, collaboratesWith, studentOf*}. Assuming that the user wants the resulting *ontology paths* to remain in this thematic region of the ontology, then the *context deviation* should be a small number, let us say 2. In this situation the *ontology paths* $p_3$ and $p_4$ are eliminated from the result list, because they contain three successive elements $isAuthorOf$, $paper_1$ and *hasAuthor*, which are not part of any context and thus considered as irrelevant for this query.

## 4.3 Temporal Context

The first type of context we consider for a query is the temporal context $C_{Temp}$. It acts as a filter and narrows down the search area to *ontology-paths* that meet a certain qualitative relation with a given temporal interval or instant. The temporal context is specified using an interval of the type $(start\_date, end\_date)$ or $(date, duration)$, together with a topological relation. The implicit topological relation is the inclusion, but all the topological temporal AML operators defined in (Miron *et al.*, 2007b) can be used: *before/after, starts/started-by, finishes/finished-by, during/contains, equals, meets/met-by, overlaps/overlapped-by*. When a time interval is specified through a pair $(date, duration)$, the temporal attributes describing tuples or objects are compared to the interval $[date, date + duration]$. The implicit temporal interval starts with the beginning of time and holds up to the present moment, but it can be adapted to the user's preferences.

When analyzing an instance $e$ (tuple or object) candidate for its integration into an *ontology-path*, the system compares its validity, given by its temporal data, specified using the ontology illustrated in Fig.7 (for individuals) and using annotation stamps (for *tuples*), to the temporal context $C_{Temp}$. If the entity has no temporal attributes, the system considers that it has always existed and that it is eternal.

On the example shown in Fig.9, let us consider that the user is only interested in collaboration relations valid during the period $[2005, 2008]$. He/she will thus specify the temporal context of the query as being $C_{Temp} = ([2005, 2008], overlaps)$. This will eliminate from the resulting *ontology-paths* $p_3$ and $p_4$ (see Fig. 9). The reason is the validity of the *tuple* $(collaboratesWith\ pers_2\ pers_3)$ given by its temporal stamp: $[2003, 2004]$. The interval $[2003, 2004]$ does not satisfy the *overlaps* relation with the $C_{Temp}$ $interval [2005, 2008]$, so the *ontology-paths* construction cannot be completed.

## 4.4 Spatial Context

The ONTOAST spatial reasoner can be used to filter and eliminate from the set of results the *ontology-paths* which contain intermediate objects or/and tuples that do not satisfy a given qualitative spatial relation with a specified region of the geographical space. We define the spatial query context $C_{Spatial}$, as the pair *(spatial object, qualitativeSpatialRelation)*, where *qualitativeSpatialRelation* is one of the spatial relations described in section 3 (see Fig. 4). Similarly to the case of the temporal context, during the analysis of an instance $e$ candidate for integration into a *ontology-path* $p$, the system compares its spatial characteristics given by its geometry or inferred by the system (in the case of objects), to the spatial context $C_{Spatial}$. If $e$ is not consistent with $C_{Spatial}$, it will not be taken into account as a possible intermediate object in a *ontology-path*. The instances which have no associated spatial extent are considered as being consistent with the spatial context $C_{Spatial}$.

Let us consider that, in the example illustrated in Fig.9, $address_1$ and $address_2$ describe two places in Paris at one street away from each other, $address_3$ is located in Versailles, $polygon_1$ describes the geometric limits of all the sites administrated by the Computer Science Laboratory from Paris and $point_2$ represents geospatial coordinates in England. Now, if the user is only interested in connections between researchers from the French Region Ile de France he/she imposes as a spatial filter for the semantic query $C_{Spatial} = (IleDeFrance, contains)$. IleDeFrance must be a spatial object defined within the accessible ontologies. $Pers_3$, whose current location (England) does not satisfy the inclusion relation becomes inconsistent with the specified context. In consequence, the construction of the ontology-paths passing by $pers_3$ ($p_3$ and $p_4$ from Table 1) cannot be completed.

## 4.5 Spatial Inference for Ontology-Path

The ONTOAST spatial reasoner can also be used by an extended *ontology-path* discovery algorithm which aims at identifying individuals who can be linked through a relevant spatial relation that stands between them. The algorithm has the following steps:

- If $x$ is not a spatial object or if its geometry is unknown, the system builds the set $S_x$ containing the spatial objects that are related to $x$ by *ontology-paths* $\alpha_1$, $\alpha_2, \ldots, \alpha_z$ of a given maximum length $l_{Max}$:$|\alpha_i| \leq l_{Max}, \forall i \in [1, z]$ and which are

consistent with the specified contexts. For a spatial object $x$ with known geometry the set $S_x$ exclusively contains the object itself.
- The same steps are taken for building the set $S_y$.
- Among the objects contained by the sets $S_x \setminus S_y$ and $S_y$, the system exhaustively infers the existence of qualitative spatial relations. The qualitative relations inferred this way are added up to the ontology and the newly obtained *ontology-paths*: $\alpha_i \overset{spatial\_relation}{\rightarrow} \beta_j$ will be taken into account as a result of the semantic analysis.

Using this algorithm, new *ontology-paths* can be discovered. For example, in the ontology of Fig. 9, after building the sets $S_{pers_2} = \{address_1, polygon_1\}$ and $S_{pers_4} = \{address_2, address_3\}$ the ONTOAST spatial reasoner checks which qualitative spatial relations exist between pairs: $address_1$ and $address_2$, $address_1$ and $address_3$, $polygon_1$ and $address_2$, $polygon_1$ and $address_3$. Let us focus on the possible inferences from the first pair. Several geocoding Web Services exist nowadays (Yahoo!Maps, Google Maps, MapPoint, ...), that support address transformation into corresponding latitude and longitude coordinates. So, obtaining geographic positions from address specifications is relatively easy. The obtained quantitative spatial information will be used by the ONTOAST spatial reasoner for inferring, through geometric computations, the spatial relations existing between addresses. Knowing that $address_1$ is in a close proximity of and at North from $address_2$, the tuples $veryClose(address_1, address_2)$, $Nof(address_1, address2)$ and $disjoint(address_1, address_2)$ will be added up to the ontology, which facilitates then the discovery of three new *ontology-paths* as illustrated in table 2.

**Table 2** *Ontology-paths* linking $pers_2$ and $pers_4$ discovered using inferred spatial relations

| |
|---|
| $p_5 = pers_2 \overset{currentLocation}{\rightarrow} address_1 \overset{object}{\rightarrow} veryClose_1 \overset{subject}{\rightarrow} address_2 \overset{currentLocation}{\rightarrow} pers_4$ |
| $p_6 = pers_2 \overset{currentLocation}{\rightarrow} address_1 \overset{Nof}{\rightarrow} address_2 \overset{currentLocation}{\rightarrow} pers_4$ |
| $p_7 = pers_2 \overset{currentLocation}{\rightarrow} address_1 \overset{disjoint}{\rightarrow} address_2 \overset{currentLocation}{\rightarrow} pers_4$ |

## 5 Related Work

Besides the related work previously cited, (Sheth *et al.*, 2002; Anyanwu and Sheth, 2003; Halaschek *et al.*, 2004), another interesting approach is presented in (Perry and Sheth, 2008). The authors propose the integration of RDF ontologies and ORACLE databases, through the use of ORACLE Semantic Data Store. The latter provides the ability to store, infer and query semantic data in the form of simple RDF descriptions or RDFS ontologies. Spatial attributes attached to ontological objects, modeled in RDF using coordinates lists (for example, instances of the *Point* class), are translated into values of a spatial datatype in ORACLE (*SDO_GEOMETRY*).

Those translations are very useful as all the ORACLE Spatial query facilities, including its topological and proximity operators, are available. However, this approach does not make it possible to combine qualitative and quantitative reasoning.

Another related approach, proposed by (Balmin *et al.*, 2008), aims at the discovery of interesting relationships between objects described in XML documents. It is based on the use of SEDA, an interactive search tool that exploits the structures of the target XML trees as well as the textual content of XML elements. A query in SEDA consists of a set of *query terms* specified in the form of *(context, search_query)* pairs. The *context* represents a node name or a root-to-leaf path while the *search_query* is any full-text search expression comprising a simple bag of keywords, a phrase query or a boolean combination of those. The result of a SEDA query is a set of tuples, where each tuple represents a connected sub-graph with *m* nodes, one for each *query term*. The main disadvantage of this approach comes from its lack of semantics as it exclusively exploits structural information and textual descriptions. The resulting tuples that link the specified *query terms* are built using the inclusion relation (which has no specified semantics and whose meaning can differ from one level of the XML tree to another) defined between XML elements, and are dependent exclusively on the structure of the analyzed document. This approach makes no distinction between objects and relations so the resulting tuples are not chains of composed relations, as presented in this article, but chains of heterogeneous XML elements.

Another framework built for link discovery within RDF(S) and OWL ontologies is Silk (Volz *et al.*, 2009). Silk features a declarative language for specifying which types of links should be discovered between data sources as well as which conditions entities must fulfill in order to be interlinked. So this framework can be used for *computing* user specified relationships between entities described within different data sources (RDF(S) graphs and/or OWL ontologies ), based on various similarity metrics applied to the target entities and the graph around them. What distinguishes our framework from Silk is the purpose of the search. Our framework discovers *semantic associations* whose structures are not known a priori and which respect given contextual restrictions. Silk is used in ontology alignment scenarios to verify which objects respect the conditions imposed by a certain type of link, and that are specified by the user.

## 6 Conclusion and Future Work

In this paper, we have studied ways of improving the analytical power of semantic associations. We propose a means for exploiting the temporal and spatial dimensions of Semantic Web resource descriptions in order to discover new *ontology-paths* linking two given individuals. Using ONTOAST as a spatial reasoner for OWL ontologies, new inferences can be produced that lead to new and possibly helpful insights into the ways in which individuals are connected within ontological domains.

We currently consider the possibility of introducing a new module of Trust management into the Semantic Analysis Framework. Its purpose would be, on the one

hand, to filter the ontological knowledge and, on the other hand, to communicate with the Result Classification Module for determining the weights of the discovered semantic associations.

As a primary future goal, we intend to implement a prototype of the Semantic Analysis Framework and to test our algorithms on an extended ontological base in order to quantify their relevance and performance in real world semantic association discovery scenarios. We also plan a detailed performance study using large synthetic OWL datasets.

We also plan to explore the definition of more complex spatial and temporal contexts, built using conjunctive and/or disjunctive regular contexts. For instance, it will be interesting to express a spatial context that represents the French or the South African territory except the territories of the Isere French department:

$C_{Spatial} = \{(France \vee SouthAfrica) \wedge (\neg Isere)\,,\ qsr\colon contains\}$.

## References


Agarwal, P.: Ontological Considerations in GIScience. International Journal of Geographical Information Science 19 (2005)

Allen, J.: Maintaining knowledge about temporal intervals. Communication of the ACM, 832–843 (1983)

Anyanwu, K., Sheth, A.: p-Queries: Enabling Querying for Semantic Associations on the Semantic Web. In: WWW 2003, Budapest, Hungary (2003)

Balmin, A., Colby, L., Curtmola, E., Li, Q., Ozcan, F., Srinivas, S., Vagena, Z.: SEDA: A System for Search, Exploration, Discovery and Analysis of XML Data. In: VLDB 2008 (2008)

Berners-Lee, T., Hendler, J.A., Lassila, O.: The Semantic Web. Scientific American 1, 34–43 (2001)

de Kunder, M.: The size of the World Wide Web (2008), `http://www.worldwidewebsize.com/`

Egenhofer, M.: Towards the Semantic Geospatial Web. In: ACM GIS 2002, vol. 1, pp. 34–43 (2002)

Gutierrez, C., Hurtado, C., Vaisman, A.: Temporal RDF. In: Second European Semantic Web Conference, Heraklion, Crete, Greece (2005)

Halaschek, C., Aleman-Meza, B., Arpinar, B., Sheth, A.: Discovering and Ranking Semantic Associations over a Large RDF Metabase. In: Proceedings of the 30th VLBD Conference, Toronto, Canada (2004)

Hobbs, J., Pan, J.: An Ontology of Time for the Semantic Web. ACM Transactions on Asian Language Information Processing 3, 66–85 (2005)

Hobbs, J., Pan, J.: Time Ontology in OWL. W3C Working Draft (September 27, 2006)

Miron, A., Capponi, C., Gensel, J., Villanova-Oliver, M., Ziebelin, D., Genoud, P.: Rapprocher AROM de OWL. In: LMO, pp. 99–116 (2007a) (in French)

Miron, A., Gensel, J., Villanova-Oliver, M., Martin, H.: Towards the Geo-spatial Querying of the Semantic Web with ONTOAST. In: Ware, J.M., Taylor, G.E. (eds.) W2GIS 2007. LNCS, vol. 4857, pp. 121–136. Springer, Heidelberg (2007b)

Moisuc, B., Davoine, P., Gensel, J., Martin, H.: Design of Spatio-Temporal Information Systems for Natural Risk Management with an Object-Based Knowledge Representation Approach. Geomatica 59(4) (2005)

O'Dea, D., Geoghegan, S., Ekins, C.: Dealing with Geospatial Information in the Semantic Web. In: Proc. Australasian Ontology Workshop (AOW 2005), Sydney, Australia, pp. 69–73 (2005), http://crpit.com/abstracts/CRPITV58ODea.html

Page, M., Gensel, J., Capponi, C., Bruley, C., Genoud, P., Ziebelin, D., Bardou, D., Dupierris, V.: A New Approach in Object-Based Knowledge Representation: the AROM System. In: Monostori, L., Váncza, J., Ali, M. (eds.) IEA/AIE 2001. LNCS (LNAI), vol. 2070, pp. 113–118. Springer, Heidelberg (2001)

Patel-Schneider, P.F., Hayes, P., Horrocks, I.: OWL Web Ontology Language Semantics and Abstract Syntax. W3C Recommendation (February 10, 2004), http://www.w3.org/TR/owl-semantics/

Perry, M., Sheth, A.: A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data. Technical Report Technical Report KNOESIS-TR-2008-01 (2008)

Ressler, J., Dean, M.: Geospatial Ontology Trade Study. In: Ontology for the Intelligence Community (OIC 2007), Columbia, Maryland (2007)

Sheth, A., Arpinar, B., Kashyap, V.: Enhancing the Power of The Internet: Studies in Fuzziness and Soft Computing. In: Relationships at the Heart of Semantic Web: Modeling, Discovering, and Exploiting Complex Semantic Relations (2002)

Tarjan, R.: Fast Algorithms for Solving Path Problems. ACM 28, 594–614 (1981)

Volz, J., Bizer, C., Gaedke, M., Kobilarov, G.: Silk - A Link Discovery Framework for the Web of Data. In: LDOW 2009, Madrid, Spain (2009)

# Statistically Valid Links and Anti-links Between Words and Between Documents: Applying TourneBool Randomization Test to a Reuters Collection

Alain Lelu and Martine Cadot

**Abstract.** Neighborhood is a central concept in data mining, and a bunch of definitions have been implemented, mainly rooted in geometrical or topological considerations. We propose here a statistical definition of neighborhood: our TourneBool randomization test processes an objects $\times$ attributes binary table in order to establish which inter-attribute relations are fortuitous, and which ones are meaningful, without requiring any pre-defined statistical model, while taking into account the empirical distributions. It ensues a robust and statistically validated graph. We present a full-scale experiment on one of the public access Reuters test corpus. We characterize the resulting word graph by a series of indicators, such as clustering coefficients, degree distribution and correlation, cluster modularity and size distribution. Another graph structure stems from this process: the one conveying the negative "counter-relations" between words, i.e. words which "steer clear" one from another. We characterize in the same way the counter-relation graph. At last we generate the couple of valid document graphs (i.e. links and anti-links) and evaluate them by taking into account the Reuters document categories.

**Keywords:** Statistical Graph Extraction, Randomization Test, Robust Data Mining, Unsupervised Learning, Text Mining.

Alain Lelu
Université de Franche-Comté / LORIA, Kiwi team,
bâtiment C, Campus scientifique,
BP 239, F-54506 Vandoeuvre-lès-Nancy Cedex, France
e-mail: Alain.Lelu@univ-fcomte.fr

Martine Cadot
Université Henri Poincaré - Nancy 1 / LORIA, ABC team,
bâtiment A, Campus scientifique,
BP 239, F-54506 Vandoeuvre-lès-Nancy Cedex, France
e-mail: Martine.Cadot@loria.fr

F. Guillet et al. (Eds.): Advances in Knowledge Discovery and Management, SCI 292, pp. 307–324.
springerlink.com 

# 1 Introduction: Rationale and Objective

The definition of the neighborhood is a central issue in data mining: this concept is at the heart of supervised methods, such as K-nearest neighbors learning, or unsupervised ones, as those based on graphs. Numerous definitions have been proposed:

- K-nearest neighbors, and their derivatives: reciprocal neighbors (Benzécri, 1982), K-reciprocal neighbors (Lelu, 2004).
- Relative neighbors (Toussaint, 1980): two points are deemed neighbors if no other point stands in the "lune" they define, i.e. the intersection of the two spheres centered in these points, which common radius is the inter-point distance.
- Gabriel neighborhood: two points are deemed neighbors if no other point lies in the sphere which diameter is their distance (Gabriel and Sokal, 1969).
- The Delaunay triangulation (Delaunay, 1934; Goodman and O'Rourke, 2004) is such as no other point lies in the sphere circumscribed to any simplex resting on these points.

These definitions rely on geometric and topologic concepts (Scuturici *et al.*, 2005); their advantage resides in their adaptiveness to orders-of-magnitude density contrasts, frequent in sparse high-dimensional spaces. But they are not immune to side-effects, such as spurious linking between two extraneous elements.

We will explore here another track, i.e. we will derive links between neighboring rows or columns of a datatable from statistical considerations. We will limit our study to binary, all-or-none features describing a set of instances or observations. From this point of view, two features will be considered significantly linked if their co-occurrence in the instances is greater than expected under the hypothesis of random distribution; and symmetrically for two instances. We will go into this notion of statistical expectation further on.

Note that one of the novelties of our approach is to define and take into account the notion of "anti-link": in the same way as a feature pair (resp. an instance pair) may share more instances (resp. features) than expected, they also may share significantly less instances (resp. features) than expected.

In section 2 we will first introduce our TourneBool randomization test and argue about it, as our objective consists in deriving two graphs pairs out of a binary data table, i.e. the statistically valid relations and counter-relations between the features on the one hand, and between the instances on the other hand. In the third section we will expose and discuss the application of our test to the RCV1 Reuters test corpus. Conclusions and perspectives will be outlined in the last section.

# 2 The TourneBool Randomization Test

## *2.1 Local vs. Context-Embedded Tests*

A common way to define the significance of the relation between two binary attributes X and Y is to consider the sole four values of the two-way table relating *X*

and $Y$, usually noted $a$, $b$, $c$, $d$ (see Table 1), and compare them to their theoretical values in case of unrelatedness. A statistical test is then used for the comparison.

**Table 1** The four elementary components of the local association indices between $X$ an $Y$

| | Y | non Y |
|---|---|---|
| X | $a$ | $b$ |
| non X | $c$ | $d$ |

The independence Chi-square test(Morineau *et al.*, 1996) is well-known, but not systematically used, for it is not well-fitted to a widespread class of datatables such as those issued from text databases. This test is actually irrelevant in case of strongly unbalanced counts: none of the four $\hat{a}$, $\hat{b}$, $\hat{c}$, $\hat{d}$ theoretical values in the case of independence has to be too low, say less than 5 or 4 (Yates, 1934). It is also irrelevant in case of large counts, where it tends to be always positive - textual data generally use to fall into this category. Amongst other tests more fitted to unbalanced data, we will cite (1) the linkage likelihood test (Lerman and Peter, 2003), which includes a probabilistic model of the imbalance, (2) the exact test (Fisher, 1936), which proceeds by counting the number of different configurations of $a$, $b$, $c$ and $d$, under the constraint of given margins $a+b$, $c+d$, $a+c$, $b+d$.

The tests based on the only four cells of Table 1 share a common feature with the Chi-square independence test: the relations extracted by these tests express local associations between two variables, whatever the values of the remaining ones. Indeed we think that this position cannot be defended, as can be easily shown with a simple thought experiment: if two binary variables are massively true for a large common set of instances, and seldom true outside this set, a local point of view would conclude to a strong association between them. But what if the other variables were also systematically true for the same set of instances ? The abovementioned association would express nothing but a general redundancy phenomenon in the data, and nothing worth of interest. Hence the necessity to consider *all* the variables for whatever conclusion about two variables.

Generally speaking, these local tests are unsuitable for the type of data mostly encountered in the data mining tasks: large numbers of variables, with very heterogeneous distributions - typically: “Zipf-like” power-law distributions (Newman, 2005). The use of these tests brings out problems that are increasingly noticed by statisticians, such as the multiple comparisons problem (Jensen and Cohen, 2000), the issue of the exhaustive sample (Press, 2004), or the relative independence (Bavaud, 1998).

Another research line is now possible, thanks to the growing performances of the computer resources, and the progress of their massive interconnection: the comparison with full-scale random simulations is feasible, and is an alternative to the traditional comparisons with asymptotic theoretical statistic distributions. In this way, noise may be added to the original datatable (bootstrap and Jacknife methods), or

purely random tables may be generated, submitted to the same structural constraints as the original one.

Two possibilities arise at this step: one may generate random versions of a datatable, starting from its sole margins: we have tried this approach in Cadot and Napoli (2003), with the drawback of having to cope with the problem of multiple ‘1 values in the same cells, which problem we have empirically solved, without proving the universality and convergence of the solution.

Or one may generate the random versions starting from the original database itself, by a sequence of elementary transformations keeping the row and column margins constant. This is the direction we eventually chose, when designing our TourneBool method and test: a method for generating random versions of a binary datatable with prescribed margins, and the ensuing test for validating relations between rows or between columns. It is to be noted that the principles of generation of random matrices with prescribed margins seem to have been discovered independently several times, in various application domains: ecology (Connor and Simberloff, 1979; Cobb and Chen, 2003), psychometrics (Snijders, 2004), combinatorics (Ryser, 1964). As for our team, one of us presented (Cadot, 2005) a permutation algorithm based on rectangular flip-flops, incorporating a monitored convergence of the algorithm. Its theoretical legitimation can be found in Cadot (2006), based on the original notion, to the best of our knowledge, of cascading flip-flops: we have shown that any Boolean matrix can be converted into any other one with the same margins in a finite number of such cascades. These cascading flip-flops are themselves compositions of elementary rectangular flip-flops. The novelty of our approach stays in that we use the resulting randomized matrices for testing the validity of the statistical links between pairs of Boolean variables, whatever the nature of the relation: positively linked variables on the one hand, or “anti-linked” variables which co-occurrence is less than expected on the other hand. Other authors (Gionis *et al.*, 2007) have also used randomized versions of an observed matrix, but with a different goal, in order to compare, in the original matrix and in the simulated ones, the global number of links which frequency count exceeds a given threshold (“frequent itemsets”).

## *2.2 Step 1: Generating the Randomized Matrices*

As is the case for all other randomization tests (Manly, 1997), the general idea comes from the exact Fisher test (Fisher, 1936), but it applies to the variables taken as a whole, and not pairwise. It behaves as a sequence of elementary flip-flops which do not modify the row and column sums. These flip-flops preserve the irreducible background structure of the datatable, but break up the meaningful links specific to a real-life data table. Consider for example a text vs. words incidence matrix: if some words appear in nearly all the texts, they will appear as such in all the simulated matrices too, and no link between these words will ensue. Now consider a few long texts systematically comprising none of these considered frequent words: the simulated matrices will not reproduce this interesting feature, which will only

be brought to light by comparison to the original one. In this way, comparing with simulations allows one to depart the background structural part of a linkage out of the other part, the one we are interested in. The background structure depends on the application domain, and on the distributions of the margins. For example, most of $texts \times words$ datatables have a power-law distribution of the words, and a binomial-like one for the number of unique words in the texts. This background structure induces our "statistical expectation" of no links conditionally to the type of corpus. Getting rid of the background structure enables our method to process any type of binary data, both (1) taking into account the marginal distributions, (2) doing this without any need to specify the statistical models of these distributions.

The number of rectangular flip-flops is controlled by two Hamming distance measures between matrices (i.e. number of cells with opposite values): 1) between the current random matrix and the one generated at the previous step, 2) between the current random matrix and the original one. The initial number of flip-flops is increased as long as these distances are growing. The value of this parameter is deemed optimal when they stabilize - in practice, about several times the number of ones in the original matrix. No bias, i.e. residual remnant of the original matrix, can be attributed then to the randomization process.

## 2.3 Step 2: Establishing the Links and Anti-links

The principle for extracting the links is as follows: Let $m_i$ and $m_j$ two words simultaneously occurring in $p_0$ texts ($p_0 \geq 0$) of the original corpus, and in $p_k$ texts ($p_k \geq 0$) of each k-th random simulation of the datatable (k in [1,K]). The link between $m_i$ and $m_j$ may be assigned to one of the three following cases:

- if $p_0$ is greater than the near-entirety of the $p_k$'s, then $link(m_i, m_j) > 0$ (attraction)
- if $p_0$ is lesser than the near-entirety of the $p_k$'s, then $link(m_i, m_j) < 0$ (repellency)
- else: $link(m_i, m_j) = 0$ (independence given the corpus[1])

In the first case, we will call it a significantly positive link, or shortly speaking, a link. In the second one we will call it a significantly negative link, or "anti-link". We will label the third one "null link", or non-significant link.

For example, Fig. 1 shows the ordered set of the 100 values $p_k$ for a word pair throughout 100 simulations (k spanning from 1 to 100). If we choose an alpha risk threshold[2] of 10%, we set up two limits (marked by triangles in Fig. 1): the value $p = 2$ corresponds to $k = 6$, the value $p = 22$ to $k = 95$. So the bilateral 90% confidence interval of $p$ in the case of no link between $m_i$ and $m_j$ is estimated by the interval $[2, 22]$, which includes the 90 less extreme values in the set of $p_k$'s.

[1] And not independence given presumed repartition laws, as is the case with parametric tests.

[2] Alpha risk: risk of making a mistake when deeming significant a value of p due to plain randomness; it corresponds to the notion of "false positive" in the context of clinical tests. The notion of "false negative" corresponds to the Beta risk: the risk of making a mistake when deeming "due to randomness" a significant p.

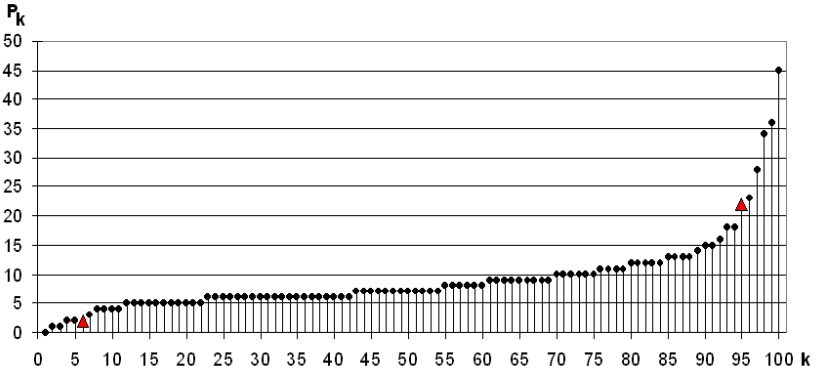



**Fig. 1** Example of ordered sequence of the $p_k$ values (numbers of co-occurrences between two words $m_i$ and $m_j$ throughout 100 simulations)

Depending on the $p_0$ value (i.e. the number of word co-occurrences in the original data-table) lying or not in this confidence interval, the link will be deemed non-significant or significant. In this example a $p_0$ value of 0 or 1 will qualify the word pair as significantly and negatively linked, i.e. the infrequency of their co-occurrence in the texts must be meaningful; as an example, it is the case we encountered for the words "USA" and "Middle Ages" in a sample of encyclopedic corpus. We may denominate "repellent" this kind of anti-link. A $p_0$ value of 23, 24 or more will indicate a significant positive link of indisputable co-occurrence. In contrast, for $p_0 = 2, 3, ..., 21, 22$ the relation is imputed to chance, and the conclusion is that the two words are not linked by meaning, but by the background structure of the corpus. Using this test enables one to bring to light the meaningful word associations, in a robust and statistically valid manner. Of course a crucial parameter is the way of splitting the texts: small statistical text units such as sentences or parts-of-speech will yield short-range associations, often due to frozen expressions or phrases; longer units such as scientific abstracts or newswires will yield the same associations, plus longer-range thematic links.

We present in the gray box (page 313) our TourneBool algorithm for finding out the significant links, whether positive or negative. Data are disposed in a Boolean table, in which columns figure the variables among which links are searched for; e.g. columns are words, rows are texts, with ones or zeros at the intersections, depending on a given word being present or not in a given text.

When using this algorithm, one must fix the values of three parameters: the number of rectangular flip-flops for generating non-biased random matrices, the number of randomized matrices, the alpha risk. The two last parameters are fixed in accordance with the usual compromises: on the computer science side, the trade-off between speed and quality - the more simulated matrices, the higher the quality of estimation, but the longer the computation time, too... We use to ask for 100 or 200 simulations. On the statistical side, the trade-off between the alpha and beta risks:

## TourneBool algorithm

Let $M$ a $(n, p)$ boolean matrix, with $n$ objects in rows and $p$ variables in columns.

**Main : TourneBool**

1. build $q$ randomized versions of $M$
2. for each column pair $(i, j)$ of $M$
   - compute the $p_0$ value of $p$, i.e. the number of co-occurrences of the column pair $(i, j)$ of $M$.
   - build the confidence interval of $p$ after the list of the $q$ randomized matrices.
   - compare $p_0$ to the bounds of the confidence interval; 3 cases:
     – if it is lesser than the lower bound, the link is declared significantly negative, and is thus kept on,
     – if it is greater than the upper bound, it is declared significantly positive, and is thus kept on,
     – if the original $2-$itemset support $p_0$ in $M$ stays in between this interval, it is declared unsignificant and is thus eliminated.

**Building a randomized version of $M$**

Choose a number $r$ of rectangular flip-flops to execute.

1. copy $M$ to $Mc$
2. repeat $r$ times :
   - randomly choose a row pair and a column pair with replacement
   - if the zeros and ones alternate at the vertices of this rectangle in $Mc$, then modify $Mc$ moving each value into its complement to 1, else do nothing.
3. store $Mc$

**Building the confidence interval, at risk alpha, of the number $p$:**

1. for each randomized version $M_k$ of $M$ compute the number $p_k$ of co-occurrences of the two columns $i$ and $j$ (dot product of the two columns). Store all the $p_k$ in a list.
2. sort the list in ascending order. The lower bound is the list element with rank $q \times alpha/2$, and the upper bound the one with $q \times (1 - alpha)/2$ rank.

the smaller the alpha value, the lesser the risk of extracting links due to the sole chance, but also the greater the beta risk of rejecting significant and meaningful links. Our experience is to fix the value to the usual 5% or 1%. As for the first parameter (the number of elementary flip-flops), our rule of thumb is to start with four times the number of ones in the matrix, and adjust it, if necessary, considering the sequence of the computed Hamming distances.

It is to be noted that the permutation tests, from which emanate the randomization tests, have been proven to be the most "powerful" ones, i.e. to minimize the beta risk for a given alpha risk (Droesbeke and Finne, 1996).

### *2.4 Scaling the Method*

After a first promising application to a small corpus (Lelu *et al.*, 2006), we have scaled up the implementation of our algorithm in order to process real-size corpora. The time and space complexities for the first step (generating $k$ randomized versions of the initial matrix) are respectively $O(k \times v)$ and $O(n \times m \times v)$, where $n$ and $m$ are the row and column numbers, $v$ the number of ones in the matrix. As composed of independent processes, this step is fitted to direct parallelization. The last step consists of making use of the randomized matrices for determining the type of each link - positive, negative or null. As a direct implementation would imply storing 100 or 200 contingency tables of size $(n,n)$, we have split this task into parallel processes, each one in charge of the same fraction of all the contingency tables. The application of this method to a Reuters test corpus (23,000 newswires, 28,000 words) as described below was executed in 36 hours as three parallel processes on a standard quadcore PC.

## 3 Application to the RCV1 Reuters Corpus

Thanks to D. Lewis and Reuters news agency[3] (Lewis *et al.*, 2004), several newswires test corpora have been made available to the machine learning community, in order to compare the methods and results on a common and public access basis.

We used the RCV1 training corpus - 23,149 newswires supplied with their lemmas (see Fig. 2) and their document categories. We chose this delivery to avoid disturbances due to a proprietary indexing, as our focus is on the processing of a given and publicly available indexed corpus, not on the indexation process. As can be seen in Fig. 2, the lemmas amount to simple word truncations, limiting the size of the vocabulary to a few ten thousands terms, but at the same time creating many ambiguities for basic English words. No noun phrases or named entities, known for being much less equivocal, are provided.

[3] Lewis, D. D. RCV1-v2/LYRL2004: The LYRL2004 Distribution of the RCV1-v2 Text Categorization Test Collection (12-Apr-2004 Version) `http://www.jmlr.org/papers/volume5/lewis04a/lyrl2004_rcv1v2_README.htm`.

.I 2290
*compan compan limit planet planet planet planet planet hollywood hollywood hollywood hollywood hollywood hollywood launch credit credit credit credit card card card card dine feel movie movie star money arnold schwarzeneg them restaur restaur rate applic chain fast outlet festoon kitsch memorabl team william mor talent agent mbna bank wilmington del perk prefer roll seat edition shirt discount food merchandis annual nnual join pop cultur ston magazin issu debt fun usual stat percent percent approv pay special introduc balanc transf cash brand advanc check orland florid off grow made spend part americ base includ intern don make fee*

**Fig. 2** Example of a Reuter's newswire indexed by lemmas

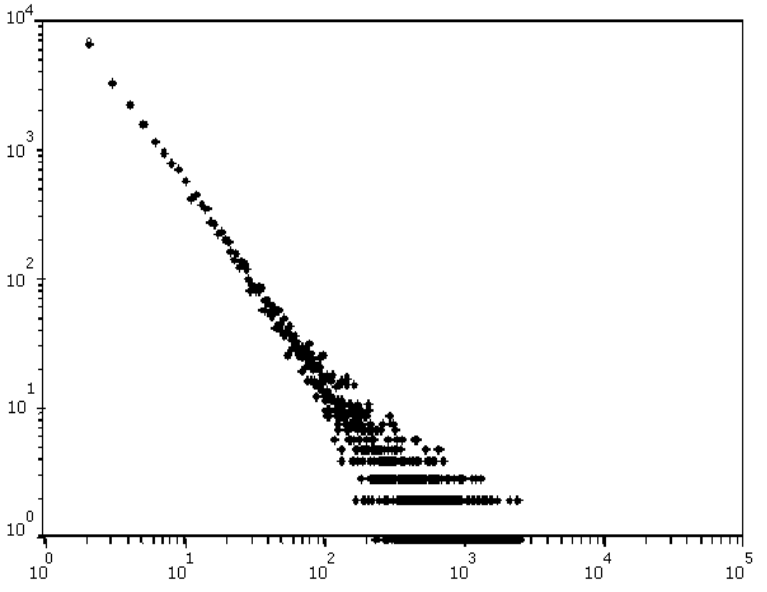



**Fig. 3** Characterizing the lemmas in the RCV1 Reuters corpus: horizontally, occurrences of each lemma in the corpus. Vertically: number of repetitions of these occurrences in the corpus. The coordinates are log-log. For example one may read: there are 7,100 lemmas occurring two times.

Each newswire in the training set is manually attributed one or several descriptive categories among 101. The main objective when delivering such corpus is to assess the quality and the generalization ability of machine learning methods. Our goal is slightly different: after characterizing the two graph pairs (links and anti-links) representing respectively the relations between words and between newswires, we

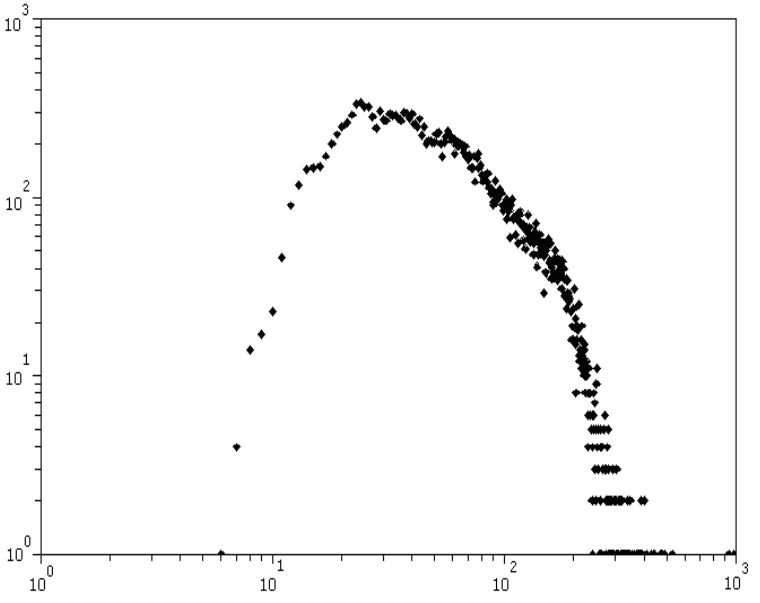



**Fig. 4** Characterizing the newswires in the RCV1 Reuters corpus: horizontally, occurrences of unique lemmas in each newswire. Vertically: number of repetitions of these occurrences in the corpus. The coordinates are log-log. For example one may read: 324 newswires are composed of 26 unique words.

will use the Reuters categories as a first attempt to assess the ability of our method to agree (or not) with links issued from a human expertise, in a non-supervised way.

To begin with, let us examine a few statistics on our corpus: after eliminating the words with unique occurrence, which support, by definition, no relational information, the size of the vocabulary amounts to 28,450 lemmas, from *a0* to *zywnociowej*; each newswire features a mean value of about 75 unique lemmas. The occurrence distribution of the lemmas has a typical Zipf-like appearance (Fig. 3), i.e. a power-law distribution with a power coefficient of around -1.5. The distribution of the number of unique lemmas per newswire is markedly skewed and unbalanced, with a mode around 26 (Fig. 4).

## *3.1 Word Graphs: Links and Anti-links*

Our TourneBool method has generated the adjacency matrices of the link graph between words at the confidence threshold of 99%. This graph comprises 28,450 vertices and 1.4 million links (see Table 2). Its density is 0.0071. As a comparison, the raw word co-occurrence matrix is much denser (0.0406). In the same way, the anti-link graph comprises 245,000 negative links (density: 0.0012).

**Table 2** Comparison of three word graphs, which adjacency matrices are 1) the raw co-occurrence matrix, 2) the matrix of valid links, 3) the matrix of valid anti-links

| FOR WORDS: | Graph of raw co-occurrences | Graph of valid links | Graph of valid anti-links |
|---|---|---|---|
| Number of vertices | 28,450 | 28,450 | 28,450 |
| Number of links | 16.0 M | 1.4 M | 0.24 M |
| Density | 0.0406 | 0.0071 | 0.0012 |
| Mean degree | 578 | 100.7 | 17.2 |
| Degree correlation between neighbors | -0.390 | +0.017 | -0.370 |
| Cliquishness coefficient | 0.808 | 0.305 | 0.294 |
| Total number of clusters | n.a. | 21 | 21,200 |
| Number of clusters of size one | n.a. | 3 | 20,000 |
| Maximal value of the modularity index for the clusters | n.a. | 0.276 | 0.011 |

### Characterizing graphs

It is a dozen years since what is now known as "complex networks" started to be extensively studied (Watts and Strogatz, 1998). Out of our specific research domain on text databases, the graph formalism of this approach impacts as dissimilar applications as the study of social networks, especially when mediated by Internet, or the gene interaction networks... As our statistical validation process of word links is an all-or-none process, an unoriented graph representation is well-fitted to a large-scale set of such links. As far as the number of involved texts and words is generally well above 10,000, one can define these relations as those of a complex network, and we will use here a few standard indicators for characterizing such graphs:

- number of vertices and edges,
- graph density: number of edges / maximum potential number of edges,
- mean degree of the vertices, and degree distribution,
- Pearson's correlation coefficient between degrees of neighboring vertices,
- mean and distribution of the clustering coefficients of the vertices ("cliquishness" indicators: from 0 when no neighbors are mutually linked, to 1 when they constitute a complete clique),
- clusters (or "communities") of densely linked vertices (words or texts in our application domain). We used the WalkTrap open access software[4](Pons and Latapy, 2006) for this task.

[4] WalkTrap `http://www-rp.lip6.fr/~latapy/PP/walktrap.html`

The mean degrees of the link and anti-link graphs are respectively 100.7 and 17.2, contrasting with 578 for the raw co-occurrence graph. The degree correlation between neighboring vertices is negligible (+0.017) for the links ("non-assortative" network), but frankly negative (-0.37) for the anti-links ("anti-assortative" network) as is the case of the co-occurrence network (-0.39): vertices with quite a different number of links tend to aggregate. It follows that the process of statistical validation of the co-occurrence links has deeply altered the structure of this baseline network.

We will not show the degree distribution of words, similar to the words-per-newswire distribution (Fig. 4), and not as scattered as the corresponding feature of the raw co-occurrence matrix; this contrasts with the same feature for the anti-link graph, which follows a clear power-law. As for most of the complex networks, the mean clustering coefficients are rather high: 0.305 for links, 0.294 for anti-links, while the co-occurrence graph exhibits a prominent 0.808.

The WalkTrap graph clustering software (see footnote 4), when parametered with a reasonable 6-steps random walk, has provided us with a hierarchical cluster tree that we have severed at the maximum value of the modularity index: the word links yield 21 clusters, the size of which is greater than 55 for 13 of them; two of them include more than 7000 words, in a very unbalanced repartition. When examining their word content, this clustering seems odd: one of the clusters is mainly made of first names, another one mainly gathers town names of Great Britain... in any case, they mix, with various proportions, place names, person or firm names, together with content words, for example in aeronautics, chemistry or computer science. A possible explanation may reside in the operation principle of the lemmatizer:

1. Many frequent words in English have just a syntactic or rhetorical function, independent of the thematic context, and do not give rise to valid links;
2. A good proportion of middle-frequency words belong to the general English and are akin to characterize thematic contexts (economy, business, politics, transportation,...), but the lemmatization process may have heavily pruned and regrouped them (*phon*, *promot*, *activ*, *typ*,...), turning them into non-contextual elements;
3. The named entities, with longer and often non-English names (*zurawsk*, *zvoncakov*, *zyuganov*,...), have better resisted this treatment, and have retained, more than others, their thematic content.

This hypothesis could be tested by indexing, for comparison, the same corpus in a linguistically more elaborate way, i.e. extracting at least noun phrases and named entities.

Considering now the word anti-links graph, WalkTrap yields 21,000 isolated elements, 143 clusters of size 2 to 10, 31 of size 11 to 50, 2 of size around 500, and 2 of size 2300. In this case the interpretation seems still trickier[5], though one may notice that the large clusters mix general English words with other elements.

[5] The interpretation of clusters of anti-linked elements is difficult indeed: the common-sense relation of closeness is transitive ("My friend's friends are my friends"), but the repellency one is different ("My enemies' enemies have no reasons to be my friends, nor my enemies").

### *3.2 Newswires Graphs: Links and Anti-links*

In the same way as for words, we have applied our TourneBool method for extracting the adjacency matrices of links and anti-links between the 23,149 newswires, at the confidence threshold of 99%. The link graph is composed of 6.4 million edges, hence its density is 0.048. As a comparison, the word graph is much less dense: 0.007. The anti-link graph comprises 3.0 million edges (density: 0.0224), when the corresponding word graph is 18-fold sparser (0.0012). The mean degree is 558.4 (resp. 259.4), while it is 100.7 (resp. 17.2) for the word graphs. The degree correlation between neighboring vertices amounts to a rather strong +0.238 for the links (assortative network), while being negligible (+0.017) for the word graph (non-assortative network); the anti-links network is anti-assortative (-0.248), as is the corresponding word graph (-0.370).

The general shape of the degree distribution of the newswires (Fig. 5) is somehow similar to the distribution of the words-per-text one (Fig. 4) while the degree distribution of the anti-links graph evokes, to some extent, a power-law shape (Fig. 6). The mean clustering coefficients are a bit higher than those of the word graphs: 0.342 vs. 0.305 for the links, 0.387 vs. 0.294 for the anti-links.

Each newswire is attributed one or several thematic categories by the Reuters indexers. The repartition of these 101 overlapping categories is very uneven (Fig. 7); the most important one (CCAT: corporate / industrial) involves nearly half of the newswires, and the whole corpus is covered by the only top-5 categories. Better

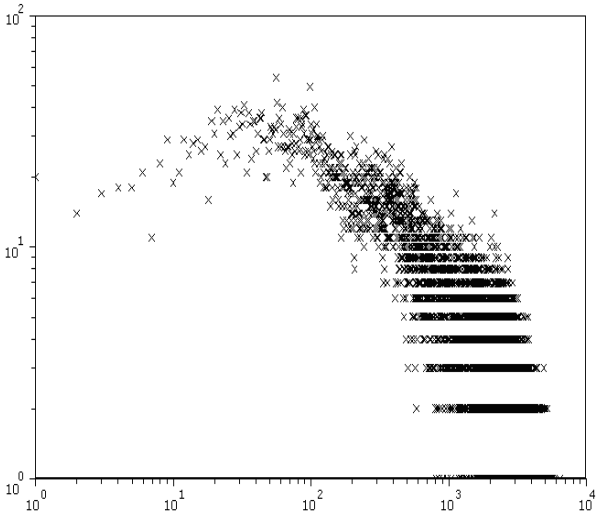



**Fig. 5** RCV1 Reuters corpus: graph of the valid links between newswires; horizontally, degrees of the newswires. Vertically: number of repetitions of these degrees. The coordinates are log-log.

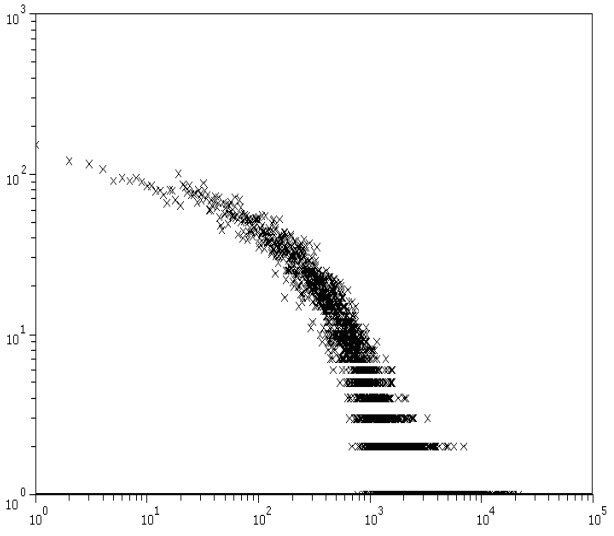


**Fig. 6** RCV1 Reuters corpus: graph of the valid anti-links between newswires; horizontally, degrees of the newswires. Vertically: number of repetitions of these degrees. The coordinates are log-log.

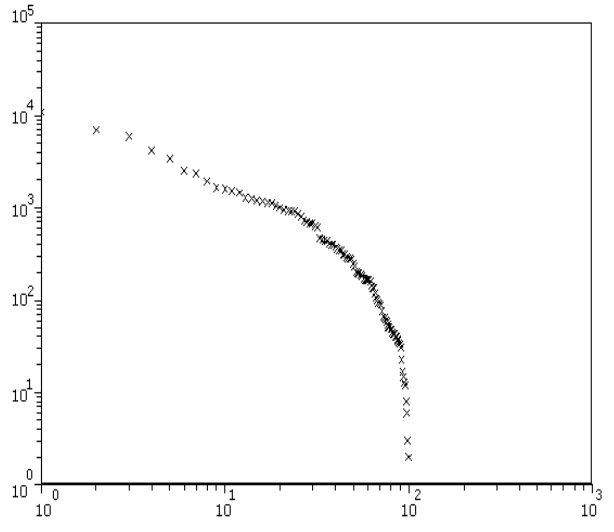


**Fig. 7** Size repartition of the 101 overlapping Reuters categories: horizontally, ranking of the categories by decreasing size. Vertically: size of the categories. The coordinates are log-log. For example there are 10,786 newswires in the largest category.

than unsupervised methods such as clustering, unavoidably arbitrary to some extent, these categories will help us to assess the relevance of our links and anti-links. We will limit ourselves in the present study to an overall check, and to a qualitative confirmation:

1. It should be desirable that a statistically valid link could correspond to at least one common Reuters category at both side of the link. This idea may be approximated by computing the density of valid links for the complete sub-graph (i.e. clique) corresponding to each category.
2. Conversely, it should be desirable that a anti-link could join two newswires with no common category. This can be tested by computing the density of anti-links for the complete sub-graph (i.e. clique) corresponding to each category: the hypothesis is verified if the density is zero, or negligible.

Actually, while the overall density of the valid links graph is 0.04, the link densities exceed this value for the near-entirety of the classes. They even exceed 0.5 in 19 of them. Table 3 shows the title of a few such classes in strong agreement with TourneBool links, and the titles of classes in the opposite case. One can verify that the classes in full agreement involve sharp and factual themes in economics and finance, when classes in low agreement are either large, fuzzy themes, such as Corporate / Industrial, or classes dealing with subjects out of economics and finance, such as arts, health, crime, science, foreign affairs,...We suggest the hypothesis that the words of the general journalistic english language are poorly represented and

**Table 3** The top-15 Reuters categories correctly (left) / poorly (right) accounted for by the valid positive links between texts

| Correctly related categories | Poorly related categories |
|---|---|
| C1511: ANNUAL RESULTS | C13 : REGULATION / POLICY |
| E121: MONEY SUPPLY | C32 : ADVERTISING / PROMOTION |
| E13 : I NFLATION / PRICES | CCAT: CORPORATE / INDUSTRIAL |
| E131: CONSUMER PRICES | G159: EC GENERAL |
| E132: WHOLESALE PRICES | GCAT: GOVERNMENT / SOCIAL |
| E14 : CONSUMER FINANCE | GCRIM: CRIME, LAW ENFORCEMENT |
| E141: PERSONAL INCOME | GDIP: INTERNATIONAL RELATIONS |
| E142: CONSUMER CREDIT | GDIS: DISASTERS AND ACCIDENTS |
| E143: RETAIL SALES | GENT: ARTS, CULTURE, ENTERTMT. |
| E31 : OUTPUT/CAPACITY | GENV: ENVIRMT., NATURAL WORLD |
| E311: INDUSTRIAL PRODUCTION | GHEA: HEALTH |
| E513: RESERVES | GODD: HUMAN INTEREST |
| E61: HOUSING STARTS | GPOL: DOMESTIC POLITICS |
| E71: LEADING INDICATORS | GSCI: SCIENCE AND TECHNOLOGY |
| M11: EQUITY MARKETS | GSPO: SPORTS |

ambiguous when lemmatized, whereas the specific vocabulary and recurrent named entities of the specialized economic and financial stories are more preserved.

The computed anti-link density successfully confirms our hypothesis of no anti-links between words of the same category: the mean density is 0.0055, with a maximum of 0.063.

## 4 Conclusions, Ongoing Directions

We have shown in this paper that, starting from a binary *instances* × *attributes* matrix, and taking as an example a real-size *texts* × *words* matrix, it is possible to derive two pairs of statistically valid graphs, one pair for the links and anti-links between instances, and the same for the attributes, using a set of randomized versions of the data matrix. This contrasts (1) with the geometric and topologic approaches for deriving the neighborhoods without any statistical validation, (2) with purely local and pairwise statistical tests. The whole data table is involved for determining each link or anti-link, and not the sole $a$, $b$, $c$, $d$ values. The resulting graphs do take into account the marginal distributions, for each link or anti-link, but without assuming whatever formal distribution law. Any application domain where the data can be expressed as sparse binary matrices, and which marginals are irreducible to classic statistical distributions may benefit from our approach.

In a first attempt to characterize these graphs, we have investigated their structural differences with the graphs issued from the raw co-occurrence tables, by means of a few usual graph structure indicators. We have also tried to assess the semantic relevance of these graphs, but we have come up against limits due to the crude type of lemmas delivered as text attributes: when clustering the lemmas' graph no clearly interpretable clusters emerged; however, we could check that the texts' anti-link graph was highly compatible with the Reuters' pre-defined categories, and that the link graph was in global agreement with these categories, with an excellent agreement for about 20% of them, i.e. the most factual and specific ones.

Lots of work remain to be done: we have to extend the variety of graph structural indices, we have to compare our graphs to graphs issued from geometric or local-statistics methods, we have to compare them to the structure of the *instances* × *attributes* bipartite graph, ... These comparisons will have to be worked out theoretically, as well as empirically. The interpretations must involve experts of the application domains, especially for the anti-links graphs, the case of which is quite a departure from the usual way of thinking, and perhaps has to involve graph patterns (star shapes ?) rather than clusters.

**Acknowledgements.** We are indebted to Richard Dickinson for his precious help improving the English correctness of our text.

## References

Bavaud, F.: Modèles et données: une introduction à la Statistique uni-, bi- et trivariée. L'Harmattan (1998)

Benzécri, J.: Construction d'une classification ascendante hiérarchique par la recherche en chaîne des voisins réciproques. Les Cahiers de l'Analyse des Données, pp. 208–218 (1982)

Cadot, M.: A Simulation Technique for extracting Robust Association Rules. In: CSDA 2005 (2005)

Cadot, M.: Extraire et valider les relations complexes en sciences humaines: statistiques, motifs et règles d'association. Ph.D. thesis, Université de Franche-Comté (2006)

Cadot, M., Napoli, A.: Une optimisation de l'extraction d'un jeu de règles s'appuyant sur les caractéristiques statistiques des données. In: RSTI, série RIA-ECA, pp. 631–656 (2003)

Cobb, G., Chen, Y.: An application of Markov chain Monte Carlo to community ecology. The American Mathematical Monthly, pp. 264–288 (2003)

Connor, E., Simberloff, D.: The assembly of species communities: Chance or competition? Ecology, 1132–1140 (1979)

Delaunay, B.: Sur la sphère vide. Izvestia Akademii Nauk SSSR, pp. 793–800 (1934)

Droesbeke, J., Finne, J.: Inférence non-paramétrique - Les statistiques de rangs. Editions de l'Université de Bruxelles (1996)

Fisher, R.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 179–188 (1936)

Gabriel, K., Sokal, R.: A new statistical approach to geographic variation analysis. Systematic Zoology, 259–270 (1969)

Gionis, A., Mannila, H., Mielikäinen, T., Tsaparas, P.: Assessing data mining results via swap randomization. ACM Trans. Knowl. Discov. Data (2007)

Goodman, J., O'Rourke, J.: Handbook of Discrete and Computational Geometry. CRC Press, Boca Raton (2004)

Jensen, D., Cohen, P.: Multiple Comparisons in Induction Algorithms. Machine Learning, 309–338 (2000)

Lelu, A.: Analyse en composantes locales et graphes de similarité entre textes. In: Purnelle, G. (ed.) JADT 2004 (2004)

Lelu, A., Cuxac, P., Cadot, M.: Document stream clustering: an optimal and fine-grained incremental approach. In: COLLNET 2006 / International Workshop on Webometrics, Informetrics and Scientometrics (2006)

Lerman, I.-C., Peter, P.: Indice probabiliste de vraisemblance du lien entre objets quelconques. Analyse comparative entre deux approches. Revue de Statistique Appliquée, pp. 5–35 (2003)

Lewis, D., Yang, Y., Rose, T., Li, F.: RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 361–397 (2004)

Manly, B.: Randomization, Bootstrap and Monte Carlo methods in Biology. Chapman and Hall/CRC (1997)

Morineau, A., Nakache, J.-P., Krzyzanowski, C.: Le modèle log-linéaire et ses applications. Cisia-Ceresta (1996)

Newman, M.: Power laws, Pareto distributions and Zipf's law. Contemporary Physics, 323–351 (2005)

Pons, P., Latapy, M.: Computing communities in large networks using random walks. Journal of Graph Algorithms and Applications, 191–218 (2006)

Press, J.: The role of Bayesian and frequentist multivariate modeling in statistical Data Mining. Statistical Data Mining and Knowledge Discovery, 1–14 (2004)

Ryser, H.: Recent Advances in Matrix Theory, Madison (1964)

Scuturici, M., Clech, J., Scuturici, V.-M., Zighed, D.: Topological Representation Model for Image Database Query. Journal of Experimental and Theoretical Artificial Intelligence, 145–160 (2005)

Snijders, T.: Enumeration and simulation methods for 0-1 matrices with given marginals. Psychometrika, 397–417 (2004)

Toussaint, G.: The relative neighbourhood graph of a finite planar set. Pattern Recognition, 261–268 (1980)

Watts, D., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature, 95–118 (1998)

Yates, F.: Contingency table involving small numbers and the Chi2 test. Journal of the Royal statistical society (supplement), 217–235 (1934)

# List of Contributors

**Emanuel Aldea** is currently a PhD student at Telecom-ParisTech (Signal and Image Processing Department). His PhD subject is about learning structured data for image interpretation. His research interests include image processing, kernel design and multiple kernel learning.

**Younès Bennani** received a BS degree in Mathematics and Computer Science from Rouen University, in 1987. Subsequently, he received the M.Sc. and the PhD degree in Computer Science from Paris 11 University, Orsay, in 1988 and 1992, respectively, and the Accreditation to lead research degree in Computer Science from the Paris 13 University in 1998. He joined the Computer Science Laboratory of Paris-Nord (LIPN-CNRS) and in 2001, he was appointed to a Full Professor of computer science in the Paris 13 University. His research interests are unsupervised learning, dimensionality reduction and large-scale data mining. He has published 2 books and approximately 150 papers in refereed conference proceedings or journals or as contributions in books. He is the head of the Machine Learning research team.

**Isabelle Bloch** is Professor at Telecom-ParisTech (Signal and Image Processing Department). Her research interests include 3D image and object processing, 3D and fuzzy mathematical morphology, decision theory, information fusion, fuzzy set theory, belief function theory, structural pattern recognition, spatial reasoning, and medical imaging.

**Marc Boullé** was born in 1965 and graduated from Ecole Polytechnique (France) in 1987 and Sup Telecom Paris in 1989. Currently, he is a Senior Researcher in the "Statistical Processing of Information" research group of France Telecom R&D. His main research interests include statistical data analysis, data mining, especially data preparation and modelling for large databases. He developed regularized methods for feature preprocessing, feature selection and construction, model averaging of selective naive Bayes classifiers and regressors.

**Romain Bourqui** received the PhD degree in 2008 from the University of Bordeaux I. He has been an assistant professor in the University of Bordeaux Department of Computer Science since september 2009. His current research interests are information visualization, graph drawing, graph clustering and their applications to bioinformatics and social science.

**Martine Cadot** received a MS in Mathematics, another one in Economics from the University of Burgundy (France), and a PhD in Computer Science from the University of Franche-Comté. She taught Mathematics, Statistics and Computer Science in Tunisia and several at universities in France (Dijon, Reims, Nancy). Her current courses take place at Nancy 1 University, CS Department, and she is a researcher in the LORIA laboratory (Nancy). During her past research activity, she has co-published with educational scientists, economists, psychologists, physicians and information science experts. Her current research interests and publications focus on the domains of machine learning, robust data mining, variables interaction modeling, and large scale randomization tests.

**Farid Cerbah** is a member of Dassault Aviation scientific department where he is conducting research projects in domains related to data mining and semantic technologies, with applications focused on exploitation of heterogeneous technical repositories. His research interests include NLP corpus-based methods and software infrastructures for terminology and ontology acquisition, Information Retrieval and categorisation methods for accessing and structuring specialised repositories, and ontology learning from structured data sources, such as relational databases.

**Fabrice Clérot** was born in 1962 and graduated from Ecole Polytechnique in 1986. He joined the Centre National d'Etudes des Télécommunications, the research centre of the (then) public French telecom operator which became the R&D Division of France Télécom. He is currently Senior Expert in data-mining and manages the "Profiling" research group. His research interests are in the application of statistical and connexionist methods in areas of interest for telecommunication companies: telecommunication network modelling, traffic prediction, data-mining, stream-mining (with an emphasis on IP networks metrology and data stream summarization) and reinforcement learning.

**Guillaume Cleuziou** is Associate Professor at the University of Orléans (France). His research is done in the "Constraints and Machine Learning" team, it concern supervised and unsupervised learning problems. Since 2004, he focused his works on clustering techniques and developed algorithms for overlapping clustering such as OKM (Overlapping-k-means). The innovative solutions proposed by Guillaume Cleuziou find their application in practical research domains such as Text Mining, Information Retrieval and Bioinformatics.

**Thanh-Nghi Do** is currently a postdoctoral researcher at Telecom Bretagne, France. He is also a lecturer at the College of Information Technology, Cantho University, Vietnam. He received the M.S. and PhD degrees in Computer Science from the University of Nantes, in 2002 and 2004, respectively. His research interests include data mining with support vector machine, kernel-based methods, decision tree algorithms, ensemble-based learning, information visualization. He has served on the program committees of international conferences and a reviewer for journals in his fields, including the the International Conference on Data Mining (DMIN), the Australasian Data Mining Conference, the Pattern Recognition Elsevier, the Journal of Experimental Algorithmics, and others.

**Céline Fiot** received a Master in Computer Engineering at the Ecole des Mines de Nantes, with a specialization in computer science for decision support systems. Since 2004, she worked on mining frequent sequences from imprecise, uncertain, or incomplete data for her PhD, she completed in 2007 at the University of Montpellier. Then, she joined the AxIS team at the INRIA Sophia Antipolis as a Post-doctoral fellow, working on using fuzzy and gradual data mining for atypical behavior detection. Since mid-2009, she is working at the French agricultural and environmental engineering research institute (Cemagref), on designing decision support systems for wine growing and making using techniques from data mining, machine learning and statistics.

**Alexis Gabadinho** is scientific collaborator at the Department of econometrics and the Laboratory of demography of the University of Geneva. He holds a post-graduate diploma in demography. His current research interests are the application of data mining methods in social sciences and the development of methods for categorical state sequences analysis, in particular measures of sequence complexity and methods for summarizing sets of sequences.

**Nesrine Gabsi** is a PhD student under the supervision of Fabrice Clérot at the Research and Development (R&D) Division of France Télécom and Georges Hebrail at the Télécom ParisTech (Paris, France). Her research interests include data mining, data streams summary and querying summarized data streams. In 2007, she received a Master of Science (MS) degree in Computer Science from the University of Versailles where she worked on shared and secured XML SCHEMA for social and medical file adapted to P2P architecture under the supervision of Georges Gardarin in the P.R.I.S.M laboratory. Since 2008, she has been responsible for organizing the BiLab Seminar, a joint laboratory with EDF R&D (Electricité de France) on Business Intelligence. `http://www.telecom-paristech.fr/~gabsi`

**Jérôme Gensel** is a Full Professor at the University Pierre Mendès France of Grenoble, France, since 2007. He received his PhD in 1995 from the University of Grenoble for his work on Constraint Programming and Knowledge Representation in the Sherpa project at the French National Institute of Computer Sciences and Automatics (INRIA). He joined the Laboratory of Informatics in Grenoble (LIG, formerly called LSR-IMAG Laboratory) in 2001. His research interests include Representation and Inference of Spatio-Temporal Information, Ontologies and Knowledge Representation, Geographic Semantic Web, and Ubiquitous Geographical Information Systems.

**Patrick Gros** has been involved in research in the field of image analysis for 18 years. He received his PhD degree from the University of Grenoble - France in 1993. After six years in the GRAVIR lab in Grenoble and one year at the Robotics Institute of CMU (Pittsburgh, PA, USA), he moved in 1999 to the INRIA of Rennes. Patrick Gros's research interests concern multimedia indexing and retrieval in very large collections with applications like copy detection, TV analysis, and audiovisual information retrieval.

**Nistor Grozavu** (Nistor.Grozavu@lipn.univ-paris13.fr) is currently member of Machine Learning Team (A3), LIPN at Paris 13 University. He received the BS and MS degrees in computer science from Technical University of Moldova in 2005 and from Aix-Marseille II University in 2006 respectively. In 2009, Nistor Grozavu received his PhD degree in Computer Science from the Paris 13 University. His research interests are in the areas of data mining: topological clustering analysis, clustering fusion and clustering collaboration.

**Fayçal Hamdi** is PhD Student in IASI-GEMO team at the University of Paris-Sud 11 and INRIA Saclay. He holds his Master degree from the University of Paris 11. He works directly on GEONTO, an ANR project about the interoperability of various data relating to geographical information. His research interests are : Ontology alignment, application of matching techniques in real-world scenarios, large-scale ontology matching and ontology engineering.

**Georges Hébrail** joined Telecom ParisTech (Paris, France) in 2002 as a Professor of Computer Science, after almost 20 years spent as a researcher at Electricité de France R&D. Since 2007, he is also head of the BILab, a joint laboratory with EDF R&D on Business Intelligence. His research interests cover the different domains of business intelligence: databases, data warehouses, statistics, and data mining. He is currently working on the use of data stream approaches to handle very large volumes of data in two application domains: telecommunications and electric power consumption. `http://www.telecom-paristech.fr/~hebrail`

**Gilles Hubert** is Associate Professor at the University of Toulouse ('Université Paul Sabatier', University of Sciences) and member of the Information Retrieval - Information Mining and Visualisation group of the IRIT research centre ('Institut de Recherche en Informatique de Toulouse') of Toulouse, France. He specialises in Information Retrieval, XML retrieval, Ontologies, Profile management, and Databases.

**Carine Hue** received the M.Sc. degree in mathematics and computer science in 1999 and the PhD degree in signal and image processing in 2003 from the University of Rennes 1, France. From 2003 to 2005, she worked at INRA (French National Institute for Research in Agronomy) as full-time researcher on Bayesian statistical modeling for agronomy. She is currently researcher in the "PROFiling and datamining" research group of France Telecom R&D. Her research interests are statistical modeling and the application of data mining algorithms to various domain as signal and image processing, agronomy or business intelligence.

**Fabien Jourdan** received the PhD degree in 2004 from the University of Montpellier 2 in the topic of network visualization. He has been a researcher in the INRA (french National Institute on Agricultural Research) since 2005. His current research interests are on the visual and structural analysis of biological networks.

**Stéphane Lallich** is currently Professor of Computer Science at the University of Lyon, with ERIC Laboratory. His main research interests are the statistical aspects of knowledge discovery in database, especially the properties of measures, learning from rules, and ensemble methods.

**Quyet Thang Le** has been a lecturer at Cantho University since 1977. He earned his PhD degree on Statistics in 1988 at University of Paris d'Orsay. He was one of the founders of the College of Information and Communication Technology of Cantho University in 1994. His research interests fall into the domain of Prediction and Simulation Modeling. At present, he is chairman of the national project PreSimDA (Prediction and Simulation for Decisions in Agriculture).

**Mustapha Lebbah** is currently Associate Professor at the University of Paris 13 and a member of Machine Learning Team (A3), LIPN. His main researches are centered on machine learning (Self-organizing maps, Probabilistic and Statistic). He is graduated from USTO University where he received his engineer diploma in 1998. Thereafter, he gained an MSC (DEA) in Artificial Intelligence from the Paris 13 University in 1999. In 2003, after three years in RENAULT, he received his PhD degree in Computer Science from the University of Versailles.

**Sébastien Lefèvre** joined in 2003 the University of Strasbourg where he is currently Associate Professor of Computer Science. He received his M.Sc. and Eng. degrees from the University of Technology of Compiègne in 1999, his PhD from the University of Tours in 2002 and his Habilitation from the University of Strasbourg in 2009. Within the Image Science, Computer Science and Remote Sensing Laboratory (LSIIT), his research interests are related to image analysis using mainly mathematical morphology and machine learning with applications in space and earth observation, and content-based image retrieval. He is currently invited scientist at INRIA Rennes to study applications of mathematical morphology to the field of multimedia indexing.

**Alain Lelu** received a PhD in Statistics from the Paris 6 University, and teached information science in several french universities. He is now an associate professor at the LORIA laboratory (Nancy, France). He published mainly on neural models and clustering methods applied to information science and text mining. His current research interests include robust and graph-based methods for data mining and recommender systems.

**Philippe Lenca** is Associate Professor of Computer Science at Telecom Bretagne, a French prestigious graduate engineering school and international research centre in the field of information technologies. He received his PhD in Computer Science from the University of Rennes in 1997 and his Habilitation from the University of Lyon in 2007. He his an editorial board member of the Revue d'Intelligence Artificielle. He is a member of the CNRS laboratory LabSTICC (Information and Communication Science and Technology Laboratory) where he leads the DECIDE (DECision aId and knowleDge discovEry) team. His main research interests are knowledge discovery in database, especially interestingness measures, rule-based classification, ensemble methods and decision aiding.

**Alban Mancheron** is an assistant researcher in Computer Sciences at the LIRMM-CNRS (CNRS is the French National Scientific Research Center), from 2008 to 2009. Its research focuses on comparative genomics. He received his PhD in Computer Sciences (bioinformatics) with highest honor from the University of Nantes (LINA-CNRS), in 2006. He was an Assistant professor at the University of French West Indies and French Guiana (GRIMAAG lab), where its research focuses on data mining, from 2005 to 2007. Hereafter, he was a postdoc fellow at the LIFL-INRIA Lille-Nord Europe Center (INRIA is the French National Institute for Research in Computer Sciences and Control) from 2007 to 2008, where its research focuses both on bioinformatics and data mining.

**Alice Marascu** is currently a Postdoctoral Research Fellow in the DREAM Project hosted by Dr. Marie-Odile Cordier and Dr. René Quiniou at IRISA/ INRIA Rennes. Alice Marascu obtained a PhD degree in Computer Science from the University of Nice-Sophia Antipolis, France, in 2009 under the

supervision of Dr. Yves Lechevallier. The subject of her PhD thesis was "Sequential Pattern Mining from Data Streams" and the research work was conducted in the AxIS Research Team at INRIA Sophia Antipolis. Alice Marascu has two Master's degrees, from the Ecole Polytechnique de Nice-Sophia Antipolis and from the West University of Timisoara, Romania. Her main research interests include sequential pattern mining and data streams.

**Florent Masseglia** is currently a researcher for INRIA. He did research work in the Data Mining Group at the LIRMM (Montpellier , France) from 1998 to 2002 and received a PhD in computer science from Versailles University, France in 2002. His research interests include data mining, data streams and databases. He is co-editor of two books and three journals special issues on data mining. He is reviewer for a dozen of major international journals.

**Annie Morin** got a PhD degree from the University of Rennes in 1989. She is associate professor in the Computer Science Department of the University of Rennes, France. Her areas of expertise include data analysis, clustering and classification, and text mining. More precisely, she works on image and text retrieval using factorial analysis methods. She is on the program committee of several international conferences and manages several cooperation programs with Japan, Croatia (University of Zagreb) and Slovenia (University of Ljubljana) on multimedia (texts or/and images) indexing.

**Alina Dia Miron** has obtained her PhD degree in December 2009, from the School of Computer Science at the Joseph Fourier University of Grenoble, for her work on spatio-temporal ontologies for the Geospatial Semantic Web. She joined the Laboratory of Informatics in Grenoble (LIG, formerly called LSR-IMAG Laboratory) in 2005. Her research interests include spatial and temporal knowledge representation and reasoning techniques for the Semantic Web, ontology engineering, spatio-temporal semantic analysis, Geospatial Semantic Web.

**Nicolas S. Müller** is currently a PhD student at the Department of sociology and the Department of econometrics of the University of Geneva. He holds a MA in sociology and a MSc in Information Systems. His PhD subject is about the links between life trajectories and health outcomes. He is interested in the application of data mining methods in social sciences, and especially sequence mining and association rules methods.

**Nguyen-Khang Pham** received his MSc degree in computer science from the Institute of French Speaking for Informatics, Hanoi, Vietnam in 2005. He is currently a PhD student at IRISA, France. Nguyen-Khang Pham is a lecturer at the College of Information Technology of Cantho University, Vietnam where he teaches image processing. His current research focuses on multimedia (texts and/or images) indexing and retrieval using correspondence analysis.

**Pascal Poncelet** Pascal Poncelet is Professor at University of Montpellier 2, and the head of the TATOO Team in the LIRMM Laboratory, France. He was a Professor and the Head of the Data Mining Group in the Computer Science Department at Ecole des Mines d'Alès. He is currently interested in various techniques of data mining with application in Stream Mining, Web Mining and Text Mining. He has published a large number of research papers in refereed journals, conference, and workshops, and been reviewer for some leading academic journals.

**Chantal Reynaud** is Professor of Computer Science in the Laboratory of Computer Science (LRI) at the University of Paris-Sud. Her areas of research are Ontology Engineering and Information Integration. In particular, she works on the following topics: Information extraction from semi-structured data (e.g. XML documents), mappings between ontologies, discovery of mappings in peer to peer data management systems, ontology evolution. She is involved in several projects combining artificial intelligence and database techniques for information integration. She is the head of the Artificial Intelligence and Inference Systems Group in LRI and member of the INRIA-Saclay ële-de-France group called Gemo. She is the author of more than 70 refereed journal articles and conference papers.

**Gilbert Ritschard** (see About the editors on page 335)

**Brigitte Safar** is an Assistant Professor of Computer Science at the University of Paris-Sud 11. She is member of the Artificial Intelligence and Inference Systems (IASI) Group in the Laboratory of Computer Science (LRI) and of the INRIA-Saclay group called Gemo. Her areas of research are Knowledge Representation, Information Integration, and the Semantic Web. In particular, she works on the following topics: logic-based mediation between distributed data sources, representation of ontology, cooperative query answering using ontology (query relaxation and refinement), ontology mapping.

**Paolo Simonetto** received the Master's degree from the University of Padua (Italy) in 2007. He is currently a PhD student at the University of Bordeaux I (France) in the Department of Computer Science. His interests include graph visualization and graph clustering, in particular applied on biological data.

**Goverdhan Singh** did his Bachelor of Technology degree in Computer Science and Engineering from Indian Institute of Technology Guwahati. This work was done during his internship in the AxIS project-team of Inria. He is specifically interested in data mining, computer networks and information security. He is now working as a software engineer in Qwest Telecom Software Services, India.

**Matthias Studer** (see About the manuscript coordinator on page 336)

**Jean-Emile Symphor** is professor in computer science at the university of French west indies and French Guiana. He is also affiliated to the CEREGMIA laboratory (Center for research and studies in economics, marketing, modelisation and applied computer science). He received his PhD degree in Computer Science from Montpellier 2 University, France. His current research interests are knowledge discovery, database systems, datastreams, marketing, medical and spatial datamining. He has published research papers in refereed journals, conference, and workshops.

**Olivier Teste** is associate professor at the University of Toulouse and researcher in the Generalised Information Systems/Data Warehouse research group (SIG/ED) of the IRIT research centre ('Institut de Recherche en Informatique de Toulouse') of Toulouse, France. His research interests include all aspects related to data warehousing, more precisely multidimensional modelling and OLAP querying.

**François Trousset** is Assistant Professor at the French Engineer School: Ecole des Mines d'Alès France. He received his PhD degree in Computer Science at the University of Besançon. His current main research interests concern secure computing while preserving privacy of data, knowledge discovery as well as decision making.

**Nischal Verma** did B.Tech in Computer Science & Engineering from Indian Institute of Technology Guwahati in 2009. This work was done during his internship in the Computer Department at Ecole des Mines d'Alès where he contributed to a research in databases of collaborative organizations. His final B.Tech project was the testing of softwares/projects using Microsoft software "CHESS".

**Marlène Villanova-Oliver** is an Assistant Professor at the University Pierre Mendès France of Grenoble, France, since 2003. In 1999, she received her MS degree in Computer Science from the University Joseph Fourier of Grenoble and the European Diploma of 3rd cycle in Management and Technology of Information Systems (MATIS). She received her PhD in 2002 from the National Polytechnic Institute of Grenoble (Grenoble INP). She is a member of the Laboratory of Informatics in Grenoble (LIG, formerly called LSR-IMAG Laboratory) since 1998. Her research interests include Representation and Inference of Spatio-Temporal Information, Ontologies and Knowledge Representation, Geographic Semantic Web, adaptability to user and context in Web-based Information Systems.

**Lionel Vinceslas** is currently a PhD Student at the university of the French West Indies and French Guiana (U.A.G.) and is affiliated to the CEREGMIA laboratory (Center for research and studies in economics, marketing, modelisation and applied computer science). His main areas of research include database systems, datamining and datastreams. In particular, he works on frequent pattern mining problems and medical datamining.

**Nicolas Voisine** was born in 1972. He received the Master's degree in Physics in 1996 and the PhD degree in signal and image processing in 2002 from the University of Rennes 1, France. Currently, he is a Researcher in the "Statistical Processing of Information" research group of France Telecom R&D. His main research interests include statistical data analysis and data mining to various domains as signal and image processing or data mining. He developed unsupervised and especially supervised classifiers such as Decision Trees, Random Forest and model averaging of classifiers.

**Haïfa Zargayouna** is an Assistant Professor in computer science at Paris 13 University. She joined the Knowledge Representation and Natural language (KRNL) team in September 2008. She holds a PhD from University of Paris-Sud. Her areas of research are ontology integration and evaluation.

# About the Editors

**Henri Briand** is an Emeritus professor at Polytech'Nantes (Polytechnic graduate School of Nantes University, France) and he is a member of the "KnOwledge and Decision" team (KOD) in the Nantes-Atlantic Laboratory of Computer Sciences (LINA, CNRS UMR 6241). He earned his PhD in 1983 from Paul Sabatier University located in Toulouse (France) and has over 100 publications in database systems and database mining. He was the head of the Computer Engineering Department at Polytechnic School of Nantes University. He was in charge of a research team in the data mining domain. He was responsible for the organization of the Data Mining Master in Nantes University.

**Fabrice Guillet** is an associate professor in Computer Science at Polytech'Nantes, the graduate engineering school of University of Nantes, and a member of the "KnOwledge and Decision" team (COD) in the LINA (CNRS UMR 6241) laboratory. He received a PhD degree in Computer Sciences in 1995 from the Ecole Nationale Superieure des Telecommunications de Bretagne, and his Habilitation in 2006 from the University of Nantes. He is a founder and current treasurer of the International French-speaking "Extraction et Gestion des Connaissances (EGC)" Association, and he has been also involved in the steering committee of the annual EGC French-speaking Conference since 2001. His research interests include knowledge quality and knowledge visualization in the frameworks of Data Mining and Knowledge Management. He has recently co-edited two refereed books of chapter entitled "Quality Measures in Data Mining" and "Statistical Implicative Analysis - Theory and Applications" published by Springer in 2007 and 2008.

**Gilbert Ritschard** is a full professor of statistics for the social sciences at the Department of econometrics of the University of Geneva where he is also a member of the chair board of the Laboratory of Demography and Family Studies. In addition, he acts as vice-dean of the Faculty of Economics and Social Sciences since 2007. He earned his PhD in econometrics and statistics

in Geneva in 1979 and taught as invited professor in Montreal, Fribourg, Toronto, Lyon, Lausanne and Los Angeles. He published over 100 papers in economics, statistics and data mining as well as on more applied topics in the field of social sciences, especially in sociology and social science history. His present research interests are in event history analysis and the application of KDD in the social sciences. He headed or co-headed several funded applied researches and leads presently an important project on methods for mining event histories in which he developed with his team the TraMineR toolbox for analysing sequence data in R.

**Djamel A. Zighed** is a full professeur in Computer Science at University Lumière Lyon 2 (France) where he heads the ERIC Laboratory specialized in Knowledge Engineering. He is the current president of the International French-speaking "Extraction et Gestion des Connaissances (EGC)" Association. He received his PhD in 1985 from the University of Lyon 1 and published in international journals as well as several books as author and editor. His research interests are in methodologies and tools for Mining Complex Data and he develops also research works on Topological Learning, a new hot topic in the area of machine learning. He is the coordinator of an Erasmus Mundus Master on Data Mining and Knowledge Management.

## About the Manuscript Coordinator

**Matthias Studer** is a PhD student at the Department of econometrics at the University of Geneva. He holds a Master degree in economics and a Master of Advanced Studies in sociology. He is currently working on gender and social inequalities at the beginning of academic careers in Switzerland. His research interests include data mining of longitudinal data such as state and event sequences, dissimilarity analysis and survival trees.

# Author Index


Aldea, Emanuel 77

Bennani, Younès 133
Bloch, Isabelle 77
Boullé, Marc 21
Bourqui, Romain 167

Cadot, Martine 307
Cerbah, Farid 271
Cleuziou, Guillaume 149
Clérot, Fabrice 181

Do, Thanh-Nghi 39

Fiot, Céline 217

Gabadinho, Alexis 3
Gabsi, Nesrine 181
Gensel, Jérôme 287
Gros, Patrick 57
Grozavu, Nistor 133

Hamdi, Fayçal 251
Hubert, Gilles 97
Hue, Carine 21
Hébrail, Georges 181

Jourdan, Fabien 167

Lallich, Stéphane 39
Le, Quyet-Thang 57
Lebbah, Mustapha 133
Lefèvre, Sébastien 113
Lelu, Alain 307
Lenca, Philippe 39

Mancheron, Alban 201
Marascu, Alice 217
Masseglia, Florent 217, 235
Miron, Alina Dia 287
Morin, Annie 57

Nicolas S. Müller 3

Pham, Nguyen-Khang 39, 57
Poncelet, Pascal 201, 217, 235

Reynaud, Chantal 251
Ritschard, Gilbert 3

Safar, Brigitte 251
Simonetto, Paolo 167
Singh, Goverdhan 217
Studer, Matthias 3
Symphor, Jean-Emile 201

Teste, Olivier 97
Trousset, François 235

Verma, Nischal 235
Villanova-Oliver, Marlène 287
Vinceslas, Lionel 201
Voisine, Nicolas 21

Zargayouna, Haïfa 251