Petra Perner (Ed.)

LNAI 6171

Advances in Data Mining

Applications and Theoretical Aspects

10th Industrial Conference, ICDM 2010
Berlin, Germany, July 2010
Proceedings

Springer

Lecture Notes in Artificial Intelligence 6171

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Petra Perner (Ed.)

Advances in Data Mining

Applications and Theoretical Aspects

10th Industrial Conference, ICDM 2010
Berlin, Germany, July 12-14, 2010
Proceedings

Springer

Series Editors

Randy Goebel, University of Alberta, Edmonton, Canada
Jörg Siekmann, University of Saarland, Saarbrücken, Germany
Wolfgang Wahlster, DFKI and University of Saarland, Saarbrücken, Germany

Volume Editor

Petra Perner
Institute of Computer Vision
and Applied Computer Sciences, IBaI
Kohlenstr. 2
04107 Leipzig, Germany
E-mail: pperner@ibai-institut.de

Library of Congress Control Number: 2010930175

CR Subject Classification (1998): I.2.6, I.2, H.2.8, J.3, H.3, I.4-5, J.1

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-642-14399-7 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-14399-1 Springer Berlin Heidelberg New York

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper 06/3180

Preface

These are the proceedings of the tenth event of the Industrial Conference on Data Mining ICDM held in Berlin (www.data-mining-forum.de).

For this edition the Program Committee received 175 submissions. After the peer-review process, we accepted 49 high-quality papers for oral presentation that are included in this book. The topics range from theoretical aspects of data mining to applications of data mining such as on multimedia data, in marketing, finance and telecommunication, in medicine and agriculture, and in process control, industry and society. Extended versions of selected papers will appear in the international journal *Transactions on Machine Learning and Data Mining* (www.ibai-publishing.org/journal/mldm).

Ten papers were selected for poster presentations and are published in the *ICDM Poster Proceeding* Volume by *ibai-publishing* (www.ibai-publishing.org).

In conjunction with ICDM four workshops were held on special hot application-oriented topics in data mining: Data Mining in Marketing DMM, Data Mining in LifeScience DMLS, the Workshop on Case-Based Reasoning for Multimedia Data CBR-MD, and the Workshop on Data Mining in Agriculture DMA. The Workshop on Data Mining in Agriculture ran for the first time this year. All workshop papers will be published in the *workshop proceedings* by ibai-publishing (www.ibai-publishing.org). Selected papers of CBR-MD will be published in a special issue of the international journal *Transactions on Case-Based Reasoning* (www.ibai-publishing.org/journal/cbr).

We were pleased to give out the best paper award for ICDM again this year. The final decision was made by the Best Paper Award Committee based on the presentation by the authors and the discussion with the auditorium. The ceremony took place at the end of the conference. This prize is sponsored by ibai solutions—www.ibai-solutions.de—one of the leading data mining companies in data mining for marketing, Web mining and E-Commerce.

The conference was rounded up by an outlook on new challenging topics in data mining before the Best Paper Award Ceremony.

We thank the members of the Institute of Applied Computer Sciences, Leipzig, Germany (www.ibai-institut.de) who handled the conference as secretariat. We appreciate the help and understanding of the editorial staff at Springer, and in particular Alfred Hofmann, who supported the publication of these proceedings in the LNAI series.

Last, but not least, we wish to thank all the speakers and participants who contributed to the success of the conference. The next conference in the series will be held in 2011 in New York during the world congress "The Frontiers in Intelligent Data and Signal Analysis, DSA2011" (www.worldcongressdsa.com) that brings together the

International Conferences on Machine Learning and Data Mining (MLDM), the Industrial Conference on Data Mining (ICDM), and the International Conference on Mass Data Analysis of Signals and Images in Medicine, Biotechnology, Chemistry and Food Industry (MDA).

July 2010 Petra Perner

Industrial Conference on Data Mining, ICDM 2010

Chair

Petra Perner	IBaI Leipzig, Germany

Program Committee

Klaus-Peter Adlassnig	Medical University of Vienna, Austria
Andrea Ahlemeyer-Stubbe	ENBIS, The Netherlands
Klaus-Dieter Althoff	University of Hildesheim, Germany
Chid Apte	IBM Yorktown Heights, USA
Eva Armengol	IIA CSIC, Spain
Bart Baesens	KU Leuven, Belgium
Isabelle Bichindaritz	University of Washington, USA
Leon Bobrowski	Bialystok Technical University, Poland
Marc Boullé	France Télécom, France
Henning Christiansen	Roskilde University, Denmark
Shirley Coleman	University of Newcastle, UK
Juan M. Corchado	Universidad de Salamanca, Spain
Antonio Dourado	University of Coimbra, Portugal
Peter Funk	Mälardalen University, Sweden
Brent Gordon	NASA Goddard Space Flight Center, USA
Gary F. Holness	Quantum Leap Innovations Inc., USA
Eyke Hüllermeier	University of Marburg, Germany
Piotr Jedrzejowicz	Gdynia Maritime University, Poland
Janusz Kacprzyk	Polish Academy of Sciences, Poland
Mehmed Kantardzic	University of Louisville, USA
Ron Kenett	KPA Ltd., Israel
Mineichi Kudo	Hokkaido University, Japan
David Manzano Macho	Ericsson Research Spain, Spain
Eduardo F. Morales	INAOE, Ciencias Computacionales, Mexico
Stefania Montani	Università del Piemonte Orientale, Italy
Jerry Oglesby	SAS Institute Inc., USA
Eric Pauwels	CWI Utrecht, The Netherlands
Mykola Pechenizkiy	Eindhoven University of Technology, The Netherlands
Ashwin Ram	Georgia Institute of Technology, USA
Tim Rey	Dow Chemical Company, USA
Rainer Schmidt	University of Rostock, Germany
Yuval Shahar	Ben Gurion University, Israel
David Taniar	Monash University, Australia

Stijn Viaene	KU Leuven, Belgium
Rob A. Vingerhoeds	Ecole Nationale d'Ingénieurs de Tarbes, France
Yanbo J. Wang	Information Management Center, China Minsheng Banking Corporation Ltd., China
Claus Weihs	University of Dortmund, Germany
Terry Windeatt	University of Surrey, UK

Table of Contents

Invited Talk

Theoretical Aspects of Data Mining

Multimedia Data Mining

Data Mining in Marketing

Data Mining in Industrial Processes

Data Mining in Medicine

Data Mining in Agriculture

WebMining

Data Mining in Finance

Aspects of Data Mining

Data Mining for Network Performance Monitoring

Moving Targets

When Data Classes Depend on Subjective Judgement, or They Are Crafted by an Adversary to Mislead Pattern Analysis Algorithms - The Cases of Content Based Image Retrieval and Adversarial Classification

Giorgio Giacinto

Dip. Ing. Elettrica ed Elettronica - Università di Cagliari, Italy
giacinto@diee.unica.it

Abstract. The vast majority of pattern recognition applications assume that data can be subdivided into a number of data classes on the basis of the values of a set of suitable features. Supervised techniques assume the data classes are given in advance, and the goal is to find the most suitable set of feature and classification algorithm that allows the effective partition of the data. On the other hand, unsupervised techniques allow discovering the "natural" data classes in which data can be partitioned, for a given set of features.These approaches are showing their limitation to handle the challenges issued by applications where, for each instance of the problem, patterns can be assigned to different data classes, and the definition itself of data classes is not uniquely fixed. As a consequence, the set of features providing for an effective discrimination of patterns, and the related discrimination rule, should be set for each instance of the classification problem. Two applications from different domains share similar characteristics: Content-Based Multimedia Retrieval and Adversarial Classification. The retrieval of multimedia data by content is biased by the high subjectivity of the concept of similarity. On the other hand, in an adversarial environment, the adversary carefully craft new patterns so that they are assigned to the incorrect data class. In this paper, the issues of the two application scenarios will be discussed, and some effective solutions and future reearch directions will be outlined.

1 Introduction

Pattern Recognition aims at designing machines that can perform recognition activities typical of human beings [13]. During the history of pattern recognition, a number of achievements have been attained, thanks both to algorithmic development, and to the improvement of technologies. New sensors, the availability of computers with very large memory, and high computational speed, have clearly allowed the spread of pattern recognition implementations in everyday life [16]. The traditional applications of pattern recognition are typically related to problems whose definition is clearly pointed out. In particular, the patterns are clearly defined, as they can be real objects such as persons, cars, etc., whose

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 1–16, 2010.

characteristics are captured by cameras and other sensing devices. Patterns are also defined in terms of signals captured in living beings. or related to environmental condition captured on the earth or the atmosphere. Finally, patterns are also artificially created by humans to ease the recognition of specific objetcs. For example, bar codes have been introduced to uniquely identify objects by a rapid scan of a laser bean. All these applications share the ssumption that the object of recognition is well defined, as well as the data classes in which the patterns are to be classified.

In order to perform classification, measurable features must be extracted from the patterns aiming at discriminating among different classes. Very often the definition itself of the pattern recognition task suggests some features that can be effectively used to perform the recognition. Sometimes, the features are extracted by understanding which process is undertaken by the human mind to perform such a task. As this process is very complex, because we barely don't know exactly how the human mind works, features are often extracted by formulating the problem directly at the machine level.

Pattern classifiers are based on statistical, stuctural or syntactic techniques, depending on the most suitable model of pattern represention for the task at hand. Very often, a classification problem can be solved using different approaches, the feasibility of each approach depending on the ease to extract the related features, and the discriminability power of each representation. Sometimes, a combination of multiple techniques is needed to attain the desired performances.

Nowadays, new challenging problems are facing the pattern recognition community. These problems are generated mainly by two causes. The first cause is the midespread use of computers connected via the Internet netwoek for a wide variety of tasks such as, personal communications, business, education, entertainment, etc. Vast part of our daily life relies on computers, and often large volumes of information are shared via social networks, blogs, web-sites, etc. The safety and security of our data is threathened in many ways by different subjects which may misuse our content, or stole our credentials to get access to bank accounts, credit cards, etc.

The second cause is the possibility for people to easily create, store, and share, vast amount of multimedia documents. Digital cameras allows capturing an unlimited number of photos and videos, thanks to the fact that they are also embedded in a number of portable devices. This vast amount of content needs to be organised, and effective search tools must be developed for these archives to be useful. It is easy to see that it is impractical to label the content of each image or different portions of videos. In addition, even if some label is added, they are subjective, and may not capture all the semantic content of the multimedia document.

Summing up, the safety and security of Internet communication requires the recognition of malicious activities performed by users, while effective techniques for the organization and retrieval of multimedia data requires the understanding of the semantic content. Why these two different tasks can be considered similar from the point of view of the theory of pattern recognition? In this paper, I will try to highlights the common challenges that this novel (and urgent) tasks pose to traditional pattern recognition theory, as well as to the broad area of "narrow" artificial intelligence, as the automatic solutions provided by artificial intelligence to some specific tasks are often referred to.

1.1 Challenges in Computer Security

The detection of computer attacks is actually one of the most challenging problems for three main reasons. One reason is related to the difficulty in predicting the behavior of software programs in response to *every* imput data. Software developers typically define the behavior of the program for legitimate input data, and design the bahavior of the program in the case the input data is not correct. However, in many cases it is a hard task to exactly define all possible incorrect cases. In addition, the complexity and the interoperability of different software programs make this task extremely difficult. It turns out that software always present weaknesses, a.k.a. *vulnerabilities*, which cause the software to exibit an unpredicted behavior in responde to some particular input data. The impact of the exploitation of these vulnerabilities often involves a large number of computers in a very short time frame. Thus, there is a huge effort in devising techniques able to detect never-seen-before attacks. The main problem is in the exact definition of the behavior that can be considered as being *normal* and which cannot. The vast majority of computers are general purpose computers. Thus, the user may run any kind of programs, at any time, in any combination. It turns out that the *normal* behaviour of one user is typically different to that of other users. In addition, new programs and services are rapidly created, so that the behavior of the same user changes over time. Finally, as soon as a number of measurable features are selected to define the *normal* behavior, attackers are able to craft their attacks so that it fits the typical feature of normal behavior.

The above discussion, clearly show that the *target* of attack detection task rapidly moves, as we have an attacker whose goal is to be undetected, so that each move made by the defender to secure the system can be made useless by a countermove made by the attacker. The rapid evolution of the computer scenario, and the fact that the speed of creation, and diffusion of attacks increases with the computing power of today machines, makes the detection problem quite hard [32].

1.2 Challenges in Content-Based Multimedia Retrieval

While in the former case, the computers are the source and the target of attacks, in this case we have the human in the loop. Digital pictures and videos capture the rich environment we experience everyday. It is quite easy to see that each picture and video may contain a large number of concepts depending on the level of detail used to describe the scene, or the focus in the description. Very often, one concept can be prevalent with respect to others, nevertheless this concept may be also decomposed in a number of "more simple" concepts. For example,

Table 1. Comparison between Intrusion Detection in Computer Systems and Content Based Multimedia Retrieval

	Intrusion Detection in Computer Systems	Content Based Multimedia Retrieval
Data Classes	The definition of the *normal* behavior depends on the Computer System at hand.	The definition of the conceptual data class(es) a given Multimedia object belongs to is highly subjective
Pattern	The definition of *pattern* is highly related to the *attacks* the computer system is subjected to	The definition of *pattern* is highly related to the *concepts* the user is focused to
Features	The *measures* used to characterise the patterns should be carefully chosen to avoid that attacks can be crafted to be a mimickry of normal behavior	The *low-level measures* used to characterise the patterns should be carefully chosen to suitably characterise the *high-level concepts*

an ad of a car can have additinal concepts, like the color of the car, the presence of humans or objects, etc. Thus, for a given image or video-shot, the same user may focus on different aspects. Moreover, if a large number of potential users are taken into acoount, the variety of concepts an image can bear is quite large. Sometimes the differences among concepts are subtle, or they can be related to shades of meaning. How can the task of retrieving *similar* images or videos from an archive can be solved by automatic procedures? How can we design automatic procedures that automatically tune the *similarity* measure to adapt to the visual concept the user is looking for? Once again, the *target* of the classification problem cannot be clearly defined beforehand.

1.3 Summary

Table 1 shows a synopsis of the above discussion, where the three main characteristics that make these two problems look-like similar are highlighted, as well as their differences. Computer security is affected by the so-called *adversarial* environment, where an adversary can gain enough knowledge on the classification/detection system that is used aither to mistrain the system, or to produce mimicry attacks [11,1,29,5]. Thus, in addition to the intrinsic difficulties of the problem that are related to the rapid evolution of design, type, and use of computer systems, a given attack may be performed in *apparently* different ways, as often the measures used for detection actually are not related to the most distingushing features. On the other hand, the user of a Multimedia classification and retrieval system cannot be modeled as an *adversary*. On the contrary, the user expects the system to respond to the query according to the concept in mind. Unfortunately, the system may appear to act as an *adversary*, by returning multimedia content which are not related with the user's goal, thus apparently *hiding* the contents of interest to the user [23].

The solutions to the above problems is far from being defined. However, some preliminary guidelines and directions can be given. Section 2 provides a brief overvuew of related works. A proposal for the design of pattern recognition systems for computer security and Multimedia Retrieval will be provided in Section 3. Section 4 will provide an example of experimental results related to the above applications where the guidelines have been used.

2 Related Works

In the field of computer security, very recently the concept of adversarial classification has been introduced [11,1,29,5]. The title of one of the seminal works on the topic is quite clear: "Can Machine Learning Be Secure?", pointing out the weaknesses of machine learning techniques with respect to an adversary that aims at evading or misleading the detection system. These works propose some statistical models that take into account the cost of the activities an adversary must take in order to evade or mislead the system. Thus, a system is robust against adversary actions as soon as the cost paid by the adversary is higher than the chances of getting an advantage. Among the proposed techniques that increase the costs of the actions of the adversary, the use of multiple features to represent the patterns, and the use of multiple learning algorithms provide solutions that not only make the task of adversary more difficult, but also may improve the detection abilties of the system [5]. Nonetheless, how to formulate the detection problem, extract suitable features, and select effective learning algorithms still remain a problem to be solved. Very recently, some papers addressed the problem of "moving targets" in the computer security community [21,31]. These papers address the problem of changes in the definition of *normal* behavior for a given system, and resort to techniques proposed in the framework of the so-called *concept drift* [34,14]. However, *concept-drift* may only partially provide a solution to the problem.

In the field of content based multimedia retrieval, a number of review papers pointed out the difficulties in provideing effective features and similarity measure that can cope with the broad domain of content of multimedia archives [30,19,12]. The shorcomings of current techniques developed for image and video has been clearly shown by Pavlidis [23]. While systems tailored for a particular set of images can exhibit quite impressive performances, the use of these systems on unconstrained domains reveal their inability to adapt dynamically to new concepts [28]. The solution is to have the user manually label a small set of representative images (the so-called *relevance feedback*), that are used as a training set for updating the similarity measure. However, how to implement relevance feedback to cope with multiple low-level representation of images, textual information, and additional information related to the images, is still an open problem [27]. In fact, while it is clear that the interpretation of an image made by humans takes into account multiple information contained in the image, as well as a number of concepts also related to cultural elements, the way all these elements can be represented and processed at the machine level has yet to be found.

We have already mentioned the theory of *concept drift* as a possible framework to cope with the two above problems [34,14]. The idea of concept drift arises in active learning, where as soon as new samples are collected, there is some context which is changing, and changes the characteristics of the patterns in itself. This kind of behavior can be seen also in computer systems, even if concept drift capture the phenomenon only partly [21,31]. On the other hand, in content based multimedia retrieval, the problem can be hardly formulated in terms of concept drift, as each multimedia content may actually bear multiple concepts. A different problem is the one of finding specific concepts in multimedia documents, such as a person, a car, etc. In this case, the concept of the pattern that is looked for may be actually drifted with respect to the original definition, so that it requires to be refined. This is a quite different problem from the one that is addressed here, i.e., the one of retrieving semantically similar multimedia documents.

Finally, ontologies have been introduced to describe hierarchies and interrelationships between concepts both in computer security and multimedia retrieval [17,15]. These approaches are suited to solve the problems of finding specific patterns, and provide complex reasoning mechanisms, while requiring the annotation of the objects.

3 Moving Targets in Computer Security

3.1 Intrusion Detection as a Pattern Recognition Task

The intrusion detection task is basically a pattern recognition task, where data must be assigned to one out of two classes: attack and legitimate activities. Classes can be further subdivided according to the IDS model employed. For the sake of the following discussion, we will refer to a two-class formulation, without losing generality.

The IDS design can be subdivided into the following steps:

1. **Data acquisition.** This step involves the choice of the data sources, and should be designed so that the captured data *allows* distinguishing as much as possible between attacks and legitimate activities.
2. **Data preprocessing.** Acquired data is processed so that patterns that do not belong to any of the classes of interest are deleted (noise removal), and incomplete patterns are discarded (enhancement).
3. **Feature selection.** This step aims at representing patterns in a feature space where the highest *discrimination* between legitimate and attack patterns is attained. A feature represents a measurable characteristic of the computer system's events (e.g. number of unsuccessful logins).
4. **Model selection.** In this step, using a set of example patterns (training set), a model achieving the best discrimination between legitimate and attack patterns is selected.
5. **Classification and result analysis.** This step performs the intrusion detection task, matching each test pattern to one of the classes (i.e. attack or

legitimate activity), according to the IDS model. Typically, in this step an alert is produced, either if the analyzed pattern matches the model of the attack class (misuse-based IDS), or if an analyzed pattern does *not* match the model of the legitimate activity class (anomaly-based IDS).

3.2 Intrusion Detection and Adversarial Environment: Key Points

The aim of a *skilled* adversary is to realize attacks without being detected by security administrators. This can be achieved by hiding the traces of attacks, thus allowing the attacker to work undisturbed, and by placing "access points" on violated computers for further stealthy criminal actions. In other terms, the IDS itself may be deliberately attacked by a skilled adversary. A *rational* attacker leverages on the weakest component of an IDS to compromise the reliability of the entire system, with minimum cost.

Data Acquisition. To perform intrusion detection, it is needed to acquire input data on events occurring on computer systems. In the *data acquisition* step these events are represented in a suitable way to be further analyzed. Some inaccuracy in the design of the representation of events will compromise the reliability of the results of further analysis, because an adversary can either exploit lacks of details in the representation of events, or induce a flawed event representation. Some inaccuracies may be addressed with an *a posteriori* analysis, that is, verifying what is actually occurring on monitored host(s) when an alert is generated.

Data pre-processing. This step is aimed at performing some kind of "noise removal" and "data enhancement" on data extracted in the data acquisition step, so that the resulting data exhibit a higher signal-to-noise ratio. In this context the noise can be defined as information that is not useful, or even counterproductive, when distinguishing between attacks and legitimate activities. On the other hand, enhancements typically take into account *a priori* information regarding the domain of the intrusion detection problem. As far as this stage is concerned, it is easy to see that critical information can be lost if we aim to remove all noisy patterns, or enhance all relevant events, as typically at this stage only a coarse analysis of low-level information can be performed. Thus, the goal of the data enhancement phase should be to remove those patterns which can be considered noisy with *high* confidence.

Feature extraction and selection. An adversary can affect both the feature definition and the feature extraction tasks. With reference to the feature definition task, an adversary can interfere with the process if this task has been designed to automatically define features from input data. With reference to the feature extraction tasks, the extraction of correct feature values depends on the tool used to process the collected data. An adversary may also inject patterns that are not representative of legitimate activity, but not necessarily related to attacks. These patterns can be included in the legitimate traffic flow that is used to verify the quality of extracted features. Thus, if patterns similar to attacks are injected in the legitimate traffic pool, the system may be forced to choose low quality features when minimizing the false alarm rate [24].

The effectiveness of the attack depends on the knowledge of the attacker on the algorithm used to define the “optimal” set of features, the better the knowledge, the more effective the attack. As “security through obscurity” is counterproductive, a possible solution is the definition of a large number of redundant features. Then, *random* subsets of features could be used at different times, provided that a good discrimination between attacks and legitimate activities in the reduced feature space is attained. In this way, an adversary is uncertain on the subset of features that is used in a certain time interval, and thus it can be *more difficult* to conceive *effective* malicious noise.

Model Selection. Different models can be selected to perform the same attack detection task, these models being either cooperative, or competitive. Again, the choice depends not only in the accuracy in attack detection, but also in the difficulty for an attackers to devise evasion techniques or alarm flooding attacks. As an example, very recently two papers from the same authors have been published in two security conferences, where program behavior has been modelled either by a graph structure, or by a statistical process for malware detection [6,2]. The two approaaches provide complementary solutions to similar problems, while leveraging on different features and differetn models.

However, no matter how the model has been selected, the adversary can use the knowledge on the selected model and on the training data to craft malicious patterns. However, this knowledge does not imply that the attacker is able to conceive *effective* malicious patterns. For example, a machine learning algorithm can be selected randomly from a predefined set [1]. As the malicious noise have to be well-crafted for a specific machine learning algorithm, the adversary cannot be sure of the attack success. Finally, when an off-line algorithm is employed, it is possible to randomly select the training patterns: in such a way the adversary is never able to know exactly the composition of the training set [10].

Classification and result analysis. To overstimulate or evade an IDS, a good knowledge of the features used by the IDS is necessary. Thus, if such a knowledge cannot be easily acquired, the impact can be reduced. This result can be attained for those cases in which a high-dimensional and possibly redundant set of features can be devised. Handling high-dimensional feature space typically require a feature selection step aimed at retaining a smaller subset of high discriminative features. In order to exploit all the available information carried out by a high-dimensional feature space, ensemble methods have been proposed, where a number of machine learning algorithms are trained on different feature subspace, and their results are then combined. These techniques improve the overall performances, and harden the evasion task, as the function that is implemented after combination in more complex than that produced by an individual machine learning algorithm [9,25]. A technique that should be further investigated to provide for additional hardness of evasion, and resilience to false alarm injection is based on the use of randomness [4]. Thus, even if the attacker has a perfect knowledge of the features extracted from data, and the learning algorithm employed, then in each time instant he cannot predict which subset of features is used. This can be possible by learning an ensemble of different machine learning algorithm on randomly selected subspaces of the entire feature set. Then, these different models can be randomly combined during the operational phase.

3.3 HMM-Web - Detection of Attacks against Web-Applicationa

As an example of an Intrusion Detection solutions designed according to the above guidelines, we provide an overview of HMM-Web, a host-based intrusion detection system capable to detect both simple and sophisticated input validation attacks against web applications [8]. This system exploits a sample of Web application queries to model normal (i.e. legitimate) queries to e web server. Attacks are detected as anomalous (not normal) web application queries. HMM-Web is made up of a set of application-specific modules (Figure 1). Each module is made up of an ensemble of Hidden Markov Models, trained on a set of normal queries issued to a specific web application. During the detection phase, each web application query is analysed by the corresponding module. A decision module classifies each analysed query as suspicious or legitimate according to the output of HMM. A different threshold is set for each application-specific module based on the confidence on the legitimacy of the set of training queries the proportion of training queries on the corresponding web application. Figure 2 shows the architecture of HMM-Web. Each query is made up of pairs <attribute,value> . The sequences of attributes is processed by a HMM ensemble, while each value is porcessed by a HMM tailored to the attribute it refers to. As the Figure shows,

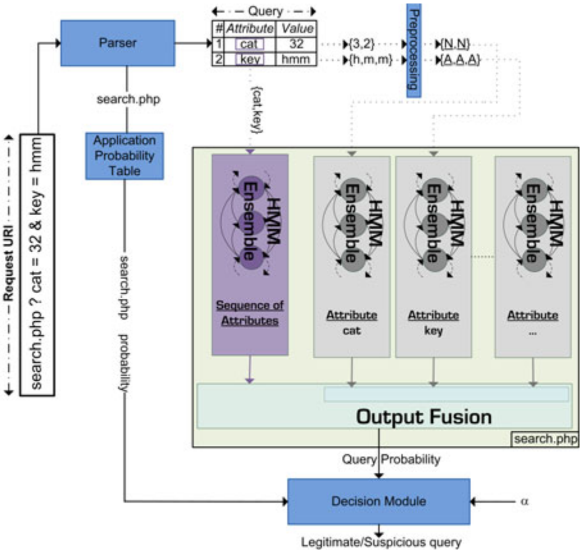


Fig. 1. Architecture of HMM-Web

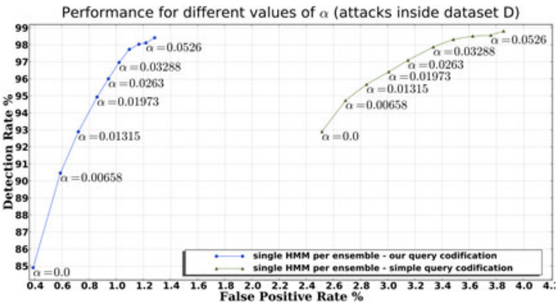


Fig. 2. Real-world dataset results. Comparison of the proposed encoding mechanism (left) with the one proposed in [18] (right). The value of α is the estimated proportion of attacks inside the training set.

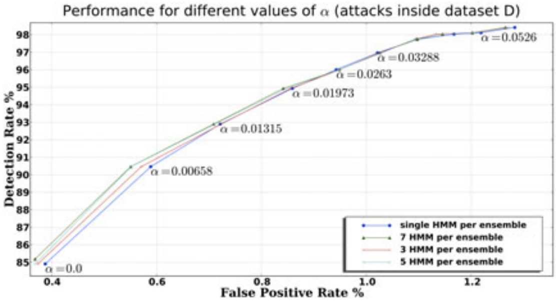


Fig. 3. Real-world dataset results. Comparison of different ensemble size. The value of α is the estimated proportion of attacks inside the training set.

two simbols ('A' and 'N') are used to represent all alphabetical characters and all numerical characters, respectively. All other characters are treated as different symbols. This encoding has been proven useful to enhance attack detection and increase the difficulty of evasion and overstimulation. Reported results in Figures 2 and 3 show the effectiveness of the encoding mechanism used, and the multiple classifier approach employed. In particular, the proposed system produce a good model of normal activities, as the rate of false alarms is quite low. In addition, Figure 2 also shows that HMM-Web outperformed another approach in the litearture [18].

4 Content Based Mutimedia Retrieval

The design of a content-based multimedia retrieval system requires a clear planning of the goal of the system. As much as the multimedia documents in the archive are of different types, are obtained by different acquisition techniques, and exhibit different content, the search for specific concepts is definitely a hard task. It is easy to see that as much as the scope of the system is limited, and the content to be searched is clearly defined, than the task can be managed by existing techniques. In the following, a short review of the basic choices a designer should make is presented, and references to the most recent literature are given. In addition, some results related to a proof of concept research tool are presented.

4.1 Scope of the Retrieval System

First of all, the scope of the system should be clearly defined. A number of content-based retrieval systems tailored for specific applications have been proposed to date. Some of the are related to sport events, as the playground is fixed, camera positions are known in advance, and the movements of the players and other objects (e.g., a ball) can be modeled [12]. Other applications are related to medical analysis, as the type of images, and the objects to look for can be precisely defined. On the other hand, tools for organizing personal photos on the PC, or to perform a search on large image and video repository are far from providing the expected perfromances. In addition, the large use of content sharing sites such as Flickr, YouTube, Facebook, etc., is creating very large repositories where the tasks of organising, searching, and controlling the use of the shared content, requires the development of new techniques. Basically, this is a matter of the numbers involved. While the answer to the question: *this archive contains documents with concept X?* may be fairly simple to be given, the answer to the question: *this document contains concept X?* is definitely harder. To answer the former question, a large number of false positives can be created, but a good system will also find the document of interest. However, this document may be confused in a large set of non-relevant documents. On the other hand, the latter request requires a complex reasoning system that is far from the state of the art.

4.2 Feature Extraction

The description of the content of a specific multimedia document can be provided in multiple ways. First of all, a document can be described in term of its properties provided in textual form (e.g., creator, content type, keywords, etc.). This is the model used in the so-called *Digital Libraries* where standard descriptors are defined, and guidelines for defining appropriate values are proposed. However, apart from descriptor such as the size of an image, the length of a video, etc., other keywords are typically given by a human expert. In the case of very narrow-domain systems, it is possible to agree on an ontology that helps describing *standard* scenarios. On the other hand, when multimedia content is shared on the web, different users may assign the same keyword to different contents, as well as assign different keywords to the same content. Thus, more complex ontologies, and reasoning systems are required to correctly assess the similarity among documents [3].

Multimedia content is also described by low-level and medium-level features [12,23]. These descriptions have been proposed by leveraging on the analogy that the human brain use these features to assess the similarity among visual contents. While at present this analogy is not deemed valid, these features may provide some additional hint about the concept represented by the pictorial content. Currently, very sophisticated low-level features are defined that take into account multiple image characteristics such as color, edge, texture, etc [7]. Indeed, as soon as the domain of the archive is narrow, very specific features can be computed that are directly linked with the semantic content [28]. On the other hand, in a broad domain archive, these feature may prove to be misleading, as the basic assumptions does not hold [23].

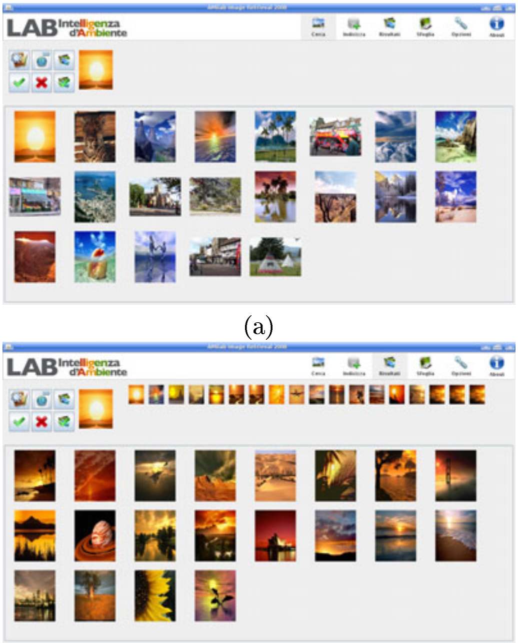
Finally, new features are emerging in the era of social networking. Aadditional information on the multimedia content is currently extracted from the text in the web pages containing the multimedia document, or in other web sites linked to the page of interest. Actually, the links between people sharing the images, and the comments that users posts on each other mutlimedia documents, provide a reach source of valuable information [20].

4.3 Similarity Models

For each feature description, a similarity measure is associated. On the other hand, when new application scenarios require the devlopment of new content descriptors, suitable similarity measures should be defined. This is the case of the exploitation of information from social networking sites: how this information can be suitably represented? Which is the most suitable measure to assess the influence of one user on other users? How we combine the information from social networks with other information on multimedia content? It is worth noting that the choice on the model used to weight different multimedia attributes and content descriptions, heavily affect the final performance of the system. On the other hand, the use of multiple representations may allow for a rich representation of content which the user may control towards feedback techniques.

4.4 The Human in the Loop

As there is no receipt to automatically capture the rich semantic content of multimedia data, except for some constrained problems, the human must be included in the process of cathegorisation and retrieval. The involvment can be implemented in a number of ways. Users typically provide tags that describe the multimedia content. They can provide impicit or explicit feedback, either by visiting the page containing a specific multimedia document in response to a given query, ot by expliciting reporting the relevance that the returned image exhibits with repect to the expected result [19]. Finally, they can provide explicit judgment on some challenge proposd by the system that helps learning the concept the user is looking for [33]. As we are not able to adquately model the human vision system, computers must rely on humans to perform complex tasks. On the other hand, computers may ease the task for human by providing a suitable visual organization of retrieval results, that allows a more effective user interaction [22].



(a)

(b)

Fig. 4. ImageHunter. (a) Initial query and retrieval results (b) retrieval results after three rounds of relevance feedback.

4.5 ImageHunter: A Prototype Content-Based Retrieval System

A large number of prototype or dimostrative systems have been proposed to date by the academia, and by computer companies[1]. ImageHunter is a proof of concept system designed in our Lab (Figure 4)[2]. This system performs visual query search on a database of images from which a number of low-level visual features are extracted (texture, color histograms, edge descriptors, etc.). Relevance feedback is implemented so that the user is allowed to mark both relevant and non-relevant images. The system implements a nearest-neighbor based learning systems which performs again the search by leveraging on the additional information available, and provides for suitable feature weighting [26]. While the results are encouraging, they are limited as the textual description is not taken into account. On the other hand these results clearly point out the need for the human in the loop, and the use of multiple features, that can be dynamically selected according to the user's feedback.

[1] An updated list can be found at `http://savvash.blogspot.com/2009/10/image-retrieval-systems.html`

[2] `http://prag.diee.unica.it/amilab/?q=video/imagehunter`

5 Conclusions

This paper aimed to provide a brief introduction on two challenging problem of the Internet Era. The Computer security problems, where humans leverage on the available computing power to misuse other computers, and the Content Retrieval tasks, where the humans would like to leverage on computing power to solve very complex reasoning tasks. Completely automatic learning solutions cannot be devised, as attacks as well as semantic concepts are conceived by human minds, and other human minds are needed to look for the needle in a haystack.

References

1. Barreno, M., Nelson, B., Sears, R., Joseph, A.D., Tygar, J.D.: Can machine learning be secure? In: ASIACCS 2006: Proceedings of the 2006 ACM Symposium on Information, computer and communications security, pp. 16–25. ACM, New York (2006)
2. Bayer, U., Comparetti, P., Hlauschek, C., Krügel, C., Kirda, E.: Scalable, behavior-based malware clustering. In: 16th Annual Network and Distributed System Security Symposium, NDSS 2009 (2009)
3. Bertini, M., Del Bimbo, A., Serra, G., Torniai, C., Cucchiara, R., Grana, C., Vezzani, R.: Dynamic pictorially enriched ontologies for digital video libraries. IEEE Multimedia 16(2), 42–51 (2009)
4. Biggio, B., Fumera, G., Roli, F.: Adversarial pattern classification using multiple classifiers and randomisation (2008)
5. Biggio, B., Fumera, G., Roli, F.: Multiple classifier systems for adversarial classification tasks. In: Benediktsson, J.A., Kittler, J., Roli, F. (eds.) MCS 2009. LNCS, vol. 5519, pp. 132–141. Springer, Heidelberg (2009)
6. Kruegel, C., Kirda, E., Zhou, X., Wang, X., Kolbitsch, C., Comparetti, P.: Effective and efficient malware detection at the end host. In: USENIX 2009 - Security Symposium (2009)
7. Chatzichristofis, S.A., Boutalis, Y.S.: Cedd: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. In: Gasteratos, A., Vincze, M., Tsotsos, J.K. (eds.) ICVS 2008. LNCS, vol. 5008, pp. 312–322. Springer, Heidelberg (2008)
8. Corona, I., Ariu, D., Giacinto, G.: Hmm-web: A framework for the detection of attacks against web applications. In: IEEE International Conference on Communications, ICC 2009, June 2009, pp. 1–6 (2009)
9. Corona, I., Giacinto, G., Mazzariello, C., Roli, F., Sansone, C.: Information fusion for computer security: State of the art and open issues. Inf. Fusion 10(4), 274–284 (2009)
10. Cretu, G.F., Stavrou, A., Locasto, M.E., Stolfo, S.J., Keromytis, A.D.: Casting out demons: Sanitizing training data for anomaly sensors. In: IEEE Symposium on Security and Privacy, SP 2008, May 2008, pp. 81–95 (2008)
11. Dalvi, N., Domingos, P., Mausam, Sanghai, S., Verma, D.: Adversarial classification. In: KDD 2004: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 99–108. ACM, New York (2004)

12. Datta, R., Joshi, D., Li, J., Wang, J.Z.: Image retrieval: Ideas, influences, and trends of the new age. ACM Comput. Surv. 40(2), 1–60 (2008)
13. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2000)
14. Harries, M.B., Sammut, C., Horn, K.: Extracting hidden context. Mach. Learn. 32(2), 101–126 (1998)
15. Kompatsiaris, Y., Hobson, P. (eds.): Semantic Multimedia and Ontologies - Theory and Applications. Springer, Heidelberg (2008)
16. Jain, A.K., Duin, R.P.W., Mao, J.: Statistical pattern recognition: a review. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(1), 4–37 (2000)
17. Joshi, A., Undercoffer, J., Pinkston, J.: Modeling computer attacks: An ontology for intrusion detection. In: Hartmanis, J., Goos, G., van Leeuwen, J. (eds.) RAID 2003. LNCS, vol. 2820, pp. 113–135. Springer, Heidelberg (2003)
18. Kruegel, C., Vigna, G., Robertson, W.: A multi-model approach to the detection of web-based attacks. Comput. Netw. 48(5), 717–738 (2005)
19. Lew, M.S., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: State of the art and challenges. ACM Trans. Multimedia Comput. Commun. Appl. 2(1), 1–19 (2006)
20. Li, X., Snoek, C.G.M., Worring, M.: Learning social tag relevance by neighbor voting. IEEE Transactions on Multimedia 11(7), 1310–1322 (2009)
21. Maggi, F., Robertson, W., Kruegel, C., Vigna, G.: Protecting a moving target: Addressing web application concept drift. In: RAID 2009: Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection, pp. 21–40. Springer, Heidelberg (2009)
22. Nguyen, G.P., Worring, M.: Interactive access to large image collections using similarity-based visualization. J. Vis. Lang. Comput. 19(2), 203–224 (2008)
23. Pavlidis, T.: Limitations of content-based image retrieval (October 2008)
24. Perdisci, R., Dagon, D., Lee, W., Fogla, P., Sharif, M.: Misleading worm signature generators using deliberate noise injection. In: 2006 IEEE Symposium on Security and Privacy, May, pp. 15–31 (2006)
25. Perdisci, R., Ariu, D., Fogla, P., Giacinto, G., Lee, W.: Mcpad: A multiple classifier system for accurate payload-based anomaly detection. Comput. Netw. 53(6), 864–881 (2009)
26. Piras, L., Giacinto, G.: Neighborhood-based feature weighting for relevance feedback in content-based retrieval. In: 10th Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS 2009, May 2009, pp. 238–241 (2009)
27. Richter, F., Romberg, S., Hörster, E., Lienhart, R.: Multimodal ranking for image search on community databases. In: MIR 2010: Proceedings of the international conference on Multimedia information retrieval, pp. 63–72. ACM, New York (2010)
28. Sivic, J., Zisserman, A.: Efficient visual search for objects in videos. Proceedings of the IEEE 96(4), 548–566 (2008)
29. Skillicorn, D.B.: Adversarial knowledge discovery. IEEE Intelligent Systems 24(6), 54–61 (2009)
30. 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)

31. Stavrou, A., Cretu-Ciocarlie, G.F., Locasto, M.E., Stolfo, S.J.: Keep your friends close: the necessity for updating an anomaly sensor with legitimate environment changes. In: AISec 2009: Proceedings of the 2nd ACM workshop on Security and artificial intelligence, pp. 39–46. ACM, New York (2009)
32. IBM Internet Security Systems. X-force® 2008 trend and risk report. Technical report, IBM (2009)
33. Thomee, B., Huiskes, M.J., Bakker, E., Lew, M.S.: Visual information retrieval using synthesized imagery. In: CIVR 2007: Proceedings of the 6th ACM international conference on Image and video retrieval, pp. 127–130. ACM, New York (2007)
34. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Mach. Learn. 23(1), 69–101 (1996)

Bioinformatics Contributions to Data Mining

Isabelle Bichindaritz

University of Washington, Institute of Technology / Computer Science and Systems
1900 Commerce Street, Box 358426
Tacoma, WA 98402, USA
ibichind@u.washington.edu

Abstract. The field of bioinformatics shows a tremendous growth at the crossroads of biology, medicine, information science, and computer science. Figures clearly demonstrate that today bioinformatics research is as productive as data mining research as a whole. However most bioinformatics research deals with tasks of prediction, classification, and tree or network induction from data. Bioinformatics tasks consist mainly in similarity-based sequence search, microarray data analysis, 2D or 3D macromolecule shape prediction, and phylogenetic classification. It is therefore interesting to consider how the methods of bioinformatics can be pertinent advances in data mining and to highlight some examples of how these bioinformatics algorithms can potentially be applied to domains outside biology.

Keywords: bioinformatics, feature selection, phylogenetic classification.

1 Introduction

Bioinformatics can be defined in short as the scientific discipline concerned with applying computer science to biology. Since biology belongs to the family of experimental sciences, generation of knowledge in biology derives from analyzing data gathered through experimental set-ups. Since the completion of the Human Genome Project in 2003 with the complete sequencing of the human genome [1], biological and genetic data have been accumulating and continue to be produced at an increasing rate. In order to make sense of these data, the classical methods developed in statistical data analysis and data mining have to adapt to the distinctive challenges presented in biology. By doing so, bioinformatics methods advance the research in data mining, to the point that today many of these methods would be advantageous when applied to solve problems outside of biology.

This article first reviews background information about bioinformatics and its challenges. Following, section three presents some of the main challenges for data mining in bioinformatics. Section four highlights two areas of progress originating from bioinformatics, feature selection for microarray data analysis and phylogenetic classification, and shows their applicability outside of biology. It is followed by the conclusion.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 17–27, 2010.

2 Bioinformatics and Its Challenges

Bioinformatics encompasses various meanings depending upon authors. Broadly speaking, bioinformatics can be considered as the discipline studying the applications of informatics to the medical, health, and biological sciences [2]. However, generally, researchers differentiate between medical informatics, health informatics, and bioinformatics. Bioinformatics is then restricted to the applications of informatics to such fields as genomics and the biosciences [2]. One of the most famous research projects in this field being the Human Genome Project, this paper adopts the definition of bioinformatics provided in the glossary of this project: "The science of managing and analyzing biological data using advanced computing techniques. Especially important in analyzing genomic research data" [1].

Among the biosciences, three main areas have benefitted the most from computational techniques: genomics, proteomics, and phylogenetics. The first field is devoted to "the study of genes and their functions" [1], the second to "the study of the full set of proteins encoded by a genome" [2], and the last one to the study of evolutionary trees, defined as "the basic structures necessary to think clearly about differences between species, and to analyze those differences statistically" [3].

Biosciences belong to the category of experimental sciences, which ground the knowledge they gain from experiences, and therefore collect data about natural phenomena. These data have been traditionally analyzed with statistics. Statistics as well as bioinformatics has several meanings. A classical definition of statistics is "the scientific study of data describing natural variation" [4]. Statistics generally studies populations or groups of individuals: "it deals with quantities of information, not with a single datum". Thus the measurement of a single animal or the response from a single biochemical test will generally not be of interest; unless a sample of animals is measured or several such tests are performed, statistics ordinarily can play no role [4]. Another main feature of statistics is that the data are generally numeric or quantifiable in some way. Statistics also refers to any computed or estimated statistical quantities such as the mean, mode, or standard deviation [4].

More recently, the science of data mining has emerged both as an alternative to statistical data analysis and as a complement. Finally, both fields have worked together more closely with the aim of solving common problems in a complementary attitude. This is particularly the case in biology and in bioinformatics.

The growing importance of bioinformatics and its unique role at the intersection of computer science, information science, and biology, motivate this article. In terms of computer science, forecasts for the development of the profession confirm a general trend to be "more and more infused by application areas". The emblematic application infused areas are health informatics and bioinformatics. For example the National Workforce Center for Emerging Technologies (NWCET) lists among such application areas healthcare informatics and global and public health informatics. It is also notable that the Science Citation Index (Institute for Scientific Information – ISI – Web of Knowledge) lists among computer science a specialty called "Computer science, Interdisciplinary applications". Moreover this area of computer science ranks the highest within the computer science discipline in terms of number of articles produced as well as in terms of total cites. These figures confirm the other data pointing toward the importance of applications in computer science. Among the journals within this category, many relate to bioinformatics or medical informatics journals. It is also noteworthy that some health informatics or bioinformatics journals are classified as well in other areas of computer science. In addition, the most cited new papers in computer science are frequently bioinformatics papers. For example, most of the papers referenced as "new hot papers" in computer science in 2008 have been bioinformatics papers.

This abundant research in bioinformatics, focused on major tasks in data mining such as prediction, classification, and network or tree mining, raises the question of how to integrate its advances within mainstream data mining, and how to apply its methods outside biology. Traditionally, researchers in data mining have identified several challenges to overcome for data miners to apply their analysis methods and algorithms to bioinformatics data. It is likely that it is around solutions to these challenges that major advances have been accomplished – as the rest of this paper will show.

3 Data Mining Challenges in Bioinformatics

Data mining applications in bioinformatics aim at carrying out tasks specific to biological domains, such as finding similarities between genetic sequences (sequence analysis); analyzing microarray data; predicting macromolecules shape in space from their sequence information (2D or 3D shape prediction); constructing evolutionary trees (phylogenetic classification), and more recently gene regulatory networks mining. The field has first attempted to apply well-known statistical and data mining techniques. However, researchers have quickly met with specific challenges to overcome, imposed by the tasks and data studied [5].

3.1 Sequence Searching

Researchers using genetic data frequently are interested in finding similar sequences. Given a particular sequence, for example newly discovered, they search online databases for similar known sequences, such as previously sequenced DNA segments, or genes, not only from humans, but also from varied organisms. For example, in drug design, they would like to know which protein would be encoded by a new sequence by matching it with similar sequences coding for proteins in the protein database SWISS-PROT. Examples of software developed for this task is the well-known BLAST ("Basic Local Alignment and Search Tool") available as a service from the National Center for Biotechnology Information (NCBI) (http://www.ncbi.nlm.nih.gov/blast/) [5]. Sophisticated methods have been developed for pair-wise sequence alignment and for multiple sequence alignments.

The main challenge here has been that two sequences are almost never identical. Consequently searches need to be based on similarity or analogy – and not on exact pattern-matching.

3.2 Microarray Data Analysis

One of the most studied bioinformatics applications to date remains the analysis of gene expression data from genomics. Gene expression is defined as the process by which a gene's DNA sequence is converted into a functional gene product, generally a protein [6]. To summarize, the genetic material of an individual is encoded in DNA. The process of gene expression comprises two major steps: translation and transcription. During translation, excerpts of the DNA sequence are first encoded as messenger RNA (mRNA). Following during transcription, the mRNA is transcribed into functional proteins [6]. Since all major genes in the human genome have been identified, measuring from a blood or tissue sample which of these has been expressed can provide a snapshot of the activity going on at the biological level in an organism. The array of expressed genes, called an expression profile, at a certain point in time and at a certain location, permits to characterize the biological state of an individual. The amount of an expression can be quantified by a real number – thus expression profiles are numeric. Among interesting questions studied in medical applications, are whether it is possible to diagnose a disease based on a patient's expression profile, or whether a certain subset of expressed genes can characterize a disease, or whether the severity of a disease can be predicted from expression profiles. Research has shown that for many diseases, these questions can be answered positively, and medical diagnosis and treatment can be enhanced by gene expression data analysis. Microarray technologies have been developed to measure expression profiles made of thousands of genes efficiently. Following microarray-based gene expression profiling can be used to identify subsets of genes whose expression changes in reaction to exposure to infectious organisms, various diseases or medical conditions, even in the intensive care unit (ICU). From a technical standpoint, a microarray consists in a single silicon chip capable of measuring the expression levels of thousands or tens of thousands of genes at once – enough to comprehend the entire human genome, estimated to around 25,000 genes, and even more [6]. Microarrays come in several different types, including short oligonucleotide arrays, cDNA or spotted arrays, long oligonucleotide arrays, and fiber-optic arrays. Short oligonucleotide arrays, manufactured by the company Affymetrix, are the most popular commercial variety on the market today [6]. See Fig. 1 for a pictorial representation of microarray data expressions.

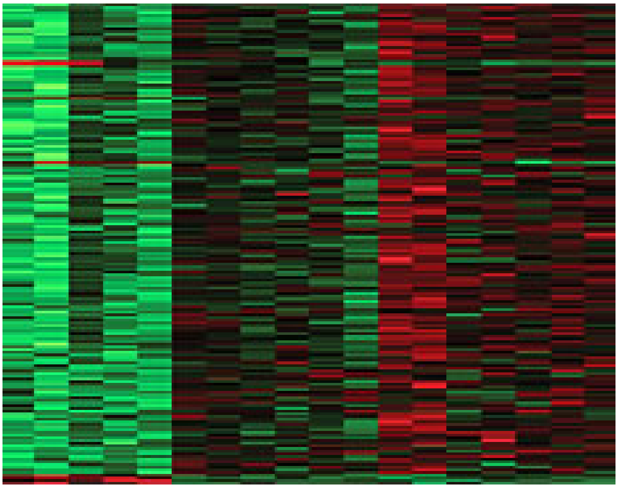


Fig. 1. A heatmap of microarray data

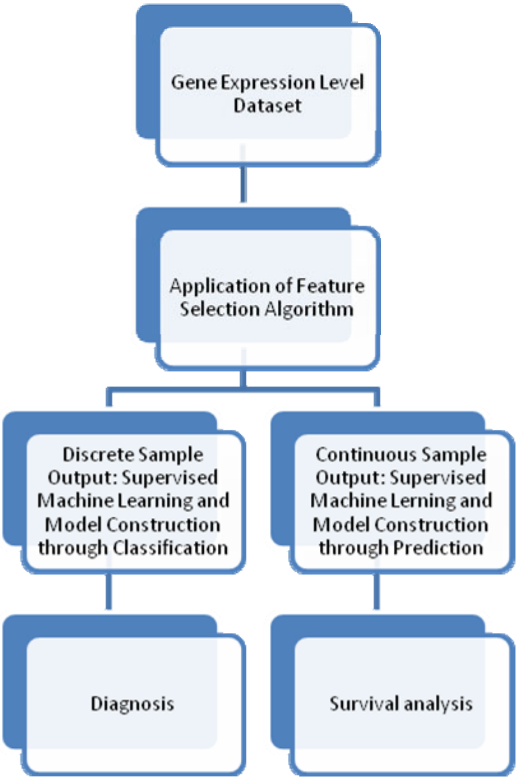

Gene Expression Level Dataset
Application of Feature Selection Algorithm
Discrete Sample Output: Supervised Machine Learning and Model Construction through Classification
Continuous Sample Output: Supervised Machine Lerning and Model Construction through Prediction
Diagnosis
Survival analysis

Fig. 2. Process-flow diagram illustrating the use of feature selection and supervised machine learning on gene expression data. Left branch indicates classification tasks, and right branch indicates prediction, with survival analysis as a special case.

Microarray data present a particular challenge for data miners, known as the curse of dimensionality. These datasets often comprise from tens to hundreds of samples or cases for thousands to tens of thousands of predictor genes. In this context, identifying a subset of genes the most connected with the outcome studied has been shown to provide better results – both in classification and in prediction. Therefore feature selection methods have been developed with the goal of selecting the smallest subset of genes providing the best classification or prediction. Similarly in survival analysis, genes selected through feature selection are then used to build a mathematical model that evaluates the continuous time to event data [7]. This model is further evaluated in terms of how well it predicts time to event. Actually, it is the combination of a feature selection algorithm and a particular model that is evaluated (see Fig. 2).

3.3 Phylogenetic Classification

The goal of phylogenetic classification is to construct cladograms (see Fig.3) following Hennig principles. Cladograms are rooted phylogenetic trees, where the root is the hypothetical common ancestor of the taxa, or groups of individuals in a class or species, in the tree.

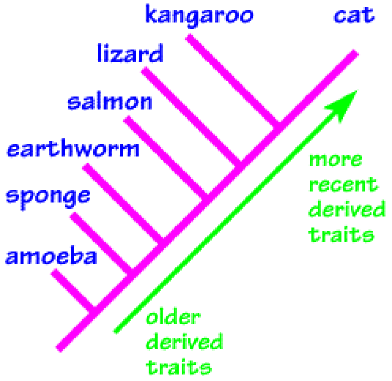


Fig. 3. A phylogenetic tree or cladogram

Methods in phyloinformatics aim at constructing phylogenetic classifications based on Hennig principles, starting from matrices of varied character values (see Fig. 4) – morphological and/or genetic and/or behavioral. There have been many attempts at constructing computerized solutions to solve the phylogenetic classification problem. The most widely spread methods are parsimony-based [8]. Another important method is compatibility.

The parsimony method attempts to minimize the number of character state changes among the taxa (the simplest evolutionary hypothesis) [9, 10]. The system PAUP [10], for Phylogenetic Analysis Using Parsimony, is classically used by phylogeneticists to induce classifications. It implements a numerical parsimony to calculate the tree that totals the least number of evolutionary steps. Swofford 2002 defines parsimony as the minimization of homoplasies [10]. Homoplasies are evolutionary mistakes. Examples are parallelism - apparition of the same derived character independently between two groups -, convergence - state obtained by the independent transformation of two characters -, or reversion - evolution of one character from a more derived state to a more primitive one -. Homoplasy is most commonly due to multiple independent origins of indistinguishable evolutionary novelties. Following this general methodic goal of minimizing the number of homoplasies defined as parsimony, a family of mathematical and statistical methods has emerged over time, such as:

- **FITCH method.** For unordered characters.
- **WAGNER method.** For ordered undirected characters.
- **CAMIN-SOKAL method.** For ordered undirected characters, it prevents reversion, but allows convergence and parallelism.
- **DOLLO method.** For ordered directed characters, it prevents convergence and parallelism, but not reversion.
- **Polymorphic method.** In chromosome inversion, it allows hypothetical ancestors to have polymorphic characters, which means that they can have several values.

All these methods are simplifications of Hennig principles, but have the advantage to lead to computationally tractable and efficient programs. The simplifications they are

	C0	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**
T1	1	0	0	0	1	0	1	1	0	0	0	0
T2	1	0	0	0	1	1	1	0	1	0	0	0
T3	1	1	1	1	0	1	1	0	0	1	0	0
T4	1	1	1	1	0	0	0	0	0	0	1	0
T5	1	0	0	1	0	0	0	0	0	0	0	1

Fig. 4. Sample taxon matrix. Rows represent taxa and columns characters. The presence of a character is indicated by a '1'.

based on are that parsimony is equivalent to Hennig principles, and that numeric parsimony perfectly handles the complexity of parsimony programs.

Compatibility methods [11, 12] aim at maximizing the number of characters mutually compatible on a cladogram. Compatible characters are ones that present no homoplasy, neither reversion, nor parallelism, nor convergence. The phylogenetic classification problem is solved here by the method of finding the largest clique of compatible characters, a clique being a set of characters presenting no homoplasy (see Fig. 5). These methods present the same advantages, and the same disadvantages, as the parsimony methods. First, their goal is similar to parsimony, since maximizing the number of characters not presenting homoplasy, is a problem equivalent to minimizing homoplasies. Following, these methods are also a simplification of Hennig principles, they are also numeric methods, but they lead to computationally tractable and efficient algorithms

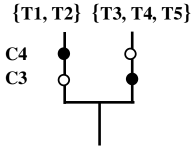


Fig. 5. Two monophyletic groups from two exclusive synapomorphies. Black dots represent presence of a character, while white dots represent its absence.

4 Contributions of Bioinformatics to Data Mining

For many years statistical data analysis and data mining methods have been applied to solving bioinformatics problems, and in particular its challenges. As a result the methods developed have expanded the traditional data analysis and mining methods, to the point that, today, many of these enhancements have surpassed the research continued outside of bioinformatics. Following these novel methods are becoming more and more applied to yet other application domains. Two examples will illustrate how these bioinformatics methods have enriched data analysis and data mining in general, such as in feature selection, or could be applied to solve problems outside of bioinformatics, such as in phylogenetic classification.

4.1 Feature Selection

As explained previously, classification or prediction systems in bioinformatics often include a feature selection step. One particular method having shown excellent results is Bayesian Model Averaging (BMA) feature selection. The strength of BMA lies in its ability to account for model uncertainty, an aspect of analysis that is largely ignored by traditional stepwise selection procedures [13]. These traditional methods tend to overestimate the goodness-of-fit between model and data, and the model is subsequently unable to retain its predictive power when applied to independent datasets [14]. BMA attempts to solve this problem by selecting a subset of all possible models and making statistical inferences using the weighted average of these models' posterior distributions.

In the application of classification or prediction, such as survival analysis, to high-dimensional microarray data, a feature selection algorithm identifies this subset of genes from the gene expression dataset. These genes are then used to build a mathematical model that evaluates either the class or the continuous time to event data. The choice of feature selection algorithm determines which genes are chosen and the number of predictor genes deemed to be relevant, whereas the choice of mathematical framework used in model construction dictates the ultimate success of the model in predicting a class or the time to event on a validation dataset. See Fig. 2 for a process-flow diagram delineating the application of feature selection and supervised machine learning to gene expression data – left branch illustrates classification tasks, and right branch illustrates prediction tasks such as survival analysis.

The problem with most feature selection algorithms used to produce continuous predictors of patient survival is that they fail to account for model uncertainty. With thousands of genes and only tens to hundreds of samples, there is a relatively high likelihood that a number of different models could describe the data with equal predictive power. Bayesian Model Averaging (BMA) methods [13, 15] have been applied to selecting a subset of genes on microarray data. Instead of choosing a single model and proceeding as if the data was actually generated from it, BMA combines the effectiveness of multiple models by taking the weighted average of their posterior distributions. In addition, BMA consistently identifies a small number of predictive genes [14, 16], and the posterior probabilities of the selected genes and models are available to facilitate an easily interpretable summary of the output. Yeung et al. 2005 extended the applicability of the traditional BMA algorithm to high-dimensional microarray data by embedding the BMA framework within an iterative approach [16].

Following their iterative BMA method has further been extended to survival analysis. Survival analysis is a statistical task aiming at predicting time to event information. In general the event is death or relapse. This task is a variant of a prediction task, dealing with continuous numeric data in the class label (see Fig. 2). However a distinction has to be made between patients leaving the study for unrelated causes (such as end of the study) – these are called censored cases - and for cause related to the event. In particular in cancer research, survival analysis can be applied to gene expression profiles to predict the time to metastasis, death, or relapse. Feature selection methods are combined with statistical model construction to predict survival analysis. In the context of survival analysis, a *model* refers to a set of selected genes whose regression coefficients have been calculated for use in predicting survival prognosis [7, 17]. In particular, the iterative BMA method for survival analysis has been developed and implemented as a Bioconductor package, and the algorithm is demonstrated on two real cancer datasets. The results reveal that BMA presents with greater predictive accuracy than other algorithms while using a comparable or smaller number of genes, and the models themselves are simple and highly amenable to biological interpretation. Annest et al. 2009 [7] applied the same BMA method to survival analysis with excellent results as well. The advantage of resorting to BMA is to not only select features but also learn feature weights useful in similarity evaluation.

These examples show how a statistical data analysis method, BMA, had to be extended with an iterative approach to be applied to microarray data. In addition, an extension to survival analysis was completed and several statistical packages were created, which could be applied to domains outside biology in the future.

4.2 Phylogenetic Classification

Phylogenetic classification can be applied to tasks involving discovering the evolution of a group of individuals or objects and to build an evolutionary tree from the

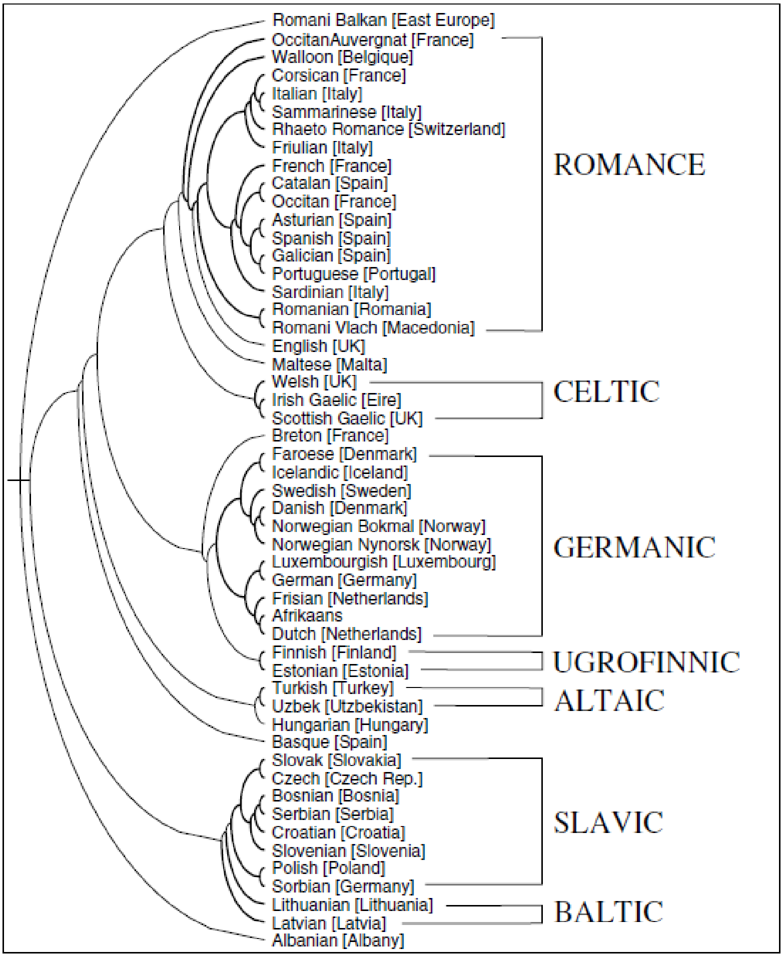


Fig. 6. Language Tree: This figure illustrates the phylogenetic-like tree constructed on the basis of more than 50 different versions of “The Universal Declaration of Human Rights” [19]

characteristics of the different objects or individuals. Courses in phylogenetic classification often teach how to apply these methods to domains outside of biology or within biology for other purposes than species classification and building the tree of life. Examples of applications of cladograms (see Fig. 3) are explaining the history and evolution of cultural artifacts in archeology, for example paleoindian projectile-points [18], comparing and grouping languages in families in linguistics [19] (see Fig. 6), or tracing the chronology of documents copied multiple times in textual criticism. Recently, important differences have been stressed between the natural evolution at work in nature and human-directed evolution [19]. Phylogenetic trees represent the evolutions of populations, while in examples from other domains classify individuals [20]. In addition, applications are often interested in finding explanations for what is observed, while in evolution, it is mostly classification that is of interest [20]. Nevertheless, researchers who have used phylogenetic classification in other domains have published their findings because they found them interesting: “The Darwinian mechanisms of selection and transmission, when incorporated into an explanatory theory, provide precisely what culture historians were looking for: the tools to begin explaining cultural lineages—that is, to answer why-type questions” [18]. Although the application of phylogenetic classification outside of biology is relatively new, it is destined to expand. For example, we could think of tracing the history and evolution of cooking recipes, or of ideas in a particular domain, for example in philosophy.

Interestingly, the methods developed for phylogenetic classification are quite different from the data mining methods building dendrograms – these do not take history or evolution through time into account. These methods have proved not adapted to phylogenetic classification, therefore the building of cladograms brings a very rich set of methods to build them that do not have equivalents in data mining.

5 Conclusion

In this paper, we have highlighted the richness and specificity of some of the major bioinformatics methods, and defended the idea that these methods can be very useful to domains outside of biology – and even to biological tasks different from those they were originally developed for. In this sense, bioinformatics has much to bring to the field of data mining in general. The near future will likely see many more applications of bioinformatics methods outside of biology. We have presented the examples of feature selection from microarray data analysis and of phylogenetic classification. Similarly sequence searching could be applied to information search, protein 2D or 3D shape reconstruction to information visualization and storage, and regulatory network mining to the Internet. The possibilities are really endless.

References

[1] DOE Human Genome Project. Genome Glossary, http://www.ornl.gov/sci/techresources/Human_Genome/glossary/glossary_b.shtml (accessed April 22, 2010)

[2] Miller, P.: Opportunities at the Intersection of Bioinformatics and Health Informatics: A Case Study. Journal of the American Medical Informatics Association 7(5), 431–438 (2000)

[3] Felsenstein, J.: Inferring Phylogenies. Sinauer Associates, Inc., Sunderland (2004)

[4] Sokal, R.R., Rohlf, F.J.: Biometry. The Principles and Practice of Statistics in Biological Research. W.H. Freeman and Company, New York (2001)

[5] Kuonen, D.: Challenges in Bioinformatics for Statistical Data Miners. Bulletin of the Swiss Statistical Society 46, 10–17 (2003)

[6] Piatetsky-Shapiro, G., Tamayo, P.: Microarray Data Mining: Facing the Challenges. ACM SIGKDD Explorations Newsletter 5(2), 1–5 (2003)

[7] Annest, A., Bumgarner, R.E., Raftery, A.E., Yeung, K.Y.: Iterative Bayesian Model Averaging: a method for the application of survival analysis to high-dimensional microarray data. BMC Bioinformatics 10, 10–72 (2009)

[8] Felsenstein, J.: The troubled growth of statistical phylogenetics. Systematic-Biology 50(4), 465–467 (2001)

[9] Maddison, W.P., Maddison, D.R.: MacClade: analysis of phylogeny and character evolution. Version 3.0. Sinauer Associates, Sunderland (1992)

[10] Swofford, D.L.: PAUP: Phylogenetic Analysis Using Parcimony. Version 4. Sinauer Associates Inc. (2002)

[11] Martins, E.P., Diniz-Filho, J.A., Housworth, E.A.: Adaptation and the comparative method: A computer simulation study. Evolution 56, 1–13 (2002)

[12] Meacham, C.A.: A manual method for character compatibility analysis. Taxon 30(3), 591–600 (1981)

[13] Raftery, A.: Bayesian Model Selection in Social Research (with Discussion). In: Marsden, P. (ed.) Sociological Methodology 1995, pp. 111–196. Blackwell, Cambridge (1995)

[14] Volinsky, C., Madigan, D., Raftery, A., Kronmal, R.: Bayesian Model Averaging in Proprtional Hazard Models: Assessing the Risk of a Stroke. Applied Statistics 46(4), 433–448 (1997)

[15] Hoeting, J., Madigan, D., Raftery, A., Volinsky, C.: Bayesian Model Averaging: A Tutorial. Statistical Science 14(4), 382–417 (1999)

[16] Yeung, K., Bumgarner, R., Raftery, A.: Bayesian Model Averaging: Development of an Improved Multi-Class, Gene Selection and Classification Tool for Microarray Data. Bioinformatics 21(10), 2394–2402 (2005)

[17] Hosmer, D., Lemeshow, S., May, S.: Applied Survival Analysis: Regression Modeling of Time to Event Data, 2nd edn. Wiley Series in Probability and Statistics. Wiley Interscience, Hoboken (2008)

[18] O'Brien, M.J., Lyman, R.L.: Evolutionary Archaeology: Current Status and Future Prospects. Evolutionary Anthropology 11, 26–36 (2002)

[19] Benedetto, D., Caglioti, E., Loreto, V.: Language Trees and Zipping. Physical Review Letters 88(4), 048702-1– 048702-1 (2002)

[20] Houkes, W.: Tales of Tools and Trees: Phylogenetic Analysis and Explanation in evolutionary Archeology. In: EPSA 2009 2nd Conference of the European Philosophy of Science Association Proceedings (2010), http://philsci-archive.pitt.edu/archive/00005238/

Bootstrap Feature Selection for Ensemble Classifiers

Rakkrit Duangsoithong and Terry Windeatt

Center for Vision, Speech and Signal Processing
University of Surrey
Guildford, United Kingdom GU2 7XH
{r.duangsoithong,t.windeatt}@surrey.ac.uk

Abstract. Small number of samples with high dimensional feature space leads to degradation of classifier performance for machine learning, statistics and data mining systems. This paper presents a bootstrap feature selection for ensemble classifiers to deal with this problem and compares with traditional feature selection for ensemble (select optimal features from whole dataset before bootstrap selected data). Four base classifiers: Multilayer Perceptron, Support Vector Machines, Naive Bayes and Decision Tree are used to evaluate the performance of UCI machine learning repository and causal discovery datasets. Bootstrap feature selection algorithm provides slightly better accuracy than traditional feature selection for ensemble classifiers.

Keywords: Bootstrap, feature selection, ensemble classifiers.

1 Introduction

Although development of computer and information technologies can improve many real-world applications, a consequence of these improvements is that a large number of databases are created especially in medical area. Clinical data usually contains hundreds or thousands of features with small sample size and leads to degradation in accuracy and efficiency of system by curse of dimensionality and over-fitting. Curse of dimensionality [1] , leads to the degradation of classifier system performance in high dimensional datasets because the more features, the more complexity, harder to train classifier and longer computational time. Over-fitting usually occurs when the number of features is high compared to the number of instances. The resulting classifier works very well with training data but very poorly on testing data.

To overcome this high dimensional feature spaces degradation problem, number of features should be reduced. There are two methods to reduce the dimension: feature extraction and feature selection. Feature extraction transforms or projects original features to fewer dimensions without using prior knowledge. Nevertheless, it lacks comprehensibility and uses all original features which may be impractical in large feature spaces. On the other hand, feature selection selects optimal feature subsets from original features by removing irrelevant and

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 28–41, 2010.

redundant features. It has the ability to reduce over-fitting, increase classification accuracy, reduce complexity, speed of computation and improve comprehensibility by preserving original semantic of datasets. Normally, clinicians prefer feature selection because of its understandbility and user acceptance.

There are many applications that applied feature selection as an important pre-processing step to improve systems efficiency, such as web text mining and e-mail classification, intrusion detection, biomedical informatics, gene selection in micro array data, medical data mining, and clinical decision support systems.

Feature selection is important whether the classifier is Multilayer Perceptron (MLP), Support Vector Machines (SVM) or any other classifier. Generally, feature selection can be divided into four categories: Filter, Wrapper, Hybrid and Embedded methods [2], [3], [4]. Filter method is independent from learning method used in the classification process and uses measurement techniques such as correlation, distance and consistency to find a good subset from entire set of features. Nevertheless, the selected subset may or may not be appropriate with the learning method. Wrapper method uses pre-determined learning algorithm to evaluate selected feature subsets that are optimum for the learning process. This method has high accuracy but is computationally expensive. Hybrid method combines advantage of both Filter and Wrapper. It evaluates features by using an independent measure to find the best subset and then uses a learning algorithm to find the final best subset. Finally, Embedded method interacts with learning algorithm but it is more efficient than Wrapper method because the filter algorithm has been built with the classifier. Example of Embedded method is Recursive Feature Elimination (RFE) that is embedded with Support Vectors Machines.

Feature selection has four basic processes [2]: Subset generation, subset evaluation, stopping criterion and subset validation. Subset generation produces

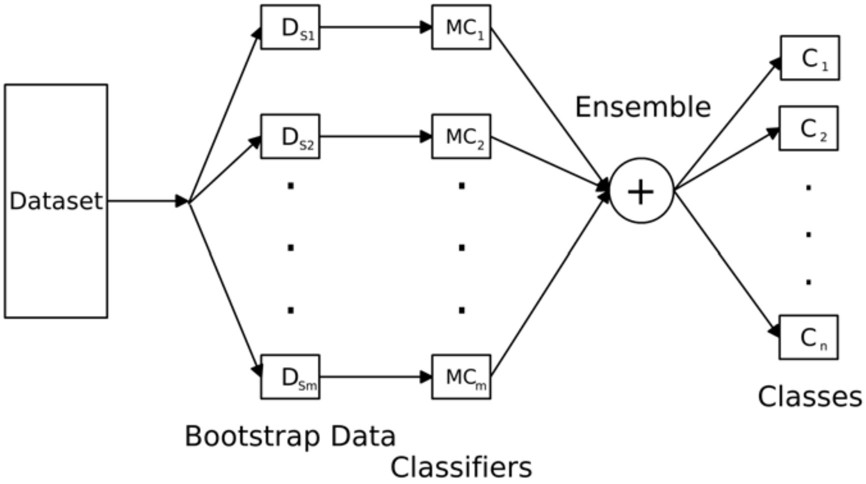


Fig. 1. Ensemble classifiers without feature selection

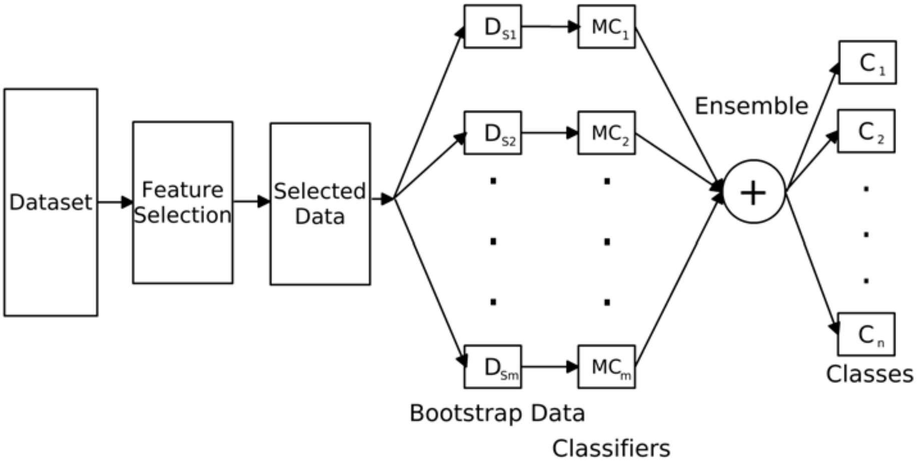


Fig. 2. Ensemble classifiers with feature selection

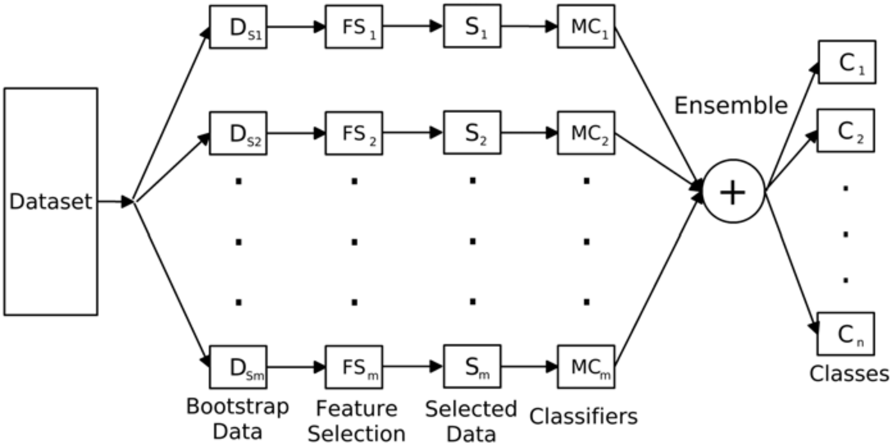


Fig. 3. Bootstrap feature selection for ensemble classifiers

candidate subset by complete, sequential or random search with three directions: forward, backward and bidirectional. After that, the candidate subset is evaluated based on criteria such as distance, dependency, information gain or consistency measurement. The process will stop when it reaches the stopping criterion. Finally, the selected subset is validated with validation data.

An ensemble classifier or multiple classifier system (MCS) is another well-known technique to improve system accuracy [5]. Ensemble combines multiple base classifiers to learn a target function and gathers their prediction together. It has ability to increase accuracy of system by combining output of multiple experts to reduce bias and variance, improve efficiency by decomposing complex problem into multiple sub problems and improve reliability by reducing uncertainty. To increase accuracy, each classifier in the ensemble should be diverse or unique in order to reduce total error such as starting with different input, initial weight, random features or random classes [6].

A typical ensemble classifier system without using feature selection is shown in Figure 1. Normally, feature selection is an essential pre-processing step to improve system performance by selecting optimal features from entire datasets as shown in Figure 2. In this paper, we present a bootstrap feature selection for ensemble classifiers as shown in Figure 3. The original dataset is divided into m bootstrap replicates and uses feature selection in order to remove redundant or irrelevant features of each bootstrap replicate. After that, selected features are passed through ensemble classifier using ensemble algorithm for training and predicting output.

The structure of the paper is the following: Introduction and related research are briefly described in section 1 and Section 2. Section 3 explains theoritical approach of feature selection, bootstrap and ensemble classifiers. The dataset and evaluation procedure are described in Section 4. Experimental results are presented in Section 5 and are discussed in Section 6. Finally, Conclusion is summarized in Section 7.

2 Related Research

Feature selection and ensemble classification have received attention from many researchers in statistics, machine learning, neural networks and data mining areas for many years. At the beginning of feature selection history, most researchers focused only on removing irrelevant features such as ReliefF [7], FOCUS [8] and Correlation-based Feature Selection(CFS) [9]. Recently, in Yu and Liu (2004) [10], Fast Correlation-Based Filter (FCBF) algorithm was proposed to remove both irrelevant and redundant features by using Symmetrical Uncertainty (SU) measurement and was successful for reducing high dimensional features while maintaining high accuracy.

According to Deisy et al. (2007) [11], SU does not have enough accuracy to quantify the dependency among features and does not take into account the effect of pairs of features on the class label during redundancy analysis. Decision Independent Correlation (DIC) and Decision Dependent Correlation (DDC) were proposed instead of using SU to remove irrelevant and redundant features, respectively. DIC and DDC provide better performance than FCBF algorithm in terms of number of selected features, computational time and accuracy.

In Chou et al. (2007) [12], modified FCBF algorithm was used to eliminate both redundant and irrelevant features for intrusion detection. In redundancy analysis, they proposed to calculate SU between features and all original features. They found that FCBF algorithm possibly keeps redundant features in the final optimal subset because it considers only SU between selected features and the rest of features at a time.

In this paper, to overcome this problem of FCBF algorithm, bootstrap feature selection for ensemble classifiers is proposed. Dataset is divided to m bootstrap replicates and selects optimal features from each bootstrap replicate. Finally, ensemble classifiers are considered by using majority vote. This algorithm also solves the small sample size problem by using bootstrap and feature selection techniques.

2.1 Feature Selection with Ensemble Classification

Although feature selection is widely used, there has been little work devoted explicitly to handling feature selection in the context of ensemble classifiers. Many previous researches have focused on determining feature subsets to combine with different ways of choosing subsets. Ho (1998) [13] presented the Random Subspace Method (RSM), one of best known feature selection with ensemble classification. It was shown that a random choice of feature subset, which allows a single feature to be in more than one subset improves performance for high dimensional problems. In Oza and Tumer (2001) [14], feature subsets are selected based on correlation between features and class. Bryll et.al. [15] presented Attribute Bagging that ranks subsets of randomly chosen features before combining. In Skurichina and Duin [16], random selection without replacement and forward features methods are used to find optimal subset. Moreover, most previous approaches have focused on determining selecting optimal features, but rarely to combine with ensemble classification.

2.2 Ensemble Feature Selection and Its Stability

In 1999, Opitz [17] proposed Genetic Ensemble Feature Selection (GEFS) algorithm by using a genetic algorithm (GA) to search and generate multiple good sets of features that are diverse from each other to use for ensemble classifiers. Asymmetric bagging of support vector machines by Li et al. [18] was proposed on predicticting drug activities for unbalanced problem between number of positive and negative samples. Munson and Caruana [19] used Bias-Variance analysis of feature selection for single and bagged model. Hybrid parallel and serial ensemble of tree-based feature selection are proposed by Tuv et al. [20] to find subset of non-redundant features after removing irrelevant features.

Y. Saeys [21] proposed a method to evaluate ensemble feature selection by measuring both stablility (robusness) and classification performance. Gulgezen et al. [22] also proposed stability measurement of MRMR (Minimum Redundancy Maximum Relevance) feature selection by using two feature selection criteria: MID (Mutual Information Difference) and MIQ (Mutual Information Quotient) and proposed new feature selection criterion MIDα.

3 Theoretical Approach

In our research, two correlation-based feature selection: Fast Correlation-Based Filter (FCBF) [10] and Correlation-based Feature Selection with Sequential Forward Floating Search (CFS+SFFS) [9],[23] are investigated for Bagging [24] ensemble classifiers, described in Section 2.2, and experimentally compared with different learning algorithms.

3.1 Feature Selection

Fast Correlation-Based Filter (FCBF). FCBF [10] algorithm is a correlation-based filter that ranks and removes irrelevant and redundant features by measuring Symmetrical Uncertainty (SU) between feature and class and between feature and feature. FCBF has two stages: relevance analysis and redundancy analysis.

Relevance Analysis. Normally, correlation is widely used to analyze relevance. In linear systems, correlation can be measured by linear correlation coefficient.

$$r = \frac{\sum_i (x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_i (x_i - \overline{x_i})^2}\sqrt{\sum_i (y_i - \overline{y_i})^2}} \tag{1}$$

However, most systems in real world applications are non-linear. Correlation in non-linear systems can be measured by using Symmetrical Uncertainty (SU).

$$SU = 2\Big[\frac{IG(X|Y)}{H(X)H(Y)}\Big] \tag{2}$$

$$IG(X,Y) = H(X) - H(X|Y) \tag{3}$$

$$H(X) = -\sum_i P(x_i) log_2 P(x_i) \tag{4}$$

where $IG(X|Y)$ is the Information Gain of X after observing variable Y. $H(X)$ and $H(Y)$ are the entropy of variable X and Y, respectively. $P(x_i)$ is the probability of variable x.

SU is the modified version of Information Gain that has range between 0 and 1 and considers each feature separately (Univariate method). FCBF removes irrelevant features by ranking correlation (SU) between feature and class. If SU between feature and class equal to 1, it means that this feature is completely related to that class. On the other hand, if SU is equal to 0, the feature is irrelevant to this class.

Redundancy analysis. After ranking relevant features, FCBF eliminates redundant features from selected features based on SU between feature and class and between feature and feature. Redundant features can be defined from meaning of predominant feature and approximate Markov Blanket. In Yu and Liu (2004) [10], a feature is predominant (both relevant and non redundant feature) if it does not have any approximate Markov blanket in the current set.

Approximate Markov blanket: For two relevant features F_i and F_j $(i \neq j)$, F_j forms an approximate Markov blanket for F_i if

$$SU_{j,c} \geq SU_{i,c} \; and \; SU_{i,j} \geq SU_{i,c} \tag{5}$$

where $SU_{i,c}$ is a correlation between any feature and the class. $SU_{i,j}$ is a correlation between any pair of feature F_i and F_j $(i \neq j)$.

Correlation-based Feature Selection (CFS). CFS [9] is one of well-known techniques to rank the relevance of features by measuring correlation between features and classes and between features and other features.

Given number of features k and classes C, CFS defined relevance of features subset by using Pearson's correlation equation

$$Merit_s = \frac{kr_{kc}}{\sqrt{k + (k-1)r_{kk}}} \tag{6}$$

where $Merit_s$ is relevance of feature subset, r_{kc} is the average linear correlation coefficient between these features and classes and r_{kk} is the average linear correlation coefficient between different features.

Normally, CFS adds (forward selection) or deletes (backward selection) one feature at a time, however, in this research, we used Sequential Forward Floating Search (SFFS) as the search direction.

Sequential Forward Floating Search (SFFS). SFFS [23] is one of a classic heuristic searching method. It is a variation of bidirectional search and sequential forward search (SFS) that has dominant direction on forward search. SFFS removes features (backward elimination) after adding features (forward selection). The number of forward and backward step is not fixed but dynamically controlled depending on the criterion of the selected subset and therefore, no parameter setting is required.

3.2 Ensemble Classifier

Bagging. Bagging [24] or Bootstrap aggregating is one of the earliest, simplest and most popular for ensemble based classifiers. Bagging uses Bootstrap that randomly samples with replacement and combines with majority vote. Bootstrap is the most well-known strategy for injecting randomness to improve generalization performance in multiple classifier systems and provides out-of-bootstrap estimate for selecting classifier parameters [5]. Randomness is desirable since it increases diversity among the base classifiers, which is known to be a necessary condition for improved performance. However, there is an inevitable trade off between accuracy and diversity known as the accuracy/ diversity dilemma [5].

Bootstrap Feature Selection algorithm. The dataset is divided to m bootstrap replicates. Feature selection will select optimal features from each bootstrap replicate and selected features will be trained by base classifier. m bootstrap replicates are randomly sampled with replacement. Each bootstrap replicate contains, on average, 63.2 % of the original dataset or $(1-1/m)^m \cong 36.8$ % will be removed. Final output will be selected from majority vote from all classifiers of each bootstrap replicate. The architecture is given in Figure 2.

4 Experimental Setup

4.1 Dataset

The medical datasets used in this experiment were taken from UCI machine learning repository [25] : heart disease, hepatitis, diabetes and Parkinson's dataset and from Causality Challenge [26]: lucas and lucap datasets. The details of datasets are shown in Table 1. The missing data are replaced by mean and mode of that dataset. The causal datasets were chosen since they are high-dimension, and furthermore our ultimate goal is to apply ensemble feature selection to causality.

Table 1. Datasets

Dataset	Sample	Features	Classes	Missing Values	Data type
Heart Disease	303	13	5	Yes	Numeric (cont. and discrete)
Diabetes	768	8	2	No	Numeric (continuous)
Hepatitis	155	19	2	Yes	Numeric (cont. and discrete)
Parkinson's	195	22	2	No	Numeric (continuous)
Lucas	2000	11	2	No	Numeric (binary)
Lucap	2000	143	2	No	Numeric (binary)

Heart disease dataset was contributed by Cleveland Clinic foundation has 303 samples, 13 attributes with 138 samples presenting for heart disease class and 165 samples for absent class.

Diabetes dataset. Prima Indians Diabetes dataset was donated by John Hopkins University has 768 samples, 8 numeric features with tested positive and tested negative classes.

Hepatitis dataset was donated by G.Gong from Carnegie-Mellon University contains 155 instances, 19 attributes with live or die classes.

Parkinson's dataset. Parkinson's disease dataset is the speech signals recorded by Max Little from University of Oxford collaborated with the National Centre for Voice and Speech, Denver, Colorado. It has 197 samples, 23 features with two classes (healthy and Parkinson's patient).

Lucas dataset. Lucas (LUng CAncer Simple set) dataset is toy data generated artificially by causal Bayesian networks with binary features. This dataset is modeling a medical application for the diagnosis, prevention and cure of lung cancer. Lucas has 11 features with binary classes and 2000 samples.

Lucap dataset. Lucap (LUng CAncer set with Probes) is Lucas dataset with probes which are generated from some functions plus some noise of subsets of the real variables. Lucap has 143 features, 2000 samples and binary classes.

4.2 Evaluation

To evaluate the feature selection process, we use four widely used classifiers: Naive-Bayes(NB), Multilayer Perceptron (MLP), Support Vector Machines (SVM) and Decision Trees (DT). The parameters of each classifier were chosen based on the base classifier accuracy. MLP has one hidden layer with 16 hidden nodes, learning rate 0.2, momentum 0.3, 500 iterations and uses backpropagation algorithm with sigmoid transfer function. SVMs uses polynomial kernel with exponent 2 and set the regularization value to 0.7 and Decision Trees use pruned C4.5 algorithm. The number of classifiers in Bagging is varied from 1, 5, 10, 25 to 50 classifiers. The threshold value of FCBF algorithm in our research is set at zero for heart disease, diabetes, parkinson's and lucas and 0.13 and 0.15 for hepatitis and lucap dataset, respectively.

The classifier results were validated by 10 fold cross validation with 10 repetitions for each experiment and evaluated by average percent of test set accuracy.

5 Experimental Result

Table 2 shows the average number of selected features for each dataset. Figure 4 and 5 present example of the average accuracy of heart disease and lucap dataset. Y-axis presents the average percent accuracy of the four base classifiers and X-axis shows the number of ensemble from 1 to 50 classifiers. Solid line presents original data set, dashed line is the result of bootstrap feature selection using FCBF and bootstrap feature selection using CFS+SFFS is shown as short-dashed line. FCBF before Bootstrap result is presented in dotted line and CFS+SFFS before bootstrap is shown in dashed with dotted line.

Figure 6 and 7 show the average accuracy of six datasets for each classifier and average of all classifiers for all six datasets, respectively. Finally, Table 3

Table 2. Average number of selected features

Dataset (Original features)	Algorithm	Average number of selected features from bootstrap					
		Whole features	1	5	10	25	50
Heart Disease(13)	FCBF	6.00	4.00	5.60	5.70	5.96	6.08
	CFS+SFFS	9.00	7.00	7.60	7.80	7.88	7.90
Diabetes(8)	FCBF	4.00	4.00	3.80	3.80	3.80	3.84
	CFS+SFFS	4.00	5.00	4.80	4.50	4.32	4.40
Hepatitis(19)	FCBF	3.00	3.00	3.40	3.40	3.20	3.38
	CFS+SFFS	10.00	5.00	6.60	7.10	7.24	7.48
Parkinson's(23)	FCBF	5.00	5.00	4.00	4.30	4.28	4.36
	CFS+SFFS	10.00	10.00	8.20	8.00	8.12	8.18
Lucas(11)	FCBF	3.00	3.00	3.00	3.00	3.08	3.10
	CFS+SFFS	3.00	3.00	3.20	3.40	3.52	3.50
Lucap(143)	FCBF	7.00	7.00	6.60	8.60	7.88	7.94
	CFS+SFFS	36.00	36.00	32.60	33.40	33.32	32.96

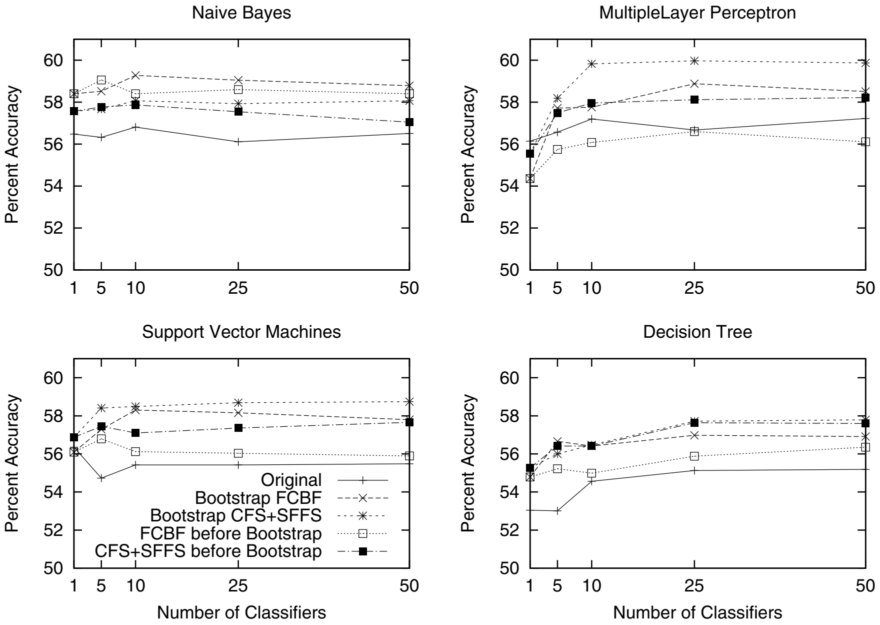


Fig. 4. Average Percent Accuracy of heart disease dataset

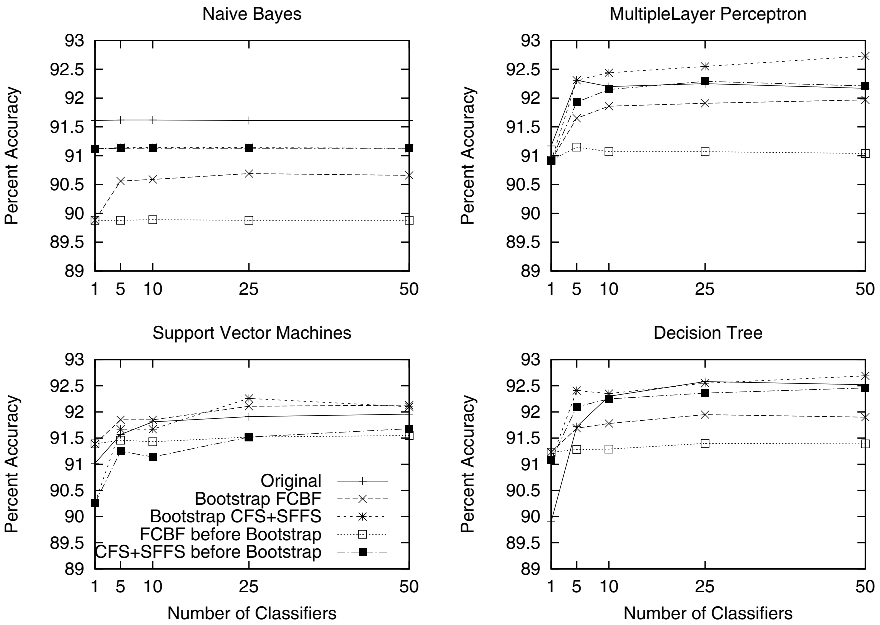


Fig. 5. Average Percent Accuracy of lucap dataset

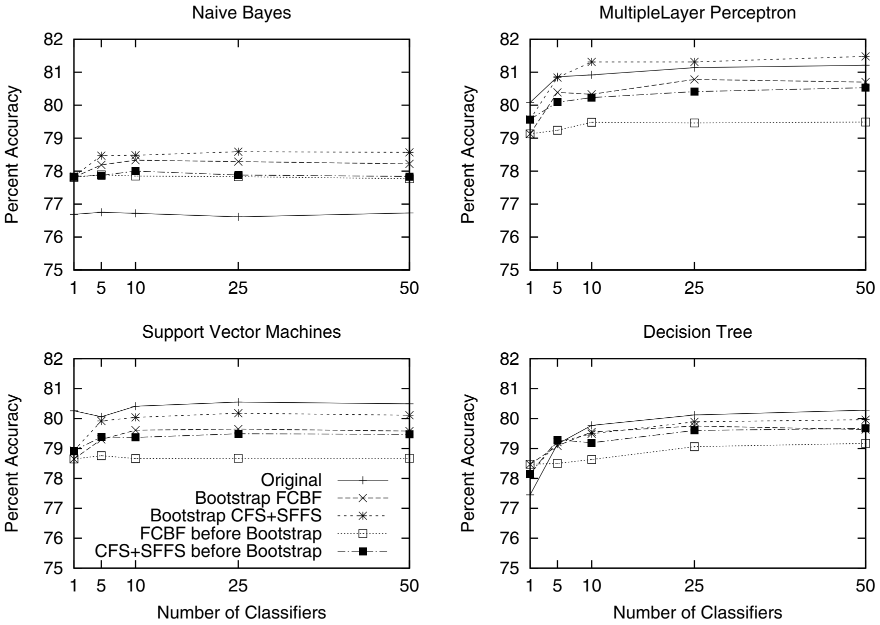


Fig. 6. Average Percent Accuracy of six datasets for each classifier

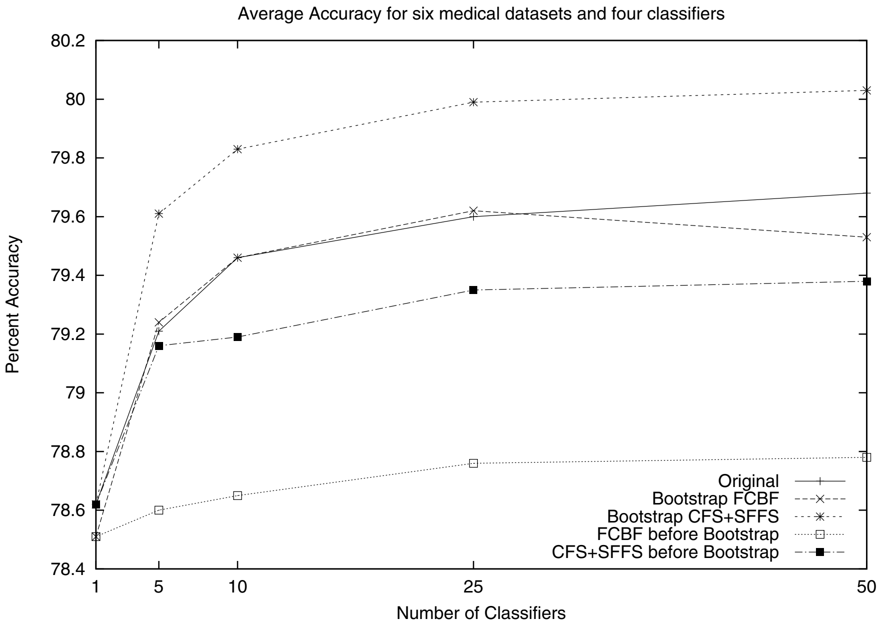


Fig. 7. Average Percent Accuracy of six datasets, four classifiers

Table 3. T-statistic test for 50 MLP classifiers of heart disease dataset

T-test		Original	bootstrap feature selection		feature selection before bootstrap	
			FCBF	CFS+SFFS	FCBF	CFS+SFFS
Original		-	1	1	0	1
Bootstrap feature selection	FCBF	0	-	1	0	0
	CFS+SFFS	0	0	-	0	0
Feature Selection before bootstrap	FCBF	1	1	1	-	1
	CFS+SFFS	0	1	1	0	-
Note: 1 = column did score significant win with regard to row Note: 0 = column did not score significant win with regard to row						

presents the example of T-statistic test (T-Test) for heart disease dataset using 50 MLP classifiers (the number of significant win of column compared to row).

6 Discussion

According to figures 4-7, bootstrap feature selection (figure 3) provides slightly better average accuracy than traditional feature selection for ensemble (figure 2 - feature selection from whole dataset before bootstrap selected data) in both FCBF and CFS+SFFS algorithms. On average over four classfiers and six datasets, figure 7 shows that bootstrap feature selection using CFS+SFFS provides better average accuracy than original features, bootstrap feature selection using FCBF, traditional feature selection for ensemble using CFS+SFFS and traditional feature selection for ensemble using FCBF algorithm, repectively. From table 2, it can be seen that FCBF algorithm can eliminate more redundant and irrelvant features than CFS+SFFS algorithm. Note that the average number of selected features for each number of bootstrap from 1-50 bootstrap replicates are dissimilar. This means that when we random sample with replacement, the selected feature can be different for each bootstrap replicate.

From the example of T statistic test (T-Tset) in Table 3 for heart disease dataset with 50 MLP classifiers, bootstrap feature selection using CFS+SFFS significantly improves average accuracy compared to other feature selection algorithms for ensemble. Bootstrap feature selection using FCBF algorithm also significantly outperforms other feature selection algorithms except bootstrap feature selection using CFS+SFFS. Feature selection before bootstrap using FCBF algorithm does not have significant accuracy improvement compared to other algorithms.

Furthermore, bootstrap feature selection has higher complexity than traditional feature selection for ensemble because it has to select optimal features for each bootstrap replicate.

7 Conclusions

In this paper, bootstrap feature selection for ensemble classifiers is presented and compared with conventional feature selection for ensemble classifiers. According to the average results, bootstrap feature selection for ensemble classifiers provides accuracy slightly higher than the traditional feature selection for ensemble classifiers. The only drawback of this algorithm is the complexity which is increased due to selection of optimal features of each bootstrap replicate. Future work will investigate the result of bootstrap causal feature selection for ensemble classifiers.

References

1. Bellman, R.E.: Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton (1961)
2. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17(4), 491–502 (2005)
3. Saeys, Y., Inza, I., Larranaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics 23(19), 2507–2517 (2007)
4. Duangsoithong, R., Windeatt, T.: Relevance and Redundancy Analysis for Ensemble Classifiers. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition, vol. 5632, pp. 206–220. Springer, Heidelberg (2009)
5. Windeatt, T.: Ensemble MLP Classifier Design, vol. 137, pp. 133–147. Springer, Heidelberg (2008)
6. Windeatt, T.: Accuracy/diversity and ensemble MLP classifier design. IEEE Transactions on Neural Networks 17(5), 1194–1211 (2006)
7. Witten, I.H., Frank, E.: Data Mining Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
8. Almuallim, H., Dietterich, T.G.: Learning with many irrelevant features. In: Proceedings of the Ninth National Conference on Artificial Intelligence, pp. 547–552. AAAI Press, Menlo Park (1991)
9. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: Proceeding of the 17th International Conference on Machine Learning, pp. 359–366. Morgan Kaufmann, San Francisco (2000)
10. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. 5, 1205–1224 (2004)
11. Deisy, C., Subbulakshmi, B., Baskar, S., Ramaraj, N.: Efficient dimensionality reduction approaches for feature selection. In: International Conference on Computational Intelligence and Multimedia Applications, vol. 2, pp. 121–127 (2007)
12. Chou, T., Yen, K., Luo, J., Pissinou, N., Makki, K.: Correlation-based feature selection for intrusion detection design. In: IEEE on Military Communications Conference, MILCOM 2007, pp. 1–7 (2007)
13. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(8), 832–844 (1998)
14. Oza, N.C., Tumer, K.: Input decimation ensembles: Decorrelation through dimensionality reduction. In: Proceeding of the 2nd International Workshop on Multiple Classier Systems, pp. 238–247. Springer, Heidelberg (2001)
15. Bryll, R.K., Osuna, R.G., Quek, F.K.H.: Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. Pattern Recognition 36(6), 1291–1302 (2003)
16. Skurichina, M., Duin, R.P.W.: Combining feature subsets in feature selection. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 165–175. Springer, Heidelberg (2005)

17. Opitz, D.W.: Feature Selection for Ensembles. In: AAAI 1999: Proceedings of the 16th National Conference on Artificial Intelligence, pp. 379–384. American Association for Artificial Intelligence, Menlo Park (1999)
18. Li, G.Z., Meng, H.H., Lu, W.C., Yang, J., Yang, M.: Asymmetric bagging and feature selection for activities prediction of drug molecules. Journal of BMC Bioinformatics 9, 1471–2105 (2008)
19. Munson, M.A., Caruana, R.: On Feature Selection, Bias-Variance, and Bagging. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009, Part II. LNCS (LNAI), vol. 5782, pp. 144–159. Springer, Heidelberg (2009)
20. Tuv, E., Borisov, A., Runger, G., Torkkila, K.: Feature Selection with Ensembles, Artificial Variables, and Redundancy Elimination. Journal of Machine Learning Research 10, 1341–1366 (2009)
21. Saeys, Y., Abeel, T., Van de Peer, Y.: Robust Feature Selection Using Ensemble Feature Selection Techniques. In: Daelemans, W., Goethals, B., Morik, K. (eds.) ECML PKDD 2008, Part II. LNCS (LNAI), vol. 5212, pp. 313–325. Springer, Heidelberg (2008)
22. Gulgezen, G., Cataltepe, Z., Yu, L.: Stable and Accurate Feature Selection. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) ECML PKDD 2009. LNCS (LNAI), vol. 5781, pp. 455–468. Springer, Heidelberg (2009)
23. Pudil, P., Novovicova, J., Kitler, J.: Floating Search Methods in Feature Selection. Pattern Recognition Letters 15, 1119–1125 (1994)
24. Breiman, L.: Bagging predictors. Machine Learning 24(2), 123–140 (1996)
25. Asuncion, A., Newman, D.: UCI machine learning repository (2007), http://www.ics.uci.edu/mlearn/MLRepository.html
26. Guyon, I.: Causality Workbench (2008), http://www.causality.inf.ethz.ch/home.php

Evaluating the Quality of Clustering Algorithms Using Cluster Path Lengths

Faraz Zaidi, Daniel Archambault, and Guy Melançon

CNRS UMR 5800 LaBRI & INRIA Bordeaux - Sud Ouest
351, cours de la Libération
33405 Talence cedex, France
{faraz.zaidi,guy.melancon}@labri.fr,
daniel.archambault@inria.fr

Abstract. Many real world systems can be modeled as networks or graphs. Clustering algorithms that help us to organize and understand these networks are usually referred to as, graph based clustering algorithms. Many algorithms exist in the literature for clustering network data. Evaluating the quality of these clustering algorithms is an important task addressed by different researchers. An important ingredient of evaluating these clustering techniques is the node-edge density of a cluster. In this paper, we argue that evaluation methods based on density are heavily biased to networks having dense components, such as social networks, but are not well suited for data sets with other network topologies where the nodes are not densely connected. Example of such data sets are the transportation and Internet networks. We justify our hypothesis by presenting examples from real world data sets.

We present a new metric to evaluate the quality of a clustering algorithm to overcome the limitations of existing cluster evaluation techniques. This new metric is based on the path length of the elements of a cluster and avoids judging the quality based on cluster density. We show the effectiveness of the proposed metric by comparing its results with other existing evaluation methods on artificially generated and real world data sets.

Keywords: Evaluating Cluster Quality, Cluster Path Length.

1 Introduction

Many real world systems can be modeled as networks or graphs where a set of nodes and edges are used to represent these networks. Examples include social networks, metabolic networks, world wide web, food web, transport and Internet networks. *Community detection* or *Clustering* remains an important technique to organize and understand these networks [6] where [22] provides a good survey of graph based clustering algorithms. A cluster can be defined as a group of elements having the following properties as described by [24]:

- Density: Group members have many contacts to each other. In terms of graph theory, it is considered to be the ratio of the number of edges present in a group of nodes to the total number of edges possible in that group.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 42–56, 2010.

- Separation: Group members have more contacts inside the group than outside.
- Mutuality: Group members choose neighbors to be included in the group. In a graph-theoretical sense, this means that they are adjacent.
- Compactness: Group members are 'well reachable' from each other, though not necessarily adjacent. Graph-theoretically, elements of the same cluster have short distances.

The Density of a cluster can be measured by the equation $d = e_{actual}/e_{total}$ where e_{actual} represents the actual number of edges present in the cluster and e_{total} represents the total number of possible edges in the cluster. Density values lie between [0,1] where a value of 1 suggests that every node is connected to every other node forming a clique.

The Separation can be calculated by the number of edges incident to a cluster, i.e the number of edges external to the clusters. This is often referred to as the cut size and can be normalized by the total number of edges incident to the cluster. Low values represent that the cluster is well separated from other clusters where high values suggest that the cluster is well connected to other clusters.

Mutuality and Compactness of a cluster can easily be evaluated using a single quantitative measure: the average path length between all the nodes of a cluster. The path length refers to the minimum number of edges connecting node A to node B. The average path length represents how far apart any two nodes lie to each other and is calculated by taking the average for all pairs of nodes. This value can be calculated for a cluster giving us the average path length of a particular cluster. Low values indicate that the nodes of a cluster lie in close proximity and high values indicate that the cluster is sparse and its nodes lie distant to each other.

Cluster Detection has a wide range of applications in various fields. For example, in social networks, community detection could lead us towards a better understanding of how people collaborate with each other. In a transport network, a community might represent cities or countries well connected through transportation means. There are many algorithms addressing the issue of clustering and readers are referred to various surveys on the topic [22,9,3] for further information. Evaluating different clustering algorithms remains essential to measure the quality of a given set of clusters. These evaluation metrics can be used for the identification of clusters, choose between alternative clusterings and compare the performance of different clustering algorithms [22].

Most of the evaluation metrics consider *density* as a fundamental ingredient to calculate the quality of a cluster. From the definition of clusters given above, density is an important factor but not the only factor to be considered while evaluating the quality of clustering. Having a densely connected set of nodes might be a good reflection of nodes being adjacent to each other or lying at short distances but the inverse conjecture might not necessarily be true as illustrated in Fig. 1. Consider the set of five nodes in Fig. 1(a,b,c) being identified as clusters by some clustering algorithm. The density of graph in Fig. 1(a) is 1 and that

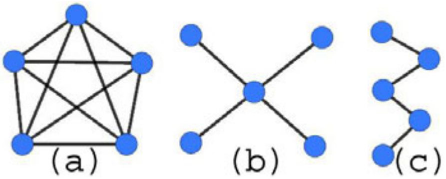


Fig. 1. (a) Represents a *clique* (b) presents a *star-like* structure and (c) is a set of nodes connected to each other in a *chain-like* structure

of (b) and (c) is 0.4. Intuitively (b) is more cohesive than (c). Moreover the average path length of (b) is lower than that of (c) suggesting that the elements of cluster (b) are closer to each other. From this example, we can deduce that, if we consider density as the only criteria, then for such an evaluation metric, (b) and (c) will be assigned a similar value which is not consistent with Mutuality and Compactness.

Another important class of evaluation metric uses connectivity of clusters to capture the notion of Separation. The simplest way to measure this is the *cut size* which is defined as the minimum number of edges required to be removed so as to isolate a cluster. Consider the graphs in Fig. 2(a,b,c) with enclosed nodes representing clusters. Calculating the cut size for all these clusters will give the same cut size, which is 1 in these examples, as each cluster is connected to the rest of the graph through exactly one edge. The example suggests that cut-size alone is not a good representation of the quality of clustering as all the clusters in Fig. 2 have the same cut-size.

More sophisticated measures combining *density* and *cut size* have been investigated with the most important example being *relative density* [13]. Even combining these two metrics, the clusters in Fig. 2(b) and (c) will be assigned an equal score, failing to incorporate Mutuality and Compactness of a cluster. Calculating the density and the cut size of these two clusters will result in the exact same value. We present other cluster evaluation techniques in Sect. 2.

If we consider Density, Mutuality and Compactness together to evaluate the quality of clusters present in Fig. 2, the highest measure should be associated

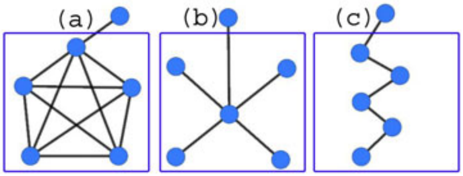


Fig. 2. Represents three graphs with enclosed nodes being the clusters. All the clusters have the same *cut size* which is equal to 1. Based on the *cut size* alone the quality of the clustering cannot be judged.

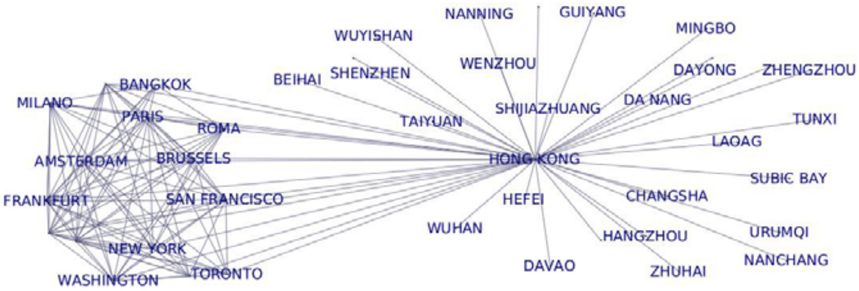


Fig. 3. Air Traffic network drawn using Hong Kong at the center and some airports directly connected to Hong Kong. We can see the worlds most important cities having a direct flight to Hong Kong whereas there are lots of regional airports connected to Hong Kong representing a *star-like* structure as discussed previously in Fig. 1(b) and 2(b).

to cluster (a) as it is the cluster with the highest Density, Mutuality and Compactness. Then cluster (b) where it has high Mutuality and Compactness but low density and finally cluster (c) which is the least Dense, Mutual and Compact cluster of the three clusters present in Fig. 2. We show that the existing cluster evaluation metrics do not evaluate the quality of clusters in this order. We discuss the details in Sect. 4.

Until now, we have argued that ignoring Mutuality and Compactness of a cluster to evaluate its quality can give inconsistent results. A simple question can be raised about the importance of these two criterion especially for real world data sets. To answer this question, we turn our focus towards some real world data sets. Consider the example of an **Air Traffic Network** which represents an airport-airport graph where two airports are connected through an edge if a direct flight exists between them [21]. In this particular case, we took Hong Kong as an example by taking some airports directly connected to it as shown in Fig. 3[1]. On one side, we can see some of the world's biggest cities having direct flights to Hong Kong where on the other hand, we have lots of regional airports also directly connected to Hong Kong. If we consider a cluster by putting Hong Kong with the regional airports, the resulting cluster will have very low density and high cut size which are undesirable features for a cluster. In the other case, where we consider Hong Kong as part of the cluster with the biggest cities in the world, the cluster with Hong Kong will have a high cut size. Moreover, the regional airports could not be clustered together as they will no longer remain connected to each other. We will end up with lots of singleton clusters which again will reduce the overall quality of any clustering algorithm.

Another example of these star-like structures comes from **Internet Tomography Networks** which is a collection of routing paths from a test host to other

[1] All the images in this paper are generated using TULIP software which is an open source software for the analysis and visualization of large size networks and graphs available at: http://www.tulip.labri.fr/

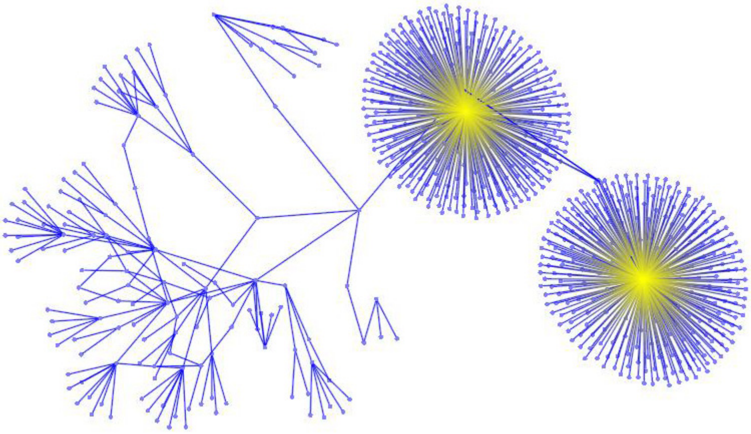


Fig. 4. Internet Tomography Network representing routing paths from a test host to other networks. Two nodes clearly dominate the number of connections as they play the role of hubs to connect several clients. Another example of *star-like* structures in the real world.

networks on the Internet. The database contains routing and reachability information, and is available to the public from the Opte Project website (`http://opte.org/`). Considering two hubs from this data set and taking all the nodes lying at distance five from these hubs, we obtain a structure as shown in Fig. 4. The two hubs dominate the number of connections in these networks presenting the *star-like* behavior in real world data sets.

As opposed to these *star-like* structures, the other most common structure present in most real world data sets is the presence of *cliques*. Social networks are good examples of networks having cliques. As an example data set, consider the collaboration network of researchers usually called the **Co-Authorship Network** [18]. Two authors are connected by an edge if they appear as authors in an article. Scientists co-authoring an article will end up having edges with every other co-author thus forming a clique. Another example of such a network is the Movie network where two actors are connected to each other if they have acted in a movie together [1]. Just as in the case of co-authorship network, actors appearing in a movie together will form a clique and thus represent dense communities.

Metrics based on density and cut size prove to be adequate for networks having densely connected nodes or cliques. Results have shown that different clustering algorithms perform well for these networks [6,16,1]. On the other hand, in case where lots of star-like structures exist (see Fig. 3 and 4), an evaluation based on density and cut size fails to perform well as shown in the examples discussed previously. To resolve this problem, we propose a new cluster evaluation metric which takes into account the underlying network structure by considering the average path lengths to evaluate the cluster quality.

Apart from these cliques and star-like structures, other interesting topologies exist in different data sets but are highly dependent on the application domain. Examples include motifs in Chemical Compounds [4] or Metabolic Networks [11] where the goal is to search motifs in graphs and not to cluster them based on some similarity. We focus our attention to generic data sets and evaluating clustering algorithms for specific data sets remains out of the scope of this paper.

The design principle for the proposed metric is very simple and intuitive. Instead of considering *density* as the fundamental component to evaluate the quality of a clustering algorithm, we use the average path length to determine the closeness of the elements of a cluster. It is obvious that in case of a clique, the path length between the nodes is 1 which is the minimum possible value for two connected nodes. But the important aspect here is that a star-like structure will have a higher average path length as compared to a chain like structure thus providing a way to evaluate how close the nodes are of a cluster, irrespective of the density of edges. We discuss the details of the proposed metric further in Sect. 3.

The paper is organized as follows. In the following section, we provide a brief overview of some widely used metrics to evaluate cluster quality. In Sect. 3 we present the proposed metric and we discuss our findings by performing a comparative study of the different evaluation metrics in Sect. 4. Finally in Sect. 5 we present our conclusions and future research directions in light of the newly proposed metric.

2 Related Work

The different approaches to evaluate cluster quality can be classified as *external*, *relative* or *internal*. The term *external* validity criteria is used when the results of the clustering algorithm can be compared with some pre-specified clustering structures [7] or in the presence of ground truth [20]. *Relative* validity criteria measure the quality of clustering results by comparing them with the results of other clustering algorithms [12]. *Internal* validity criteria involve the development of functions that compute the cohesiveness of a clustering by using density, cut size, distances of entities within each cluster, or the distance between the clusters themselves etc [14,19,8].

For most real world data sets, an external validity criteria is simply not available. In the case of relative validity criteria, as Jain[9] argues, there is no clustering technique that is universally applicable in uncovering the variety of structures present in multidimensional data sets. Thus we do not have an algorithm that can generate a bench mark clustering for data sets with varying properties. For these reasons we focus our attention on internal quality metrics only. Further more, we deal with quality metrics for partitional or flat clustering algorithms that are non-overlapping.

Modularity(Q) [16] (Q metric) is a metric that measures the fraction of the edges in the network that connect within-community edges minus the expected value of the same quantity in a network with the same community divisions but random connections between the vertices. If the number of within-community edges is no better than random, we will get Q = 0. Values approaching Q = 1, which is the maximum, indicate strong community structure.

Another metric used by Auber *et al.* [1] to effectively evaluate the quality of clustering for small world graphs is the MQ metric initially proposed by Mitchell *et al.* [15] as a partition cost function in the field of software reverse engineering. It comprises of two factors where the first term contributes to the positive weight represented by the mean value of edge density inside each cluster. The second term contributes as a negative weight and represents the mean value of edge density between the clusters.

The Relative Density [13] of a cluster calculates the ratio of the edge density inside a cluster to the sum of the edge densities inside and outside that cluster. The final Relative Density is the averaged sum of the these individual relative densities for all clusters.

For our experimentation and comparison, we use the three metrics presented above. Other notable metrics used to evaluated the quality of clustering include coverage [2], conductance [10], performance [2] but since they are based on more or less the same principles to evaluate the quality of clusterings, we do not include them in this study.

3 Proposed Metric for Cluster Evaluation: Cluster Path Lengths

As we discussed earlier, the design principle which makes our metric novel, is the fact that we consider the path length of elements of a cluster. The metric is composed of two components, the positive component($M^+(G)$) which assigns a positive score to a cluster and a negative component($M^-(G)$) which attributes a negative score to edges between clusters. The positive score is assigned on the basis of the density, compactness and mutuality of the cluster whereas the negative score is assigned on the basis of the separation of the cluster from other clusters. The final quality of a cluster is simply the sum of the two components given by the equation:

$$M(G) = M^+(G) - M^-(G) \tag{1}$$

In the above equation, the two components are weighted equally. An option can be to assign different weights to the two components, for example a higher weight to the positive component, for the sake of simplicity, we have not experimented with different weights. We discuss the details of how the positive and the negative components are calculated below.

3.1 Positive Component

The goal is to assign a quantitative value to a cluster based on its density, compactness and mutuality. Looking at the different clusters in Fig. 2, if we calculate the average path length of the nodes within the cluster, the least value would be assigned to cluster (a), then cluster (b) and finally (c). This is quite intuitive as we reduce the average distance between nodes of a cluster, the density tends to increase. Lets call the average path length of each cluster Cluster Path Length. The best possible average path length for any cluster can be 1 in the case when every node is connected to every other node forming a clique. The normalized cluster path length can be given by the following equation:

$$CPL_i = \frac{1}{AvgPathLen_i} \tag{2}$$

Where CPL_i represents the normalized cluster path length of cluster i and $AvgPathLen_i$ represents the average path length of the nodes in cluster i. Higher this value is for a cluster, better is the quality of the cluster where the values lie in the range of [0,1]. The overall cluster path length is then averaged for all clusters where k is the total number of clusters, giving us the value for the positive component to evaluate the quality of the clustering:

$$M^+(G) = CPL_{1 \cdots k} = \frac{1}{k}\sum_{i=1}^{k} CPL_i \tag{3}$$

3.2 Negative Component

The next step is to assign a negative score to penalize the inter-cluster edges. The value of M^- evaluates the separation of the two clusters. This score is calculated for each pair of clusters and is based on the number of edges that link two clusters i and j compared to the total number of edges possible between these two clusters. Let n_i and n_j be the number of nodes contained in clusters i and j respectively. Therefore, the edge penalty for the edges present between these two cluster would be given by the equation:

$$EdgePenalty_{(i,j)} = \frac{e_{ij}}{n_i * n_j} \tag{4}$$

Where e_{ij} is the number of edges present between clusters i and j. The overall Edge Penalty ($M^-(G)$) is the average calculated for all pair of clusters given by the equation:

$$M^-(G) = \frac{2}{k*(k-1)} \sum_{i=1,j=1}^{k} EdgePenalty_{(i,j)} \quad where(i \neq j) \tag{5}$$

The negative score sums all edge penalties over all pairs of clusters and then normalizes the value by $k(k-1)/2$ to produce an overall penalty in the range [0,1]. This value is linearly proportional to the number of edges present between clusters where low values correspond to few broken edges and a better clustering quality.

To summarize the proposed metric, we use the cluster path lengths to assign a positive score to evaluate the quality of clustering subtracted by a negative score which is based on the inter-cluster density. The values lie in the range of [0,1] where low values indicate poor clustering and high values indicate better clustering. We refer to the metric as *CPL* for Cluster Path Lengths (although we subtract the Edge penalties from the CPLs calculated).

4 Experimentation

For evaluating different cluster quality metrics, we use two different experiments. The first, where we generate artificial data sets and the second where we use real world data sets.

4.1 Artificial and Clustered Data Set

For the artificial data set, we directly generate clusters to avoid biasing the experiment using any particular clustering algorithm. We generate three clustered graphs of size n. We generate a random number k between 1 and Max to determine the size of a cluster. For the first graph, we add k nodes such that each node is connected to the other forming a *clique*. For the second graph, k nodes are added such that a *star-like* is formed and finally k nodes are added to the third graph forming a chain like structure. The process is repeated until the maximum number of nodes in the graphs reach n. The clusters in each of these graphs are connected by randomly adding *RandE* edges. This number decides the number of inter-cluster edges that will be produced for each graph. The choice of selecting the variables n, Max and *RandE* are independent of the experiment and do not change the final evaluation. For our experiment, we used $n = 200$, $Max = 20$ and $RandE = 40$.

Two important inferences can be drawn from the experiment described above. The first, where we compare how the different evaluation metrics perform for evaluating the quality of clusters where each cluster is a clique with some inter-cluster edges. Looking at the high values for the all the evaluation metrics, we can justify that all the metrics are consistent in evaluating the quality of clusters including the newly proposed metric. As discussed previously, density based metrics perform well when the clusters are densely connected, and so does the proposed metric.

The other important result can be derived by comparing the values assigned to the *star-like* clusters and *chain-like* clusters by different evaluation metrics.

Table 1. Evaluating the quality of clustering using three topologically different and artificially generated clustered data sets

Cluster Quality Metric	**Cliques**	**Star-like**	**Chain-like**
Cluster Path Length	0.998	0.611	0.374
Q metric	0.975	0.281	0.281
MQ metric	0.998	0.844	0.844
Relative Density	0.862	0.711	0.711

Clearly the other metrics fail to differentiate between how the edges are distributed among the clusters ignoring the Mutuality and Compactness of a cluster whereas CPL does well by assigning higher values to *star-like* clusters as compared to *chain-like* clusters. This justifies the use of cluster path length as a metric to evaluate the quality of clusters specially where dense clusters are not expected.

4.2 Real World Data Sets and Clustering Algorithms

The second experiment uses real world data sets. We use four different data sets, two of them were briefly introduced earlier in Sect. 1. We give the source and description of each data set below.

The Co-authorship network is network of scientists working on network theory and experiments, as compiled by M. Newman in May, 2006 [18]. The network was compiled from the bibliographies of two review articles on networks, M. E. J. Newman, SIAM Review and S. Boccaletti et al., Physics Reports, with a few additional references added by hand. The biggest connected component is considered for experimentation which contains 379 nodes and 914 edges.

The Air Transport Network is an undirected graph where nodes represent airports and edges represent a direct flight from one airport to the other. The network contains 1540 nodes and 16523 edges. The node-edge density of the graph indicates that the average degree of node is around 10, but actually the graph follows a scale free degree distribution where some nodes have very high degree and many nodes have low degree (see [21] for more details). This is quite understandable because the worlds busiest airports like Paris, New York, Hong Kong, London etc have flights to many other destinations and small cities or regional airports have very restricted traffic as shown in Fig. 3.

The Internet network is a network mapping data which consists of paths from a test host towards other networks on the Internet containing routing and reachability information. The complete data set is available from the Opte Project website (www.opte.org). The entire data set contains 35836 nodes and 42387 edges. Since the Divisive Clustering algorithm has a high time complexity, we only consider a subset of the actual data set constructed by considering a hub and the nodes connected at distance 5 from it. The subset consists of 1049 nodes and 1319 edges.

The fourth data set is a Protein-Protein interactions network. The data represents a set of S. cerevisiae interactions identified by TAP purification of protein complexes followed by mass-spectrometric identification of individual components used by [5]. The data is available from http://dip.doe-mbi.ucla.edu/dip and contains 1246 nodes and 3142 edges. Around 80 nodes were disconnected from the biggest connected component and were removed for this experimentation.

The choice of Air Traffic, the Internet Tomography and the Protein network is purely based on the fact that these networks do not have densely connected components. Rather there are components that have chain-like structures and star-like structures. On the other hand we use the co-authorship network to show the efficiency of the clustering algorithms used as they perform well in detecting communities present in the network.

To cluster these data sets, we use two known clustering algorithms, the Bisecting K-Means algorithm [23] and the Divisive Clustering algorithm based on Edge Centrality [6]. The choice of these algorithms is based on the criteria that these algorithms do not try to optimize or influence the clustering algorithm based on the density or some other cluster quality metric as compared to other algorithms present in the literature such as [17]. We also use the Strength Clustering algorithm proposed by [1] which was initially introduced to cluster social networks. The algorithm has been shown to perform well for the identification of densely connected components as clusters.

The Bisecting K-Means algorithm and the Divisive Clustering algorithm based on Edge Centrality are both divisive algorithms, i.e. they start by considering the entire graph as a single cluster and repeatedly divide the cluster into two clusters. Both these algorithms can be used to create a hierarchy where the divisive process stops when each cluster has exactly one node left. Instead of generating the entire hierarchy, we stop the process as soon as the minimum number of nodes in the cluster reaches around 20 nodes. Moreover since we do not propose a method to evaluate the quality of a hierarchical clustering algorithm, we consider the leaves as a single partitional clustering. Note that the clustering algorithm might create singletons but while evaluating the quality of clusters we do not consider clusters having a single element. The results for evaluating the clusters obtained for the two data sets are given in Table 2.

The Strength clustering algorithm uses the strength metric for clustering. This metric quantifies the neighborhoods cohesion of a given edge and thus identifies if an edge is an intra-community or an inter-community edge. Based on these strength values, nodes are judged to be part of the same cluster (see [1] for more details). The reason for using this clustering algorithm is to demonstrate that irrespective of the clustering algorithm, the CPL metric evaluates the quality of a clustering. Since the other two algorithms do not force the detection of strongly connected components, we use Strength clustering as a representative of clustering algorithms that try to detect densely connected nodes.

Analyzing the results presented in Table 2, first we look at the Co-authorship network. The high values of the Divisive algorithm for all the evaluation metric suggest that the algorithm does well to find the good clusters. Bisecting K-Means seem to perform quite well also for this data set although values for the CPL and MQ metric are comparatively lower than the divisive algorithm. Looking at the results of Strength Clustering using CPL and MQ, the values are quite high indicating that the algorithm found high quality clusters but the low Q metric and Relative Density values create some doubt about the performance of the algorithm. This variation is due to the large number of clusters generated by Strength clustering (122) as compared to Divisive (23) and Bisecting K-Means (38) algorithm. While evaluating the quality using Q metric and Relative Density, this high number of clusters reduces its quality as it results in high number of inter-cluster edges.

In case of the Air Traffic network, the clusterings generated by the Bisecting K-Means and Divisive algorithms are relatively poorly judged as compared to the CPL and MQ metric. This is a clear indication that when considering the star-like structures as clusters which are present in abundance in the Air-Traffic network, the evaluation metrics judge the performance of the clustering algorithms to be poor. This is because there are not many densely connected airports in the network. High values of CPL indicate that even though, the clusters are not densely connected, they lie in close proximity and thus are judged to be good clusters. The overall node-edge density plays an important role as well since the entire network has a high node-edge density, Q metric and Relative Density expect highly dense clusters to be found and their absence results in low values for these metrics. As mentioned in the introduction, there are a few nodes that have a very high number of connections, airports such as Paris, London and New York, which increases the overall density of the network, but most of the airports have a very low number of connections. Thus many clusters found are representatives of regional or with-in country airports connecting all its cities, as shown in Fig 3. These results are a good justification of why the CPL is a good cluster evaluation metric as it does not rate the quality of such clusters poorly as compared to the other metrics.

Next, we look at the Internet Network. Almost all the evaluation metrics rate the quality of clustering highly for the three clustering algorithms except for the Strength clustering-Q metric value. Again, we refer to the overall node-edge density of this graph which is quite low. Due to this, Q metric and Relative Density do not expect highly dense clusters and thus even though there are lots of star-like clusters found in this network, their quality is rated as good.

Finally the analysis of the Protein network is quite close to that of the Airport network. The overall density is not that high, but still the node-edge ratio is 1:3.

Table 2. Evaluating the quality of clustering real world data sets using the existing and the proposed cluster evaluation technique

		Cluster Quality Metric			
Data Set	Clustering Algorithm	CPL	MQ	Q	Relative Density
Co-Authorship	Divisive Clustering	0.672	0.531	0.772	0.630
	Bisecting K-Means	0.589	0.425	0.775	0.636
	Strength Clustering	0.846	0.832	0.264	0.232
Air Traffic	Divisive Clustering	0.614	0.399	0.093	0.105
	Bisecting K-Means	0.499	0.238	0.012	0.122
	Strength Clustering	0.676	0.528	0.024	0.078
Internet	Divisive Clustering	0.498	0.324	0.790	0.697
	Bisecting K-Means	0.581	0.415	0.592	0.582
	Strength Clustering	0.666	0.503	0.356	0.554
Protein	Divisive Clustering	0.527	0.315	0.638	0.498
	Bisecting K-Means	0.595	0.410	0.336	0.316
	Strength Clustering	0.683	0.529	0.165	0.291

The network is a good mix of some highly dense clusters and some star-like and/or chain-like clusters. The strength algorithm again generates a very high number of clusters (169) as compared to Divisive (91) and Bisecting K-Means (117). The divisive algorithm has the lowest number of clusters and thus has relatively high Q metric and Relative Density values.

For all the different data sets and algorithms, the CPL metric assigns high values consistently. This is an indication that by definition and from previous experimental results on a wide variety of data sets, these algorithms perform well in grouping similar items together. The Q metric and the Relative density are heavily dependent on the overall node-edge density for the evaluation of a clustering. In case of high node-edge density, these metrics expect highly dense clusters and in case of low node-edge density, less dense clusters can be rated as high quality irrespective of the underlying cluster topology, where we have argued that Mutuality and Compactness should be taken into consideration. The CPL metric is consistent with algorithms and dense data sets where tightly connected clusters are expected as is the case with the co-authorship network and to some extent, the protein network.

We would like to mention that the experimentation and the results described in this paper compare different cluster evaluation techniques and should not be generalized to compare the different clustering algorithms. This is because the number of clusters and their sizes vary from one clustering algorithm to the other. Specially, Bisecting K-Means and Divisive Clustering based on Edge Centrality can not be compared with the Strength clustering algorithm in terms of performance and quality of clusters generated as strength clustering generates many small size clusters as compared to the other clustering algorithms.

5 Conclusion and Future Research Directions

In this paper we introduced a new metric called the CPL metric to evaluate the quality of clusters produced by clustering algorithms. We argued that Density and Cut Size based metrics play an important role in the evaluation of dense graphs but Mutuality and Compactness are also important for the evaluation of clusters in graphs that are not densely connected. The proposed metric takes into account the underlying network structure and considers the average path length as an important factor in evaluating the quality of a cluster. We evaluated the performance of some existing cluster evaluation techniques showing that the new metric actually performs better than the metrics used largely by the research community.

As part of future work, we intend to extend the metric to evaluate the quality of hierarchical clustering algorithms based on the principles introduced in this paper. A more extended study is needed to compare different clustering algorithms for data sets having varying network topologies to comprehend the behavior of different clustering algorithms which in turn can lead us towards a better understanding of how to judge these algorithms.

References

1. Auber, D., Chiricota, Y., Jourdan, F., Melancon, G.: Multiscale visualization of small world networks. In: INFOVIS 2003: Proceedings of the IEEE Symposium on Information Visualization, pp. 75–81 (2003)
2. Brandes, U., Erlebach, T.: Network Analysis: Methodological Foundations. LNCS. Springer, Heidelberg (March 2005)
3. Brandes, U., Gaertler, M., Wagner, D.: Engineering graph clustering: Models and experimental evaluation. ACM Journal of Experimental Algorithmics 12 (2007)
4. Corneil, D.G., Gotlieb, C.C.: An efficient algorithm for graph isomorphism. Journal of the ACM (JACM) 17, 51–64 (1970)
5. Gavin, A.-C., Bosche, M., Krause, R., Grandi, P., Marzioch, M., Bauer, A., Schultz, J., Rick, J.M., Michon, A.-M., Cruciat, C.-M., Remor, M., Hofert, C., Schelder, M., Brajenovic, M., Ruffner, H., Merino, A., Klein, K., Hudak, M., Dickson, D., Rudi, T., Gnau, V., Bauch, A., Bastuck, S., Huhse, B., Leutwein, C., Heurtier, M.-A., Copley, R.R., Edelmann, A., Querfurth, E., Rybin, V., Drewes, G., Raida, M., Bouwmeester, T., Bork, P., Seraphin, B., Kuster, B., Neubauer, G., Superti-Furga, G.: Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature 415(6868), 141–147 (2002)
6. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Proc. Natl. Acad. Sci. USA 99, 8271–8276 (2002)
7. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: Cluster validity methods: Part i. ACM SIGMOD Record 31, 2002 (2002)
8. Halkidi, M., Vazirgiannis, M.: Clustering validity assessment: Finding the optimal partitioning of a data set (2001)
9. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. 31(3), 264–323 (1999)
10. Kannan, R., Vempala, S., Vetta, A.: On clusterings good, bad and spectral. Journal of the ACM 51(3), 497–515 (2004)
11. Lacroix, V., Fernandes, C., Sagot, M.-F.: Motif search in graphs: Application to metabolic networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(4), 360–368 (2006)
12. Maimon, O., Rokach, L.: Data Mining and Knowledge Discovery Handbook. Springer, Heidelberg (September 2005)
13. Mihail, M., Gkantsidis, C., Saberi, A., Zegura, E.: On the semantics of internet topologies, tech. rep. gitcc0207. Technical report, College of Computing, Georgia Institute of Technology, Atlanta, GA, USA (2002)
14. Milligan, G.W.: A monte-carlo study of 30 internal criterion measures for cluster-analysis. Psychometrica 46, 187–195 (1981)
15. Mitchell, B., Mancoridis, S., Yih-Farn, C., Gansner, E.: Bunch: A clustering tool for the recovery and maintenance of software system structures. In: International Conference on Software Maintenance, ICSM (1999)
16. Newman, M.E., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E Stat. Nonlin. Soft Matter Phys. 69(2 Pt. 2) (February 2004)
17. Newman, M.E.J.: Fast algorithm for detecting community structure in networks. Physical Review E 69, 066133 (2004)
18. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) 74(3) (2006)

19. Nguyen, Q.H., Rayward, Smith, V.J.: Internal quality measures for clustering in metric spaces. Int. J. Bus. Intell. Data Min. 3(1), 4–29 (2008)
20. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66(336), 846–850 (1971)
21. Rozenblat, C., Melançon, G., Koenig, P.-Y.: Continental integration in multilevel approach of world air transportation (2000-2004). Networks and Spatial Economics (2008)
22. Schaeffer, S.E.: Graph clustering. Computer Science Review 1(1), 27–64 (2007)
23. Steinbach, M., Karypis, G., Kumar, V.: A comparison of document clustering techniques. Technical report, Departement of Computer Science and Engineering, University of Minnesota (2000)
24. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge (1994)

Finding Irregularly Shaped Clusters Based on Entropy

Angel Kuri-Morales[1] and Edwin Aldana-Bobadilla[2]

[1] Department of Computation,
Autonomous Technological Institute of Mexico,
Rio Hondo No. 1,
Mexico City, Mexico
akuri@itam.mx

[2] Institute of Research in Applied Mathematics and Systems,
Autonomous University of Mexico,
University City, Mexico City, Mexico
ealdana@uxmcc2.iimas.unam.mx

Abstract. In data clustering the more traditional algorithms are based on similarity criteria which depend on a metric distance. This fact imposes important constraints on the shape of the clusters found. These shapes generally are hyperspherical in the metric's space due to the fact that each element in a cluster lies within a radial distance relative to a given center. In this paper we propose a clustering algorithm that does not depend on simple distance metrics and, therefore, allows us to find clusters with arbitrary shapes in n-dimensional space. Our proposal is based on some concepts stemming from Shannon's information theory and evolutionary computation. Here each cluster consists of a subset of the data where entropy is minimized. This is a highly non-linear and usually non-convex optimization problem which disallows the use of traditional optimization techniques. To solve it we apply a rugged genetic algorithm (the so-called Vasconcelos' GA). In order to test the efficiency of our proposal we artificially created several sets of data with known properties in a tridimensional space. The result of applying our algorithm has shown that it is able to find highly irregular clusters that traditional algorithms cannot. Some previous work is based on algorithms relying on similar approaches (such as ENCLUS' and CLIQUE's). The differences between such approaches and ours are also discussed.

Keywords: clustering, data mining, information theory, genetic algorithms.

1 Introduction

Clustering is an unsupervised process that allows the partition of a data set X in k groups or *clusters* in accordance with a similarity criterion. This process is unsupervised because it does not require a priori knowledge about the clusters. Generally the similarity criterion is a distance metrics based in *Minkowsky Family of metrics* [1] which is given by:

$$d_{mk}(P,Q) = \sqrt[p]{\sum_{i=1}^{n} |P_i - Q_i|^p} \tag{1}$$

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 57–70, 2010.

where *P* and *Q* are two vectors in an *n*-dimensional space. From the geometric point of view, these metrics represent the spatial distance between two points. However, this distance is sometimes not an appropriate measure for our purpose. For this reason sometimes the clustering methods use statistical metrics such as *Mahalanobis'* [2], *Bhattacharyya's* [3] or *Hellinger's* [4], [5]. These metrics statistically determine the similarity of the probability distribution between random variables *P* and *Q*. In addition to a similarity criterion, the clustering process typically requires the specification of the number of clusters. This number frequently depends on the application domain. Hence, it is usually calculated empirically even though there are methodologies which may be applied to this effect [6].

1.1 A Hierarchy of Clustering Algorithms

A large number of clustering algorithms has been proposed which are usually classified as follows:

Partitional. Which discover clusters iteratively relocating iteratively elements of the data set between subsets. These methods tend to build clusters of proper convex shapes. The most common methods of this type are k-means [7], k-medoids or PAM (Partitioning Around Medoids) and CLARA (Clustering Large Applications) [8].

Hierarchical. In which large clusters are merged successively into smaller clusters. The result is a tree (called a *dendrogram*) whose nodes are clusters. At the highest level of the *dendrogram* all objects belong to the same cluster. At the lowest level each element of the data set is in its own unique cluster. Thus, we must select the adequate cut level such that the clustering process is satisfactory. Representative methods in this category are BIRCH [9], CURE and ROCK [10].

Density Based. In this category a cluster is a dense (in some pre-specified sense) region of elements of the data set that is separated by regions of low density. Thus, the clusters are identified as areas highly populated with elements of the data set. Here each cluster is flexible in terms of their shape. Representative algorithms of this category are DBSCAN [11] and DENCLUE [12].

Grid Based. Which use space segmentation through a finite number of cells and from these performs all operations. In this category are STING (Statistical Information Grid-based method) described by Wang et al. [13] and Wave Cluster [14].

Additionally, there are algorithms that use tools such as fuzzy logic or neural networks giving rise to methods such as Fuzzy C-Means [15] and Kohonen Maps [16], respectively. The performance of each method depends on the application domain. However, Halkidi [17] present several approaches that allow to measure the quality of the clustering methods via the so-called "quality indices".

1.2 Desired Properties of Clustering Algorithms

In general a good clustering method must:

–Be able to handle multidimensional data sets.
–Be independent of the application domain.

–Have a reduced number of settings.
–Be able to display computational efficiency.
–Be able to yield irregular shaped clusters.

With respect to last point, the great majority of the clustering methods restrict the shape of the clusters to hyperspherical shapes (in the space of the metric) owing to the use of some sort of distance as a similarity criterion. The distance between each point inside a cluster and its center is smaller than the radius of an *n*-dimensional sphere as illustrated in Figure 1 (for n=2).

An ideal case would allow us to obtain arbitrary shapes for the clusters that adequately encompass the data. Figure 2 illustrates this fact.

Therefore, we propose a clustering algorithm that better approaches the problem of finding clusters with irregular shapes.

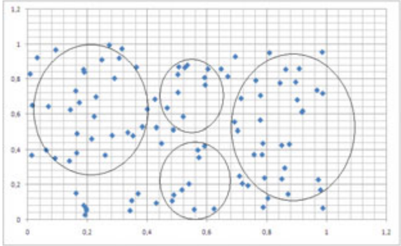


Fig. 1. Clusters with circular shapes

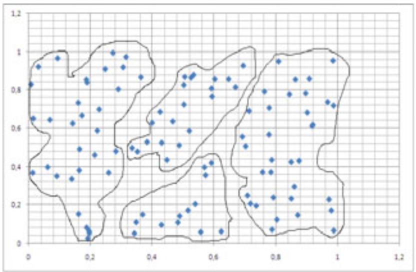


Fig. 2. Clusters with arbitrary shapes

2 Related Work

Our proposal is based on maximizing density in terms of the entropy of the area of the space that represents a cluster. There are previous works with similar approaches. Cheng et al [18], for example, present an algorithm called ENCLUS (Entropic Clustering) in which they link the entropy with two concepts that the authors call *coverage* and *density*. These are determined by the space's segmentation. Segmentation is made iteratively. Henceforth, several conditions have to be satisfied for every iteration of the algorithm. The space segmentation is a partition on non-overlapping rectangular units based on CLIQUE (Clustering in Quest) algorithm where a unit is dense if the fraction of the elements contained in the unit is greater than a certain threshold. A cluster is the maximum set of connected dense units. Another work is the so-called COOLCAT algorithm [19] which also approaches the clustering problem on entropic considerations but is mainly focused on categorical sets of data. The difference of our proposal is that the space is quantized through a hypercube that encapsulates all elements of the data set. The hypercube is composed of units of quantization that called "hypervoxels" or, simply, "voxels". The number of voxels determines the resolution of the hypercube. Contrary to ENCLUS, our algorithm does not iterate to find the optimal space quantization. Here the hypercube is unique and its resolution is given a priori as a parameter. The units of quantization become the symbols of the source's alphabet which allow an analysis through information theory. Our working hypothesis is that areas with high density have minimum entropy with respect to areas with low density.

3 Generalities

In what follows we make a very brief mention of most of the theoretical aspects having to do with the proper understanding of our algorithm. The interested reader may see the references.

3.1 Information Theory

Information theory addresses the problem of collecting and handling data from a mathematical point of view. There are two main approaches: the statistical theory of communication (proposed by Claude Shannon [20]) and the so-called algorithmic complexity (proposed by Andrei Kolmogorov [21]). In this paper we rely on the statistical approach in which information is a series of symbols that comprise a *message,* which is produced by an *information source* and is received by a *receiver* through a *channel.*

Where:

Message. It is a finite succession or sequence of symbols.

Information Source. It is a mathematical model denoted by S which represents an entity which produces a sequence of symbols (message) randomly. The space of all possible symbols is called source alphabet and is denoted as Σ (see [22]).

Receiver. It is the end of the communication's channel which receives the message.

Channel. It is the medium used to convey a *message* from an *information source* to a *receiver.*

In this document we apply two key concepts which are very important for our proposal.

Self Information. It is the information contained in a symbol s_i, which is defined as[1]:

[1] The base for the logarithms is arbitrary. When (as above) we choose base 2 the information is measured in "bits".

$$I(s_i) = -\log_2 p(s_i) \tag{2}$$

Where $p(s_i)$ is the probability that the symbol s_i is generated by the source S. We can see that the information of a symbol is greater when its probability is smaller. Thus, the self information of a sequence of statistically independent symbols is:

$$I(s_1 s_2 \dots s_n) = I(s_1) + I(s_2) + \cdots + I(s_n) \tag{3}$$

Entropy. The entropy is the expected value of the information of the symbols generated by the source S. This value may be expressed as:

$$H(S) = \sum_{i=1}^{n} p(s_i) I(s_i) = -\sum_{i=1}^{n} p(s_i) \log_2 p(s_i) \tag{4}$$

Where n is the size of the alphabet Σ. Therefore, we see that entropy is greater the more uniform the probability distribution of symbols is.

3.2 Genetic Algorithms

Genetic Algorithms (GA) (a very interesting introduction to genetic algorithms and other evolutionary algorithms may be found in [23]) are optimization algorithms which are frequently cited as "partially simulating the process of natural evolution". Although this is a suggestive analogy behind which, indeed, lies the original motivation for their inception, it is better to understand them as a kind of algorithms which take advantage of the implicit (indeed, unavoidable) granularity of the search space which is induced by the use of the finite binary representation in a digital computer.

In such finite space numbers originally thought of as existing in $\Re^n$ actually map into $\boldsymbol{B}^m$ space. Thereafter it is simple to establish that a genetic algorithmic process is a finite Markov chain (MC) whose states are the populations arising from the so-called genetic *operators*: (typically) selection, crossover and mutation. As such they display all of the properties of a MC. From this fact one may infer the following mathematical properties of a GA: 1) The results of the evolutionary process are independent of the initial population and 2) A GA preserving the best individual arising during the process will converge to the global optimum (albeit the convergence process is not bounded in time). For a proof of these facts the interested reader may see [24]. Their most outstanding feature is that, as opposed to other more traditional optimization techniques, the GA iterates simultaneously over *several* possible solutions. Thereafter, other plausible solutions are obtained by combining (*crossing over*) the *codes* of these solutions to obtain hopefully better ones. The solution space (SS) is, therefore, traversed stochastically searching for increasingly better plausible solutions. In order to guarantee that the SS will be globally explored some bits of the encoded solution are randomly selected and changed (a process called *mutation*). The main concern of GA-practitioners (given the fact that well designed GAs, in general, will find the best solution) is to make the convergence as efficient as possible. The work of Forrest et al. has determined the characteristics of the so-called *Idealized GA* (IGA) which is impervious to GA-hard problems [25].

3.3 Vasconcelos' Genetic Algorithms

The implementation of the IGA is unattainable in practice. However, a practical approximation called the Vasconcelos' GA (VGA) has been repeatedly tested and proven to be highly efficient [26]. The VGA, therefore, turns out to be an optimization algorithm of broad scope of application and demonstrably high efficiency.

A statistical analysis was performed by minimizing a large number of functions and comparing the relative performance of six optimization methods[2] of which five are GAs. The ratio of every GA's absolute minimum (with probability *P=0.95*) relative to the best GA's absolute minimum may be found in Table 1 under the column "Relative Performance". The number of functions which were minimized to guarantee the mentioned confidence level is shown under "Number of Optimized Functions".

Table 1. Relative Performance of Different Breeds of Genetic Algorithms

Algorithm	Relative Performance	Number of Optimized Functions
VGA	1.000	2,736
EGA	1.039	2,484
TGA	1.233	2,628
SGA	1.236	2,772
CGA	1.267	3,132
RHC	3.830	3,600

It may be seen that the so-called Vasconcelos' GA (VGA) in this study was the best of all the analyzed variations. Interestingly the CGA (the classical or "canonical" genetic algorithm) comes at the bottom of the list with the exception of the random mutation hill climber (RHC) which is not an evolutionary algorithm. According to these results, the minima found with the VGA are, on the average, more than 25% better than those found with the CGA. Due to its tested efficiency, we now describe in more detail the VGA.

Outline of Vasconcelos' Genetic Algorithm (VGA)

1. Generate random population of *n* individuals (suitable solutions for the problem).
2. Evaluate the fitness *f(x)* of each individual *x* in the population.
3. Order the *n* individuals from best (top) to worst (bottom) for *i=1, 2,. . . , n* according to their fitness.
4. Repeat steps A-D (see below) for $i = 1,2,...,\lfloor n/2 \rfloor$.

   A. *Deterministically* select the *i-th* and the $(n - i + 1) - th$ individuals (the *parents*) from the population.

[2] VGA: Vasconcelos' GA; EGA: Eclectic GA; TGA: Elitist GA; SGA: Statistical GA; CGA: Canonical (or Simple) GA; RMH: Random Mutation Hill Climber.

B. With probability *Pc* cross over the selected parents to form two new individuals (the *offspring*). If no crossover is performed, offspring are an exact copy of the parents.
C. With probability *Pm* mutate new offspring at each locus (position in individual).
D. Add the offspring to a new population

5. Evaluate the fitness *f(x)* of each individual *x* in the new population
6. Merge the newly generated and the previous populations
7. If the end condition is satisfied, stop, and return the best solution.
8. Order the *n* individuals from best to worst (*i=1, 2, . . . , n*) according to their fitness
9. Retain the top *n* individuals; discard the bottom *n* individuals
10. Go to step 4

As opposed to the CGA, the VGA selects the candidate individuals deterministically picking the two extreme (ordered according to their respective fitness) performers of the generation for crossover. This would seem to fragrantly violate the survival-of-the-fittest strategy behind evolutionary processes since the genes of the more apt individuals are mixed with those of the least apt ones. However, the VGA also retains the best *n* individuals out of the *2n* previous ones. The net effect of this dual strategy is to give variety to the genetic pool (the lack of which is a cause for slow convergence) while still retaining a high degree of elitism. This sort of elitism, of course, guarantees that the best solutions are not lost. On the other hand, the admixture of apparently counterpointed plausible solutions is aimed at avoiding the proliferation of similar genes in the pool. In nature as well as in GAs variety is needed in order to ensure the efficient exploration of the space of solutions[3]. As stated before, all GAs will eventually converge to a global optimum. The VGA does so in less generations. Alternatively we may say that the VGA will outperform other GAs given the same number of generations. Besides, it is easier to program because we need not to simulate a probabilistic process. Finally, the VGA is impervious to negative fitness's values.We, thus, have a tool which allows us to identify the best values for a set of predefined metrics possibly reflecting complementary goals.

For these reasons we use in our work the VGA as the optimization method. In what follows we explain our proposal based in the concepts mentioned above.

4 Evolutionary Entropic Clustering

Let *X* be a data set of elements x_i such that $x_{i=}\{x_{i1}, x_{i2}, \dots, x_{in}\}$, let *D* be an *n*-dimensional space such that $xi \in D$ and let c_j be a subset of D called cluster. Then we must find a function that associates each element of *X* to the *j-th* cluster c_j as:

$$f(x_i) = c_j;\ \forall x_i \in X \wedge 2 \le j \le k \tag{5}$$

[3] The Latin American philosopher José Vasconcelos proposed that the admixture of all races would eventually give rise to a better one he called the *cosmic* race; hence the algorithm's name.

Where k is the number of clusters and $f(x_i)$ is called the membership function. Now we describe a method which attempts to identify those elements within the data set which share common properties. These properties are a consequence of (possibly) high order relationships which we hope to infer via the entropy of a quantized vector space. This space, in what follows, will be denoted as the *Hypercubic Wrapper*.

4.1 Hypercubic Wrapper

A *Hypercubic Wrapper* denoted as HW is an n-dimensional subspace of D such that:

$$x_i \in HW \quad \forall x_i \in X \tag{6}$$

HW is set of elements v_m called voxels, which are units in n-dimensional that can contain zero or more elements of the set X. The cardinality of HW depends on the maximum number of voxels that we specify in each dimension of the space D such that:

$$|HW| = \prod_{i=1}^{n} L_i \tag{7}$$

Where L_i is the number of voxels in the *i-th* dimension and n is the dimension number of D. From equation (7) it follows that $\forall v_m \in HW$:

$$0 < m \leq \prod_{i=1}^{n} L_i \tag{8}$$

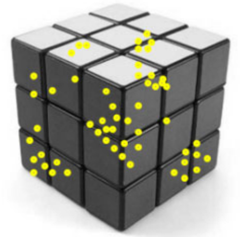


Fig. 3. Hypercubic Wrapper in a *tri*-dimensional space where the points represent elements of the data set and the subdivisions are voxels

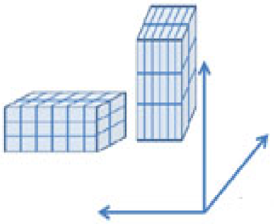


Fig. 4. Hypercubes with different lengths per dimension

Figure 4 shows a graphical representation of *HW* in a *tri*-dimensional space, where the cardinality is equal to 27 (since $L_i=3$ for all *i*).

In general, $L_i \neq L_j \ \forall i, j$ and it is, therefore, possible to define different *HWs* by changing the values of the L_i's as shown in Figure 4

Once this wrapper's characteristics are defined, we may use a clustering method based in the concept of *Self Information*, *Entropy*, and *VGA*. We describe our proposal (which we call the Fixed Grid Evolutionary Entropic Algorithm (FGEEA)).

4.2 Fixed Grid Evolutionary Entropic Algorithm

Definition 1. Every voxel that includes at least one element of the data set *X* is called a non-empty voxel; otherwise it is called an empty voxel

Definition 2. For the purpose of entropy calculation, any non-empty voxel is identified with the *i-th* symbol and will be denoted by s_i.

Definition 3. The space of all symbols is an alphabet denoted by Σ.

Definition 4. The data set is equivalent the source of information S. Such source only produces symbols of Σ.

Definition 5. The probability that the symbol s_i is produced by *S* is the cardinality of *S* divided by the cardinality of the data set X. This probability is denoted as $p(s_i)$.

Corollary. The density of a symbol s_i is proportional to its probability $p(s_i)$.

It follows that:

$$|\Sigma| \leq |HW| \tag{9}$$

$$H(S) = \sum_{i=1}^{n} p(s_i) I(s_i) = -\sum_{i=1}^{n} p(s_i) \log_2 p(s_i) \tag{10}$$

Our idea is to use the entropy for determine the membership of every symbol in the *j-th* cluster, based on the follows assumptions:

Assumption 1. The source *S* always produces the same symbols for a given data set *X*.

Assumption 2. The spatial position of each symbol (voxel) is invariant for a given data set *X*. Therefore, *H(S)* is constant.

According to the working hypothesis, the areas with high density have minimum entropy with respect to areas with low density. Therefore the areas with minimum entropy correspond to a cluster.

To determine the entropy of a cluster we introduce a concept we call *intracluster entropy*, defined as:

$$H(c_i) = -\sum p(s_j) \log_2 p(s_j) \ \ \forall \ s_j \in \ c_i \tag{11}$$

Where $H(c_i)$ is the intracluster entropy of *i-th* cluster. In order to determine that s_j belongs to c_i we use a genetic algorithm, as discussed in what follows.

Application of Vasconcelos' Genetic Algorithm

Our aim is to find the areas of the subspace *HW* where the entropy is minimal. We find groups of voxels such as each group have minimal entropy (intracluster entropy). Clearly this is an optimization problem. For reasons already discussed we use a VGA. The individuals of the algorithm have been encoded as follows. a) The length of the genome is equal to the cardinality of Σ. [It is composed by all symbols (or *genes*)]. b) Each gene is assigned a label that represents the cluster to which it belongs. c) It has a sequential index. Such index will allow mapping all symbols to subspace HW. Fig.5 exemplifies a genome for *k=3*.

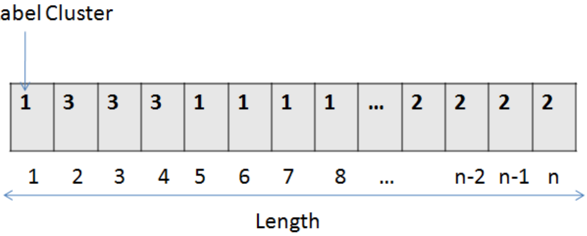


Fig. 5. Genome of the individual (*k=3*)

Now, we defined the fitness function as:

$$f(individual_j) = min \sum_{i=0}^{k} H(c_i) \quad for\, j \leq N \tag{12}$$

Subject to:

$$\sum_{i=0}^{k} H(c_i) \geq H(S) \tag{13}$$

$$\left|\sum_{i=0}^{k} H(c_i) - H(S)\right| \geq \Delta_1 \tag{14}$$

Where *N* is the size of the population and Δ_1 is a parameter that represents a threshold of the difference between the sum of intracluster entropies and the entropy of source *S*. Additionally we have introduced a constraint called *intracluster density* defined as:

$$dc_i \leq \varepsilon \tag{15}$$

where ε is the threshold density. One last constraint is the intracluster density (dc_i). It is the number of elements of data set *X* which belong to the symbols of *i-th* cluster:

$$dc_i = \frac{\alpha}{\beta} \tag{16}$$

where α is the number the elements that belong to the data set *X* and β is the number of symbol within cluster *i*. This constraint ensures that entropy is minimal within any given cluster. The algorithm yields a best individual which represents a set of clusters of symbols that are map into sets of voxels in the subspace *HW*, as shown in Fig. 6.

In what follows we show some experiments that allow us to test the effectiveness of the algorithm presented previously

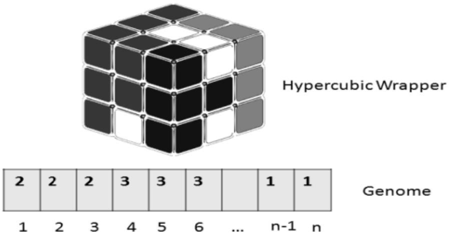


Fig. 6. Possible clustering delivered by the VGA. Different intensities in the cube represent a different cluster. (White voxels are empty).

5 Experimental Results

Our algorithm was tested with a synthetic data set which consists of a set of points contained by three disjoint spheres. The features and parameters of the first test are given in Table 2. The values of the parameters were determined experimentally.

The VGA was run 20 times (with different seeds of the pseudo random number generator) yielding an average effectiveness of 98%. Notice that no information other than the number of clusters is fed to FGEEA.

Table 2. Features and parameters first test

Feature	Value	Parameter	Value
Sample size	192	N (Number of individuals)	500
Elements per cluster	64	G (Generations)	1000
Dimensions	3	Pm (Mutation Probability)	0.001
Data Distribution	Disjoint sphere	Pc (Crossover Probability)	0.99
		Δ_i	$3.5<\Delta_i<3.6$
Cardinality of ∑	26	ε	5

Table 3. Results obtained with Kohonen Maps and Fuzzy C-Means

Algorithm	Average Effectiveness
Kohonen Maps	0.99
Fuzzy C- Means	0.98

The same data set was tested with other algorithms such as *Kohonen Maps* and *Fuzzy C-Means*. The results obtained are shown in Table 3.

This us allow see that the result of our proposal is similar to result given by some alternative algorithms. The high effectiveness in all cases is probably due to the spatial distribution of data set.

Next, we test with other data set whose spatial distribution yields presents overlapping clusters as is shown in Fig. 7. For clarity we show a *bi*-dimensional example. The actual runs consisted of three dimensional data.

In this case the size sample is 192. Its elements are, a priori, distributed in three clusters. The cardinality is 64 in all cases. The results obtained are shown in Table 4.

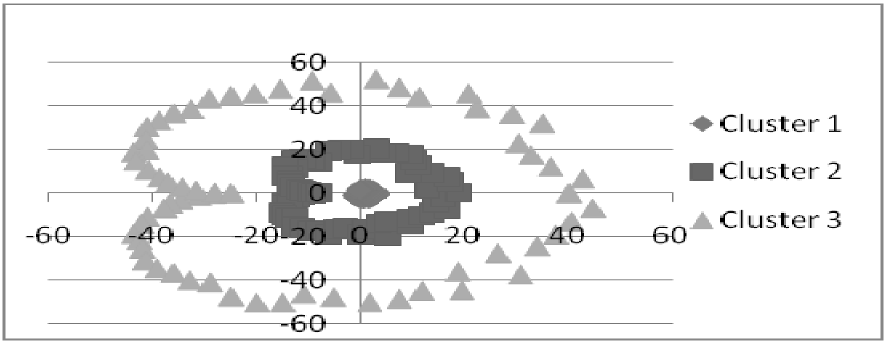


Fig. 7. Overlapping clusters

Table 4. Results of FGEEA with overlapping data set

Algorithm	Average Effectiveness
Kohonen Maps	0.62
Fuzzy C- Means	0.10
FGEEA	0.73

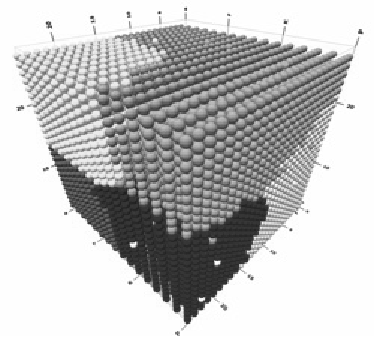


Fig. 8. Irregular clusters. The number of voxels is 15625 (25 voxels per dimension). The white voxels are empty.

Here the effectiveness decreases significantly in general. But FGEEA showed the better results.

Finally we tested our algorithm with a data set in tridimensional space with an unknown spatial distribution. For $k = 3$ (number of clusters) the algorithm found a solution that is shown in Fig. 8. Here the clusters are irregularly shaped.

These last results were not compared with other clustering algorithms. However, we can see that in principle our approach is feasible.

6 Conclusions and Future Work

These results allow us to test the feasibility of our algorithm. This is not enough, however, to assume its effectiveness in general. To achieve this proof we require testing with several data sets and applying more solid clustering validation techniques. Computationally, the analysis of the geometric and spatial membership relation between elements of a multidimensional data set is hard. Our approach showed that in principle, membership relations in a data set can be found through of its entropy without an excessive demand on computational resources. Even though the results obtained are limited (since they correspond to particular cases and in tri-dimensional data) they are promissory. Therefore, future work requires to generalize our method for a data set in n-dimensional space (with $n>3$), to analyze its computational complexity and to test its detailed mathematical formulation. We will report on these issues shortly.

References

1. Cha, S.H.: Taxonomy of Nominal Type Histogram Distance Measures, Massachusetts (2008)
2. Mahalanobis, P.C.: On the genaralized distance in statistics (1936)
3. Bhattacharyya, A.: On a measure of divergence between two statistical populations defined by probability distributions, Calcutta (1943)
4. Pollard, D.E.: A user's guide to measure theoretic probability. Cambridge University Press, Cambridge (2002)
5. Yang, G.L., Le Cam, L.M.: Asymptotics in Statistics: Some Basic Concepts. Springer, Berlin (2000)
6. Li, X., Wai, M., Kwong Li, C.: Determining the Optimal Number of Clusters by an Extended RPCL Algorithm. Hong Kong Polytechnic University, Hong Kong (1999)
7. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of 5th Berkley Sysmposium on Mathematical Statiscs and Probability, Berkley, pp. 281–297 (1967)
8. Ng, R., Han, J.: Effecient and Effective Clustering Methods for Spatial Data Mining, Santiago de Chile (1994)
9. Zhang, T., Ramakrishnman, R., Linvy, M.: BIRCH: An Efficient Method for Very Large Databases, Montreal, Canada (1996)
10. Guha, S., Rastogi, R., Shim, K.: An efificient Clustering Algorithm for Large Databases (1998)

11. Ester, M., Kriegel, H., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, Portland, pp. 226–223 (1996)
12. Hinneburg, A., Keim, D.: An Efficient Approach to Clustering in Large Multimedia Databases with noise (2000)
13. Wang, W., Yang, J., Muntz, R.: STING: A Statistical Information Grid Approach to Spatial Data. In: Proceedings of the 23rd VLDB Conference, Athens (1997)
14. Sheikholeslami, G., Chatterjee, S., Zhang, A.: Wavecluster: A multi-resolution clustering. In: Proceedings of the 24th VLDB conference (1998)
15. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters, pp. 32–57 (1973)
16. Kohonen, T.: Self-Organizing Maps. Series in Information Sciences (1995)
17. Halkidi, M., Batistakis, Y., Vzirgiannis, M.: On Clustering Validation Techniques, pp. 107-145 (2001)
18. Cheng, C., Fu, A.W., Zhang, Y.: Entropy- based Subspace Clustering for Mining Numerical Data (1998)
19. Barbará, D., Julia, C., Li, Y.: COOLCAT: An entropy-based algorithm for categorical clustering, George Mason University (2001)
20. Shannon, C.E.: A mathematical theory of communication, pp. 379–423 (1948)
21. Kolmogorov, A.N.: Three approaches to the quantitative definition of information, pp. 1–7 (1948)
22. Gray, R.M.: Entropy and Information Theory. Springer, Heidelberg (2008)
23. Bäck, T.: Evolutionary Algorithms in Theory and Practice. Oxford University Press, Oxford (1996)
24. Rudolph, G.: Convergence Analysis of Canonical Genetic Algorithms. IEEE Transactions on Neural Networks (1994)
25. Forrest, S., Mitchell, M.: What makes a problem hard for a genetic algorithm? Machine Learning (1993)
26. Kuri, A.: A Methodology for the Statistical Characterization of Genetic Algorithms, pp. 79–88. Springer, Heidelberg (2002)

Fuzzy Conceptual Clustering

Petra Perner and Anja Attig

Institute of Computer Vision and applied Computer Sciences, IBaI
Leipzig, Germany
pperner@ibai-institut.de
www.ibai-institut.de

Abstract. Grouping unknown data into groups of similar data is a necessary first step for classification, indexing of data bases, and prediction. Most of today's applications, such as news classification, blog indexing, image classification, and medical diagnosis, obtain their data in temporal sequence or on-line. The necessity for data exploration requires a graphical method that allows the expert in the field to study the determined groups of data. Therefore, incremental hierarchical clustering methods that can create explicit cluster descriptions are convenient. The noisy and uncertain nature of the data makes it necessary to develop fuzzy clustering methods. We propose a novel fuzzy conceptual clustering algorithm. We describe the fuzzy objective function for incremental building of the clusters and the relation among the clusters in a hierarchy. The operations that can incrementally re-optimize the fuzzy-based hierarchy based on the newly arrived data are explained. Finally, we evaluate our method and present results.

1 Introduction

Grouping unknown data into groups of similar data is a necessary first step for classification, indexing of data bases, and prediction. Most of today's-applications, such as news classification, blog-indexing, image classification, and medical diagnosis, obtain their data in temporal sequence or on-line. The necessity for data exploration requires a graphical method that allows the expert in the field to study the determined groups of data and their relations. Therefore, incremental hierarchical clustering methods that can create explicit cluster descriptions are convenient. The noisy and uncertain nature of the data makes it necessary to develop fuzzy clustering methods. We propose a novel fuzzy conceptual clustering algorithm. This algorithm can generate clusters and the relations among the clusters based on the actual available data. As soon as new data arrive, the algorithm is able to update the clusters and the hierarchy based on the new data without reconsidering the old data.

To ensure this behavior we developed hierarchy-construction operators that check based on a fuzzy objective-function if a new item should either be placed into existing groups or create a new group or merge or split groups. The algorithm can incrementally learn the cluster hierarchy from the arriving data and redesign and optimize the cluster hierarchy. The explicit concept-description of a cluster is worked out during the clustering process and updated according to the new data. An off-line calculation process as it is done for conventional hierarchical clustering is thus not necessary. The

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 71–85, 2010.

hierarchical structure shows the clusters and their data from a coarse to grain level. Displayed on a graphical user interface, the hierarchical structure provides a domain expert with an excellent data exploration capability. The number of clusters must not be determined in advance; the algorithm is able to determine them automatically.

In Section 2 we describe related work. In Section 3, our novel fuzzy conceptual clustering algorithm is described. We describe the fuzzy objective function for incremental building of the clusters. The operations that can incrementally re-optimize the fuzzy-based cluster hierarchy based on the newly arrived data are explained. In Section 4, we present the data we used for the evaluation of our algorithm and the ordering of the data. We evaluate our method and present results in Section 5. Finally, we discuss the characteristic of the algorithm in Section 6 and present directions for further studies. Conclusions are given in Section 7.

2 Related Work

Fuzzy c-means (FCM) is a method of clustering, which allows to group data according to its degree of membership in two or more groups. It takes into account the uncertainty of the data and can create overlapping clusters, not only crisp clusters. For many applications this representation of the problem solutions space is much more convenient since it considers the uncertain nature of the data existing for many applications. The method was developed by Dunn in 1973 [1] and improved by Bezdek in 1981 [2]. Fuzzy c-means is frequently used for many applications. Meanwhile many variations of the original fuzzy c-means algorithm have been developed. They can be roughly divided into fuzzy clustering based on fuzzy relations [5] and fuzzy clustering based on objective function [5][3]. For example, the Gustafson-Kessel-Algorithm [3] employs an adaptive distance norm. In their paper, Fadili et al. [6] review different objective fuzzy functions and finally use a function based on the ratio of a) the ratio of the fuzzy compactness and the average separation between the clusters and b) the fuzzy relationship between clusters in terms of fuzzy union and intersection.

The fuzzy c-mean is a partitioning algorithm and does not structure the clusters in hierarchical fashion; this is not convenient for explanation, classification or data base indexing purposes. The main drawback of the algorithm is that it cannot automatically decide on the number of clusters. The number of clusters has to be known or determined in advance. Otherwise, an optimization strategy has to be applied that selects the best number of clusters while iteratively applying the fuzzy c-mean with different numbers of clusters to the same data set [2][6].

The fuzzy c-mean does not consider the on-line nature of the data existing for many applications such as in text classification for newswires and blogs or image processing. The full data set has to be available for clustering. Once new data arrive, the data set has to be updated by the new data and the whole clustering procedure has to be repeated in order to obtain the new clusters underlying the updated data set.

A few hierarchical fuzzy clustering algorithms are known. However, hierarchical fuzzy clustering algorithms have not yet been extensively studied.

Torra [8] studied hierarchical clustering with the help of fuzzy c-mean for text document classification in newswires. In the top-down strategy the fuzzy c-mean are iteratively applied to the clusters that are too large. In the bottom-up strategy, the last clustering result is the starting set for the next fuzzy c-mean clustering. Therefore, the selection of the number of clusters for the fuzzy c-means is still a problem. The algorithm is not incremental because most clusters do not change if some new documents arrive as input.

Rodriques et al. [9] developed an algorithm for streaming time series. A top-down strategy is used to build a binary tree hierarchy with a semi-fuzzy assignment. This means that, the membership-function for a vector x_i to a cluster c is $u_{ic} = u_{ip}(n_{ip})^{-1}$, where p is the parent of c and n_{ip} the number of clusters to which the vector x_i was assigned (at level p)). The hierarchy is restricted to binary trees but the algorithm is incremental. However as described before, the fuzzy membership function is very simple.

Bordogna et al. [10] developed an (incremental) hierarchical fuzzy clustering algorithm for news-filtering. This algorithm has two modes. In the online mode news can belong either to an existing cluster or it creates a new cluster. During this mode only the membership functions are updated, but not the prototypes (centroids) of the clusters. In an offline mode the hierarchy is rebuilt based on the updated data set with a bottom up algorithm and the fuzzy c-mean using the cosine similarity. The online mode is incremental but the hierarchy building operator can only add new instances and update the membership function but not re-optimize the whole cluster hierarchy. After a while this strategy results in a non-optimal cluster hierarchy that must be redesigned during the off-line mode based on the old and newly collected data set.

All these hierarchical fuzzy clustering algorithms do not generate an explicit concept of a cluster. The concept has to be calculated after the hierarchy has been built.

Fuzzy conceptual clustering for text categorization is proposed by Quan et. al. [11]. The algorithm is a three step approach. First, a formal concept analysis of the data based on fuzzy logic is performed to create a fuzzy concept lattice. Next, a fuzzy conceptual clustering technique is proposed to cluster the fuzzy concept lattice into fuzzy clusters. In the third step, the hierarchical relations are generated among conceptual clusters for constructing the concept hierarchy. Although the concepts are explicit, the algorithm is not incremental.

Hierarchical conceptual clustering algorithms can create an explicit concept description for a cluster while forming the clusters and incrementally update the cluster-hierarchy based on the new data. Well known algorithms are COBWEB [12], UNIMEM [13], and CLASSIT [14]. These algorithms differ in the kind of data representations (numerical, symbolical or mixed data) they can handle, the kind of operations they can perform to re-optimize the hierarchical structure of the clusters, and the kind of objective function they use to decide which operations should be carried out next to update the hierarchical structure. They employ a similarity-based or a probabilistic concept description.

We propose a fuzzy conceptual clustering algorithm that can build explicit concepts while forming the hierarchy and update the hierarchy based on the new data.

3 The Algorithm

Hierarchical conceptual clustering constitutes learning the clusters incrementally. It enables flexibly learning the hierarchy of observations according to the arriving data stream. There is no need to know the number of groups in advance; this will be learnt while clustering the instances. The concept description of a class is explicit. We propose a fuzzy conceptual clustering algorithm. Therefore, we need a fuzzy concept building function and, in contrast to crisp concepts in a fuzzy framework, a case may belong to one or more classes with a membership value.

A fuzzy concept may be described as follow:

Let N be the number of samples $\{ x_1, x_2, ..., x_i, ..., x_N \}$ that can be partitioned into M clusters $\{ C_1, C_2, ..., C_k, ..., C_M \}$. The cluster may overlap and a particular sample x_t may belong to one or more clusters C_k with a particular membership degree u_{kt}:

$$u_{kt} = \left[\sum_{l=1}^{M} \left(\frac{d^2(x_t, m_k)}{d^2(x_t, m_l)} \right)^{\frac{1}{n-1}} \right]^{-1} \tag{1}$$

with $\sum_{k=1}^{M} u_{kt} = 1 \quad \forall t \in \{1, \ldots, N\}$ and $\sum_{t=1}^{N} u_{kt} > 0 \quad \forall k \in \{1, ..., M\}$. n is the degree of fuzziness (n=2) and $d^2(x, m)$ any similarity measure. We chose the squared Euclidean norm as similarity measure. The fuzzy prototype m_k of cluster *k* is

$$m_k = \frac{\sum_{t=1}^{N} u_{kt}^n x_t}{\sum_{t=1}^{N} u_{kt}^n} \tag{2}$$

with $k \in \{1, ..., M\}$.

A particular cluster C_l may be described by the prototype m_l, the samples $x_{l1}, x_{l2}, ..., x_{li}, ..., x_{lN}$ and the membership values u_{lt}:

$$C_l = \{(x_{l1}, u_{l1}), (x_{l2}, u_{l2}), ..., (x_{li}, u_{li}), ..., (x_{ls}, u_{li})\} \wedge \{m_l\} \tag{3}$$

In general the algorithm can be described as follow:

- Each new case is tentatively placed into the actual concept hierarchy level by level beginning with the root node until a terminal node is reached.

- In each hierarchy level one of these four different kinds of operations is performed:
- The case is incorporated into one or more existing child nodes with a membership value $u_{kt} > u^{*}$(tested $u^{*} = 0.1$ and $u^{*} = 0.2$),
- A new empty child node is created where the case is incorporated,
- A child node is split into grandchild nodes.

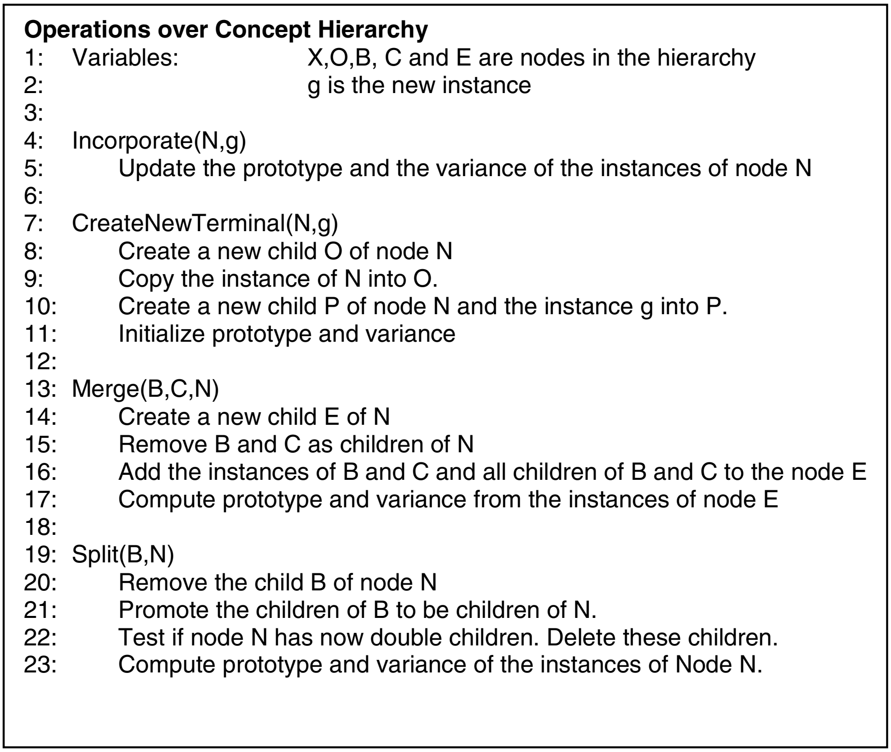
Which action is performed to create the hierarchy is decided based on a fuzzy objective function. Figures 1 and 2 show the overall algorithm.

```
Input:           Concept Hierarchy CB with current node N
                 An unclassified instance g
Output:          Modified Concept Hierarchy CB'
Top-level call:  ConceptHierarchy(top-node, g)
Variables:       A, B, C, D, E are nodes in the hierarchy
                 W, X, Y, Z are (clustering)partition scores

ConceptHierarchy(N,G)
    If N is a terminal node
            Then CreateNewTerminals(N,g)
                 Incorporate(N,g)
    Else
            For each child A of node N
                 Compute the score for placing g in A.
            Let B be the node with the highest score W.
            Let C be the node with the second highest score.
            Let X be the score for placing g in a new node D.
            Let Y be the score for merging B and C into one node E.
            Let Z be the score for splitting B into its children.

    If W is the best score
            Then ConceptHierarchy(B,G)(place g in case class B)
                 For each child A (without B) of node N
                     If $u_{Ag} > u^{*}$
                         Then ConceptHierarchy(A,G)
    Else If X is the best score
            Then Input an new node D
    Else If Y is the best score
            Then Merge(B,C,N)
                 Let E be the node from merging B and C.
                 ConceptHierarchy(E,g)
    Else If Z is the best score
            Then Split(B,N)
                 ConceptHierarchy(N,g)
```

Fig. 1. The Algorithms Part A



```
Operations over Concept Hierarchy
Variables:          X,O,B, C and E are nodes in the hierarchy
                    g is the new instance

Incorporate(N,g)
    Update the prototype and the variance of the instances of node N

CreateNewTerminal(N,g)
    Create a new child O of node N
    Copy the instance of N into O.
    Create a new child P of node N and the instance g into P.
    Initialize prototype and variance

Merge(B,C,N)
    Create a new child E of N
    Remove B and C as children of N
    Add the instances of B and C and all children of B and C to the node E
    Compute prototype and variance from the instances of node E

Split(B,N)
    Remove the child B of node N
    Promote the children of B to be children of N.
    Test if node N has now double children. Delete these children.
    Compute prototype and variance of the instances of Node N.
```

Fig. 2. The Algorithm Part B

3.1 Evaluation Function

The evaluation score expresses the average separation between the clusters and the fuzzy compactness as follow:

$$Score = \frac{1}{M}\sum_{k=1}^{M}\left(\frac{n_k}{N}d^2(m_k,\overline{x}) - \frac{1}{n_k\pi_k}\sum_{t=1}^{N}u_{kt}^{n}d^2\left(x_t,m_k\right)\right) \tag{4}$$

with $\pi_k = \sum_{t=1}^{N} u_{kt}$ the cluster cardinality, the membership degree u_{kt} (see formula 1), and M the number of clusters on the particular hierarchy level. The ratio $\frac{n_k}{N}$ and $\frac{1}{n_k}$ has been introduced to force the algorithm to prefer clusters with a large number of samples. To be able to perform the fuzzy clustering in incremental hierarchical fashion, we calculate the prototype by $m_k = \frac{1}{n_k}\sum_{t=1}^{n_k} x_t$ and not according to the fuzzy prototype in formula 2. It is clear that this is a limitation but our tests have shown that this calculation is not that far from being optimal (see Section 4.3). In case of fuzzy clustering the clusters may overlap. A sample may belong to several nodes at a particular level of the hierarchy. Therefore, this must be tested. For this purpose, we perform with the sample the operations with the four best scores under the condition that the membership is below a threshold *T*.

3.2 Reoptimization of the Hierarchy

The hierarchy is adapted to each single new instance. For this purpose, several optimization operators are needed to adapt the hierarchy to the new samples. There are five new operators. Figure 3 shows how to insert a new node into the hierarchy. Figure 4 shows how to create a terminal node while Figure 5 demonstrates merging two clusters and Figure 6 demonstrates splitting two clusters. Inserting a case into several nodes of a certain level of the hierarchy according of its membership values is presented in Figure 7.

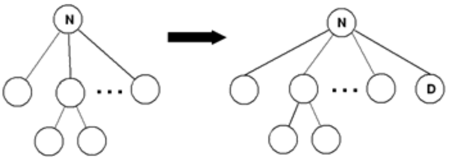


Fig. 3. Input a New Node

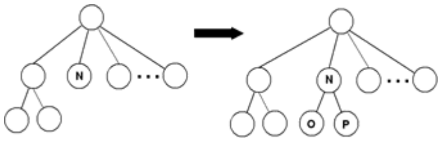


Fig. 4. Create a Terminal Node

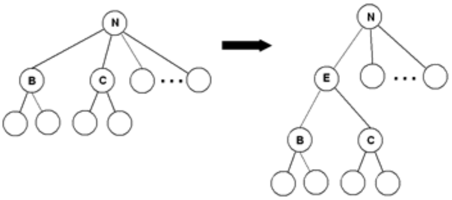


Fig. 5. Merge Cluster B and C into a New Node E

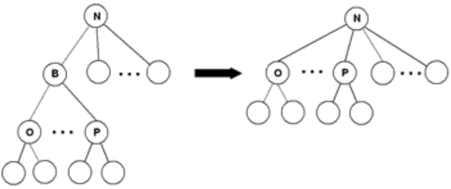


Fig. 6. Split Cluster B into Cluster O and P

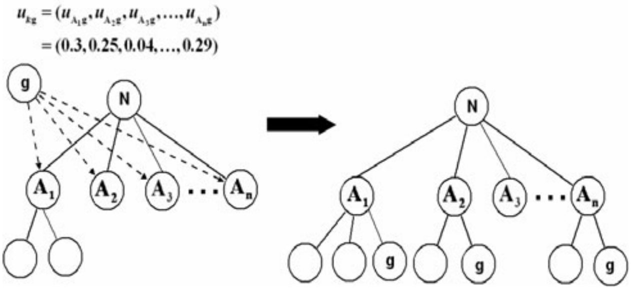

$u_{kg} = (u_{A_1g}, u_{A_2g}, u_{A_3g}, \ldots, u_{A_ng})$
$= (0.3, 0.25, 0.04, \ldots, 0.29)$
g
N
A1
A2
A3
An
N
A1
A2
A3
An
g
g
g

Fig. 7. Add a case to a one or more nodes

4 The Data Set

For testing we created a synthetic data set for a two-dimensional feature space. The diagram is shown in Figure 8. The real data values with their reference number are shown in table 1.

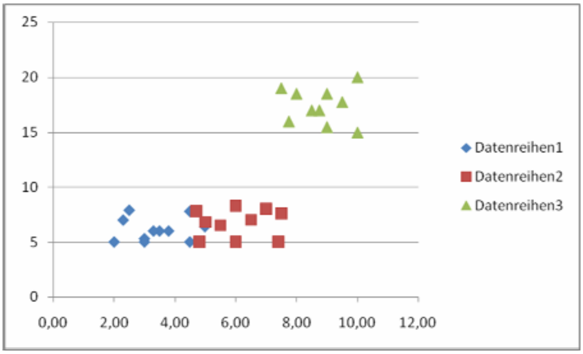


Fig. 8. Synthetic data set

Table 1. Data table with reference number for the data sample data and class

Class&Number	X1	X2
1-1	2,00	5
1-2	3,00	5,3
1-3	3,50	6
1-4	4,50	7,8
1-5	4,50	5
1-6	3,00	5
1-7	3,80	6
1-8	2,30	7
1-9	2,50	7,9
1-10	3,30	6
1-11	5,00	6,4
2-1	4,70	7,8
2-2	6,00	8,3
2-3	5,00	6,8
2-4	6,50	7
2-5	6,00	5
2-6	4,80	5
2-7	7,50	7,6
2-8	7,40	5
2-9	7,00	8
2-10	5,50	6,5
3-1	10,00	20
3-2	10,00	15
3-3	7,50	19
3-4	7,75	16
3-5	8,50	17
3-6	8,00	18,5
3-7	9,00	15,5
3-8	9,00	18,5
3-9	9,50	17,75
3-10	8,75	17

Table 2. Ordering_1 of the Samples

1-1,1-2,1-3,1-4,1-5,1-6,1-7,1-8,1-9,1-10,1-11 2-1, 2-2, 2-3, 2-4, 2-5, 2-6, 2-7, 2-8, 2-9, 2-10, 3-1, 3-2, 3-3, 3-4, 3-5,3-6,3-7,3-8,3-9,3-10

Table 3. Ordering_2 of the Samples

3-2,1-1,2-1,1-2,1-3,2-9,1-5,1-8,2-6,1-9,1-10,1-11,3-3,2-2,2-4,1-4,2-5,3-10,2-7,2-8,2-10,3-1,2-3,3-4,3-5,3-6,1-6,3-7,3-8,3-9,1-7

We chose two different orderings for the sample set. The ordering_1 is shown in table 2 that contains the samples of each class one followed by the other. This is the worst case for a sample sequence that might not so often happening in reality but can give us insights in the limitation of the algorithm. The ordering_2 is shown in table 3 and is a randomly created sequence.

As a second data set we chose the well-known IRIS data set. This data set is larger than the synthetic data set. It has 150 samples and 3 classes. Each class has 50 samples and four features. It is well known that the class Setosa is separated well from the classes Versicolor and Viriginica while the later two classes overlap.

5 Results

5.1 Cluster Structure and Separation

The performance of the clustering algorithm was tested under several conditions. Figure 9 shows the hierarchy of clusters with the threshold $u_{kt} \geq 0.2$ and before the input of instance 3-9. The tree becomes too bushy. It has four clusters under the root node. The analytical analysis of these observations showed that under certain conditions the merge-operator does not work optimally. The algorithm suggests this operation too often.

The final cluster hierarchy obtained from two different orderings of the samples and after introduction of a threshold for the membership value into the algorithm is shown in Figures 10 and 11 for a membership value $u_{kt} \geq 0.2$ and is shown for a membership value of $u_{kt} \geq 0.1$ in Figures 12 and 13. The tree in Figure 10 shows three clusters after the root node and the tree in Figure 13 two clusters after the root node. Four clusters after the root node are shown in Figure 12 and two clusters after the root node are shown in Figure 13. The class distribution in the two dimensional solution-space for the different variations of the method are shown in Figure 14 to Figure 17. The results show that the value of the threshold for the fuzzy membership has a great influence on the final result, as expected. Besides that, the order of the samples has an influence on the result. This means that the reorganization of the hierarchy according to the operators described in Section 3.2 is not optimal yet.

The result shown in Figure 14 represents the distribution of the classes very well except for the two green samples next to cluster_1 and cluster_2. These samples are wrongly classified. Because of the low number of samples left in the data stream, the algorithm is no longer able to merge these two samples to any of cluster_1 or cluster_2. This is a general problem of an incremental algorithm.

Next, we tested our algorithm on the well-known IRIS data set. The result for $u^{*} = 0.2$ is shown in table 4. Compared to our synthetic data set, the IRIS data set has many more samples (see Section 4). We obtain four clusters after the root node.

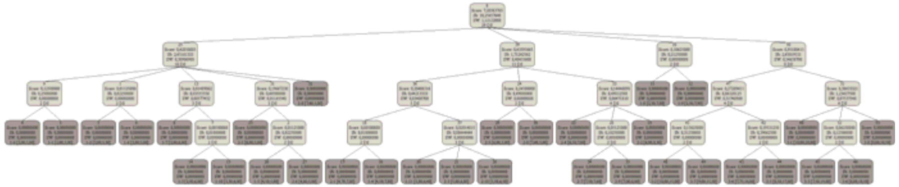


Fig. 9. Hierarchy of Clusters before input Instance 3-9

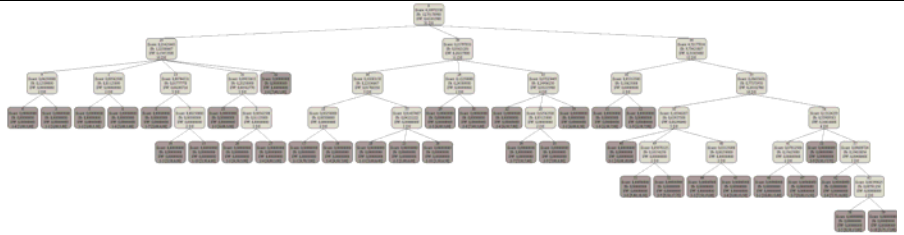


Fig. 10. Hierarchy of Clusters after data sample input order_1 and $u^* = 0.2$

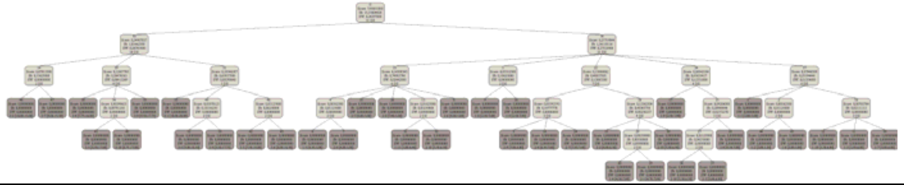


Fig. 11. Hierarchy of Clusters after data sample input order_2 and $u^* = 0.2$

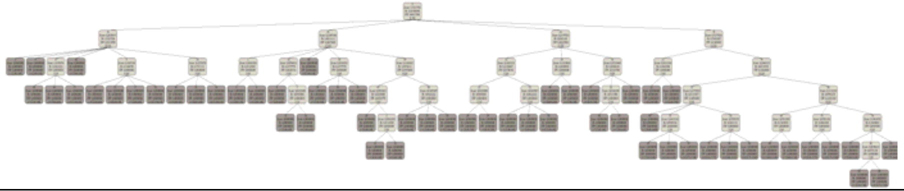


Fig. 12. Hierarchy of Clusters after data sample input order_1 and $u^* = 0.1$

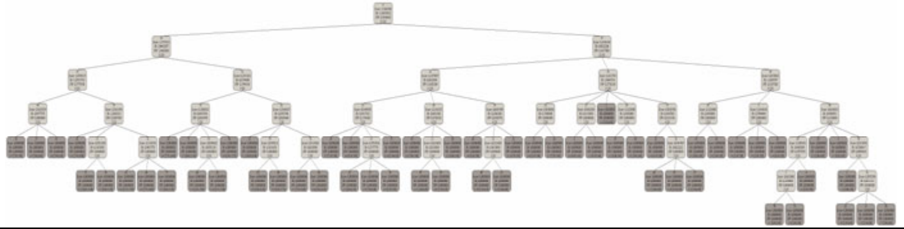


Fig. 13. Hierarchy of Clusters after data sample input order_2 and $u^* = 0.1$

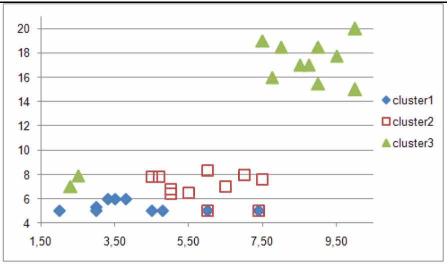


Fig. 14. Class distribution after data sample input order_1 and $u^* = 0.2$

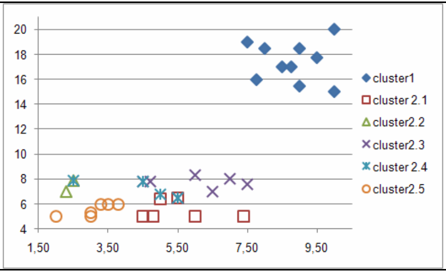


Fig. 15. Class distribution after data sample input order_2 and $u^* = 0.2$

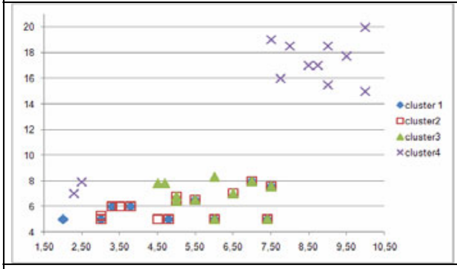


Fig. 16. Class distribution after data sample input order_1 and $u^* = 0.1$

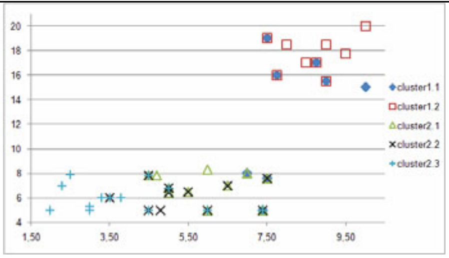


Fig. 17. Class distribution after data sample input order_2 and $u^* = 0.1$

Table 4. Results for the IRIS Data Set with $u^* = 0.2$

class	Cluster_1 (node 97)	Cluster_2 (node 101)	Cluster_3 (node 90)	Cluster_4 (node 548)	Cluster_4.1 (node 470)	Cluster_4.2 (node 232)	Cluster_4.3 (node 526)	Cluster_4.4 (node 549)
n_k	33	20	24	100	38	9	45	67
Setosa	31	20	21	0	0	0	0	0
Versicolor	2	0	3	50	2	8	8	42
Virginica	0	0	0	50	36	1	37	25

The cluster_4 has four sub-clusters. Cluster_1 and Cluster_2 represent mainly the class Setosa. Cluster_3 represents the class Setosa with 21 samples and the class Versicolor with 3 samples. This cluster should have been merged by the algorithm with Cluster_1, Cluster_2, and Cluster_3; in this case, the class Setosa would have been well represented. Cluster_4 represents Versicolor and Virginica with 50 samples each. Cluster_4.1 is based on 36 samples of the class Virginica and two samples of the class Versicolor. Cluster_4.3 represents 37 samples of the class Virginica and 8 samples of the class Versicolor. It would have been better if these two sub-clusters had been merged to one cluster but unfortunately the samples that come in later do not force the algorithm to choose this operation.

Cluster_4.2 is based on 8 samples of the class Versicolor and one sample of the class Virginica. The cluster contains the lowest number of samples.

Cluster_4.4 mostly represents the class Versicolor (42 samples) and the class Virginica with 25 samples.

Next, we have established the cluster-hierarchy in table 4. We studied the influence of the samples. We inserted the following data again at the end of the data list: Versicolor8, Versicolor11, Versicolor13, Versicolor30, Versicolor31, Versicolor32, Versicolor44, Versicolor49, and Virginica7. The number of samples is no longer 150, it is now 159. This means that we have some repetition in the data set. The result is shown in table 5. We obtain two main clusters in which the class Setosa is well represented in cluster_1. Cluster_2 represents the class Versicolor and Virginica and confirms the observation of table 4. Cluster_2.1 and cluster_2.2 represents mainly the class Virginica while the class Versicolor is represented in cluster_2.3.

Table 5. Test with the IRIS data set and with $u^* = 0.2$

class	Cluster_1 (node 592)	Cluster_2 (node 603)	Cluster_2.1 (node 462)	Cluster_2.2 (node 518)	Cluster_2.3 (node 604)
n_k	60	107	38	46	77
Setosa	50	0	0	0	0
Versicolor	10	6	2	9	45
Virginica	0	51	36	37	22

5.2 Prototype Calculation

The influence of the kind of prototype calculation has been studied by calculating the prototype and the fuzzy prototype on each level of the established hierarchy of the tree. The fuzzy prototype was calculated according to the formula 2 on the data samples of each hierarchy level. The relative mean squared error between the crisp prototype and the fuzzy prototype and the recalculated fuzzy prototype and the fuzzy prototype after updating the hierarchy is shown in table 6.

Table 6 shows that it is better to update the prototype after updating the hierarchy.

5.3 Prediction with the Established Classes of the Clustering Algorithm Compared to the Expert´s Classification

Finally, based on the different labeled data sets obtained from the clustering algorithms, we learnt a decision tree classifier based on our tool *Decision Master* (www.ibai-solutions.de) and calculated the accuracy of the pruned and unpruned decision tree classifier by cross validation. The result for the pruned decision tree is shown in Figure 18. We achieve the highest accuracy of 87.10% for the expert labeled data set. For the crisp standard fuzzy c-mean algorithm we achieve 74.19% and for

Table 6. Relative mean squared error between prototype and fuzzy prototype and updated prototype and fuzzy prototype

Node	$error_k(m_k^1, m_k^2)$	$error_k(m_k^1, m_k^3)$	$error_k(m_k^2, m_k^3)$
1	0,02984876	0,21629228	0,2452405
2	0,01220975	0,01220975	0
7	0,09943115	0,09943115	0
8	0,05337351	0,05337351	0
9	9,01024581	9,9054681	0,84087822
10	0,6281344	0,6281344	0
13	0,17688003	0,14734399	0,05814645
18	0,04961404	0,04961404	0
19	0,08370257	0,08370257	0
22	0,10734538	0,10734538	0
23	0,00554033	0,00554033	0
24	0,65608652	0,65608652	0
25	8,80472526	12,8536812	4,01109733
28	0,16614215	0,16614215	0
31	1,81287314	3,24509496	1,41410651
32	0,9467677	1,46209474	0,51190012
33	0,63681832	0,63681832	0
36	0,28670425	0,28670425	0
37	0,06902223	0,06902223	0
39	2,73639036	1,63861976	1,42009094
40	0,04244045	0,04244045	0
41	0,15389293	0,15389293	0
42	0,00123248	0,00123248	0
43	0,00149571	0,00149571	0
48	0,03889242	0,03889242	0
49	0,1823191	0,1823191	0
50	0,3848048	0,44396147	0,08170046
53	0,37145551	0,37145551	0
54	10,0228777	10,7817879	0,92098506
55	0,12090512	0,12090512	0
56	1,02624322	1,34545759	0,3540054
57	0,24419508	0,24419508	0

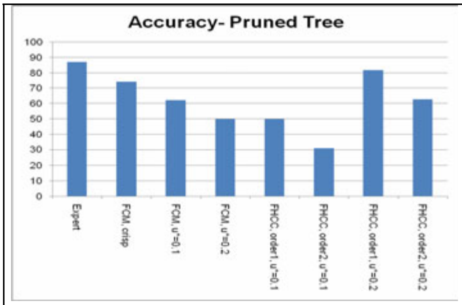


Fig. 18. Accuracy of the pruned Decision Tree Classifier based on the data set from different clusterings

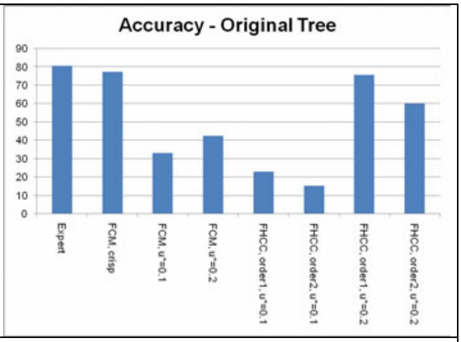


Fig. 19. Accuracy of the Decision Tree Classifier based on the data set from different clusterings

the fuzzy conceptual clustering algorithm we achieve 81.82% accuracy in case of $u^* = 0.2$, ordering_1 of the data.

The lowest accuracy we obtain for the standard fuzzy c-mean and the fuzzy conceptual clustering algorithm with $u^* = 0.1$.

In summarizing the above, the fuzzy conceptual clustering algorithm with $u^* = 0.2$ achieves, after the classifier derived from the expert data, the best accuracy, followed by the crisp fuzzy c-mean algorithm.

Figure 19 shows the unpruned decision tree illustrating different circumstances.

The best result is achieved for the crisp fuzzy c-mean followed by the fuzzy conceptual clustering algorithm with $u^* = 0.2$.

When pruning the tree we see the opposite for the accuracy for the fuzzy c-mean. This is not the case for the fuzzy conceptual clustering. The results are more stable and not so noisy as in case of the fuzzy c-mean.

6 Discussion

We have tested our new fuzzy conceptual clustering algorithm on two different data samples: the standard IRIS data set and a synthetic data set. Our synthetic example has a low number of samples in each class. This fact might have a great influence on the result but the chosen example allowed us to visualize the result.

In general, we can say that our new clustering algorithm works very well and can incrementally produce a cluster hierarchy that needs no human interaction. The threshold for the membership value is necessary to come up with clusters that represent the natural uncertainty of the data. Otherwise the algorithm produces too much overlapping clusters. It seems that the value of 0.2 seems to be the best value.

The chosen calculation for the fuzzy prototype works well and comes close to the true value.

The chosen operations to re-optimize the hierarchy work well but need to have some improvement for the merge-operation. This will be studied in future work.

The evaluation of the decision tree classifier shows that the accuracy of the decision tree is very good for the fuzzy conceptual clustering algorithm and the accuracy comes close to the expert´s accuracy.

7 Conclusions

We have developed a new fuzzy conceptual clustering algorithm. We studied the behavior of the algorithm on a synthetic data set and the IRIS data set.

The algorithm can cluster samples in temporal sequence in the way the samples appear in most of the nova-day applications. It is an incremental clustering algorithm. When to perform which operation is decided based on a fuzzy score. This fuzzy score averages the separation between the clusters and the compactness.

The fuzzy conceptual clustering algorithm can well establish fuzzy concepts.

References

1. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. Journal of Cybernetics 3, 32–57 (1973)
2. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)

3. Gustafson, E.E., Kessel, W.C.: Fuzzy Clustering with a Fuzzy Covariance Matrix. In: IEEE CDC, San Diego, Californian, pp. 761–766. IEEE-Press, Los Alamitos (1979)
4. Pedrycz, W.: Knowledge-Based Clustering. John Wiley & Sons, Inc., Chichester (2005)
5. Yang, M.S.: A Survey of Fuzzy Clustering. Mathl. Comput. Modelling 18(11), 1–16 (2003)
6. Fadili, M.J., Ruan, S., Bloyet, D., Mayozer, B.: On the number of clusters and the fuzziness index for unsupervised FCA application to BOLD fMRI times series. Medical Image Analysis 5, 55–67 (2001)
7. Mendes Rodrigues, M.E.S., Sacks, L.: A Scalable Hierarchical Fuzzy Clustering Algorithm for Text Mining. In: Proc. of the 4th International Conference on Recent Advances in Soft Computing, RASC 2004, Nottingham, UK, pp. 269–274 (2004)
8. Torra, V.: Fuzzy c-means for fuzzy hierarchical clustering. In: Fuzz-IEEE 2005, Reno, Nevada, May 22-25, pp. 646–651. IEEE-Press, Los Alamitos (2005)
9. Rodrigues, P.P., Gama, J.: A Semi-Fuzzy Approach for Online Divisive-Agglomerative Clustering. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) EPIA 2007. LNCS (LNAI), vol. 4874, pp. 133–144. Springer, Heidelberg (2007)
10. Bordogna, G., Pagani, M., Pasi, G.: An Incremental Hierarchical Fuzzy Clustering for Category-Based News Filtering. In: Bouchon-Meunier, B., Marsala, C., Rifqi, M., Yager, R.R. (eds.) Uncertainty and Intelligent Information Systems, pp. 143–154. World Scientific Publishing Company, Singopore (2008)
11. Quan, T.T., Hui, S.C., Cao, T.H.: A Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data. In: Snášel, V., Bělohlávek, R. (eds.) CLA 2004, pp. 1–12, Technical University of Ostrava, Dept. of Computer Science (2004) (ISBN 80-248-0597)
12. Fisher, D.H.: Knowledge acquisition via incremental conceptual clustering. Machine Learning 2(2), 139–172 (1987)
13. Lebowitz, M.: Experiments with incremental concept formation: UNIMEM. Machine Learning 2(2), 103–138 (1987)
14. Gennaria, J.H., Langleya, P., Fisher, D.: Models of incremental concept formation. Artificial Intelligence 40(1-3), 11–61 (1989)

Mining Concept Similarities for Heterogeneous Ontologies

Konstantin Todorov[1], Peter Geibel[2], and Kai-Uwe Kühnberger[3]

[1] Laboratory MAS, École Centrale Paris, F-92 295 Châtenay-Malabry, France
[2] TU Berlin, Fakultät IV, Franklinstr. 28/29, 10587 Berlin, Germany
[3] Institute of Cognitive Science, Universtity of Osnabrück, Germany

Abstract. We consider the problem of discovering pairs of similar concepts, which are part of two given source ontologies, in which each concept node is mapped to a set of instances. The similarity measures we propose are based on learning a classifier for each concept that allows to discriminate the respective concept from the remaining concepts in the same ontology. We present two new measures that are compared experimentally: (1) one based on comparing the sets of support vectors from the learned SVMs and (2) one which considers the list of discriminating variables for each concept. These lists are determined using a novel variable selection approach for the SVM. We compare the performance of the two suggested techniques with two standard approaches (Jaccard similarity and class-means distance). We also present a novel recursive matching algorithm based on concept similarities.

1 Introduction

In A[rtificial] I[ntelligence], an ontology, in the broadest sense, is understood as a collection of *concepts* and *relations* defined on these concepts, which altogether describe and structure the knowledge in a certain domain of interest. The *O[ntology] M[atching]* problem stems from the fact that different communities, independently from one another, are likely to adopt different ontologies, given a certain domain of interest. In consequence, multiple *heterogeneous* ontologies, describing similar or overlapping fractions of the world are created. An ontology matching procedure aims at reducing this heterogeneity by yielding assertions on the relatedness of cross-ontology concepts, in an automatic or semi-automatic manner. To these ends, according to [5], a matching procedure commonly relies on extensional (related to the concepts instances), structural (related to the inter-ontology concepts relations), terminological (language-related) or semantic (related to logical interpretation) information, separately or in combination.

In this paper, we will expand on one of these general types of ontology matching, known as *instance-based* matching. This comprises a set of approaches for measuring the similarity of concepts from two source ontologies based on their extensions – the instances that populate the respective concepts [9]. We consider two training sets with classified examples, one for each of the two source ontologies. We assume that the examples in each training set are described using the

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 86–100, 2010.

same set of variables or attributes. The classes of the examples, however, belong to two separate conceptual systems that serve as taxonomies and possess a hierarchical, tree-like structure. The ontology nodes are thus implicitly assigned the relevant examples from the training set, taking into account that the hierarchical structure represents an *is_a*-relationship between concept nodes.

Being given these two training sets together with the source ontologies, we now consider the *data mining task* of discovering similarities between concepts of the source ontologies based on the instances and the structures of the ontologies. In the case that the similarities imply a suitable one-to-one mapping between concept nodes of the two source ontologies, they can be used for computing the "intersection" of the two source ontologies which forms the result of the matching process.

In contrast to other learning-based matching approaches, which aim at estimating joint probabilities for concept pairs [4], we compute the similarity of two concepts based on the similarity of their intra-ontology classifiers. In the case of the S[upport] V[ector] M[achines] [3], this can be achieved by comparing the support vectors characterizing the respective concepts. The support vectors are examples for the respective concept that can be considered important for discriminating it from other classes in the same ontology, and are thus relevant for characterizing it with respect to the whole ontology.

We present a second generic approach, which bases concept similarity on variable selection techniques that capture characteristics of the data in terms of the relevance of variables for classifier learning. In the case of text documents, these approaches allow to determine those words or terms that discriminate the respective concept from other concepts in the same ontology. We introduce a new technique for selecting variables for SVMs and use it for determining the similarity of two concepts by comparing their lists of discriminative attributes. The two novel similarity measures are tested against two standard techniques used in state-of-the-art approaches: the Jaccard similarity and the class-means distance.

The remainder of the paper has the following structure. Section 2 presents relevant related approaches to ontology matching. Section 3 sets the ontology matching framework in terms of definitions and assumptions. The two new approaches to measure concept similarity are suggested further in Section 4 (comparison of support vectors) and Section 5 (variable selection). A variable selection criterion for SVM, together with a short introduction to the classifiers is presented in Section 6. Section 7 descriebs a recursive matching procedure based on the proposed similarity measures. Finally, we present our experimental results in Section 8 before we conclude with Section 9.

2 Related Work

As mentioned in the introduction, an important part of the existing OM-approaches, including ours, are characterized as *extensional*, i.e. grounded in the external world, relying on instances in order to judge intensional similarity. Among the basic assumptions of such approaches is that two ontologies use the same instances to populate different conceptual structures and when this is not so, mechanisms for extracting instances (from text corpora or other external sources) should be made available (FCA-merge [15]). Other techniques rely on estimating the concepts similarity by measuring class-means distances (Caiman [11]) or estimating joint probabilities by the help of machine learning techniques (Glue [4]). Most of these standard approaches are based on rather restrictive assumptions, tend to be costly on a large scale or perform well for leaf-nodes but fail to capture similarities on higher levels.

It is a relatively old idea that the gap between two conceptual systems can be bridged by using the relations of the concepts within each of the systems. The use of structure for judging concept similarities has found response in the OM community, some examples of such algorithms being Anchor-Prompt [13], Absurdist [6] and Onion [12].

Our work is much in line with the tradition of extensional concept representation and similarity measurement. The relations between concepts are used in order to improve and optimize the extensional similarity judgments. They are taken into account by the recursive matching algorithm that is based on pairwise concept similarities. In technical terms, an advantage of our method is that most of it is accomplished with the training phase of the classification task. In contrast to most instance-based techniques, our matching approach does not rely on intersections of instance sets, nor on the estimation of joint probabilities. It works with instance sets that might be different for both ontologies, which avoids taking the costly step of extracting instances from external sources. Finally, in case of textual instances, the method makes available the list of the most important words that characterize a similar pair of concepts - information not readily available in the approaches cited above.

3 Populated Ontologies: Definition and Assumptions

Throughout this paper, an ontology, whose concepts are labels of real-world instances of some kind, will be defined in the following manner (modifying a definition found in [15]).

Definition 1. *A* ***populated ontology*** *is a tuple* $O = \{C, \texttt{is\_a}, R, I, g\}$, *where* C *is a set whose elements are called concepts,* `is_a` *is a partial order on* C, R *is a set of other (binary) relations holding between the concepts from the set* C, I *is a set whose elements are called instances and* $g : C \rightarrow 2^I$ *is an injection from the set of concepts to the set of subsets of* I.

In the formulation above, a concept is *intensionally* defined by its relations to other concepts via the partial order and the set R, and *extensionally* by a set of instances via the mapping g. We note that the sets C and I, are compulsorily non-empty, whereas R can be the empty set. In view of this remark, the definition above describes a *hierarchical ontology*: an ontology which, although not limited to subsumptional relations, necessarily contains a hierarchical backbone, defined by the partial order on the set of concepts. If non-empty, R contains relations, defined by the ontology engineer (for instance, `graduated_at`, `employed_by`, etc.). The set I is a set of concept instances – text documents, images or other (real world data) entities, representable in the form of real-valued vectors. The injection g associates a set of instances to every concept. By definition, the empty set can be associated to a concept as well, hence not every concept is expected or required to have instances. Whether g takes inheritance via subsumption into account in defining a concept's instance-set (hierarchical concept instantiation) or not (non-hierarchical instantiation) is a semantics and design-related issue [9].

Let us consider two ontologies O_1 and O_2 and their corresponding instance-sets $I_1 = \{\mathbf{i}_1^1, ..., \mathbf{i}_{m_1}^1\}$ and $I_2 = \{\mathbf{i}_1^2, ..., \mathbf{i}_{m_2}^2\}$, where each instance is represented as an n-dimensional vector and m_1 and m_2 are integers. For a concept $A \in C_1$ from ontology O_1, we define a labeling $S^A = \{(\mathbf{i}_j^1, y_j^A)\}$, where y_j^A takes a value $+1$ when the corresponding instance $\mathbf{i}_j^1$ is assigned to A, and -1 otherwise, for $j = 1, ..., m_1$. The labels split the instances of O_1 into those that belong to the concept A (positive instances), and those that do not (negative ones) defining a binary classification training set. The same representation can be acquired analogously for any concept in both ontologies O_1 and O_2.

4 Concept Similarity via Comparison of Intra-ontlogy Classifiers

A straightforward idea for determining the similarity $sim(A, B)$ of two concepts A and B consists in comparing their instance sets $g(A)$ and $g(B)$. For doing so, we thus need a similarity measure for instances $\mathbf{i}^A$ and $\mathbf{i}^B$. We can use, for instance, the scalar product and the cosine $s(\mathbf{i}^A, \mathbf{i}^B) = \frac{\langle \mathbf{i}^A, \mathbf{i}^B \rangle}{\|\mathbf{i}^A\| \|\mathbf{i}^B\|}$ (i.e. the normalized scalar product). Based on this similarity measure for elements, the similarity measure for the sets can be defined by computing the similarity of the mean vectors corresponding to class prototypes, i.e.

$$sim_{proto}(A, B) = s\Big(\frac{1}{|g(A)|} \sum_{j=1}^{|g(A)|} \mathbf{i}_j^A, \frac{1}{|g(B)|} \sum_{k=1}^{|g(B)|} \mathbf{i}_k^B\Big) . \tag{1}$$

This method underlies the Caiman approach [11] in which concepts are assumed to be represented by their mean vector. While this approach might be suitable for leaf nodes, in which the data might be characterized by a unimodal distribution, it is generally bound to fail for nodes higher up in the tree, whose instance set might be composed of several subsets resulting in a multi-modal distribution.

The theory of hierarchical clustering (e.g., [1]) provides alternative methods for defining similarities for pairs of sets. Examples are the similarity measures

$$sim_{min}(A, B) = \min_{j,k} s(\mathbf{i}_j^A, \mathbf{i}_k^B), \quad sim_{max}(A, B) = \max_{j,k} s(\mathbf{i}_j^A, \mathbf{i}_k^B) \tag{2}$$

and

$$sim_{avg}(A, B) = \frac{1}{|g(A)||g(B)|} \sum_{j,k} s(\mathbf{i}_j^A, \mathbf{i}_k^B) . \tag{3}$$

The three measures correspond to different types of clustering methods: complete link, single link, and average link clustering.

It is well-known that computing the similarity based on these measure can be quite complex if the training sets are large. Moreover, the min- and max-based measures can get easily spoiled by outliers in the training sets, whereas the average link approach is also prone to the problem of multi-modal distributions.

Ontologies are based on the idea of discriminating between different concepts or classes in order to conceptualize a domain. This idea also forms the basis of the SVM described in more detail in Section 6. After successfully learning a classifier, the support vectors of class A correspond to those examples in $g(A)$ that have turned out to be most important for discriminating it from the other class containing the instances for the concepts in $C_1 \setminus \{A\}$. This means in particular that the support vectors for the A-classifier are elements of $g(A)$, whereas those for B can be found in $g(B)$. The idea which we propose is to base the measures in (2) and (3) only on the support vectors. If we can train the classifiers successfully (i.e., with a low error), this will reduce complexity and can help solve the problem of outliers and irrelevant examples.

5 Concept Similarity via Variable Selection

V[ariable] S[election] techniques (reviewed in [7]) serve to rank the input variables of a given problem (e.g. classification) by their importance for the output (the class affiliation of an instance), according to certain evaluation criteria. Technically speaking, a VS procedure assigns to each variable a real value – *a score* – which indicates the variable's pertinence. This can be of help for dimensionality reduction and for extracting important input-output dependencies. Assuming that instances are represented as real-valued vectors, a VS procedure in our study indicates which of the vector dimensions are most important for the separation of the instances (within a single ontology) into those that belong to a given concept and those that do not. In the case of documents, these might be words or tokens that distinguish the respective concept from others in the same ontology.

By the help of variable selection procedures carried out independently for two concepts $A \in C_1$ and $B \in C_2$, on their corresponding sets $S^A = \{(\mathbf{i}^1_j, y^A_j)\}$, $j = 1, ..., m_1$ and $S^B = \{(\mathbf{i}^2_k, y^B_k)\}$, $k = 1, ..., m_2$, one scores the input variables by their importance for the respective class separations. In that, the concepts A and B can be represented by the lists of their corresponding variables scores in the following manner:

$$Scores(A) = (s^A_1, s^A_2, ..., s^A_n),\ Scores(B) = (s^B_1, s^B_2, ..., s^B_n), \tag{4}$$

Note that to score the input variables, one could rely on various selection techniques. In previous studies, we have tested structural dimension reducing methods (discriminant analysis), standard feature selection techniques for text categorization (point-wise mutual information, chi-square statistics and document frequency thresholding), as well as an SVM-based method [16]. The latter falls in the focus of the current paper and will be, therefore, introduced in more detail in the following section.

On the basis of the concept representations in (4), different measures of concept similarity can be computed [16]. The k-TF measure looks for re-occurring elements in two lists of top $k-$scored variables. Alternatively, parameter-free measures of statistical correlation, which act as measures of similarity, can be computed over the ranks or directly over the scores associated to the variables. Pearson's coefficient, which has been used in the experimental part of this paper, is given by

$$r = \frac{\sum_{i=1}^{n}(s_i^A - s_{mean}^A)(s_i^B - s_{mean}^B)}{\sqrt{\sum_{i=1}^{n}(s_i^A - s_{mean}^A)^2}\sqrt{\sum_{i=1}^{n}(s_i^B - s_{mean}^B)^2}}, \tag{5}$$

where s_{mean}^A and s_{mean}^B are the means of the two respective score lists over all n variables.

6 A Variable Selection Method for the SVM

In the following, Section 6.1 aims at familiarizing the reader with several concepts from the SVM theory that are relevant for the introduction of our SVM-based variable selection criterion described, in turn, in Section 6.2.

6.1 Support Vector Machines

The SVMs are inductive machine learners, initially designed to solve binary classification tasks [3]. For reasons of space, we will provide knowledge about the method limited to what is sufficient to understand the ideas behind SVM-based variable selection (comprising existing methods and our approach).

Let us consider the following binary classification layout. Assume we have l observations $\mathbf{x}_i \in \mathbb{R}^n$ and their associated "truth" $y_i \in \{-1, 1\}$. Data are assumed to be i.i.d. (independent and identically distributed), drawn from an unknown probability distribution $P(\mathbf{x}, y)$. The goal of binary classification is to "learn" the mapping $\mathbf{x}_i \mapsto y_i$ which is consistent with the given examples. Let $\{f(\mathbf{x}, \sigma)\}$ be a set of such possible mappings, where σ denotes a set of parameters. Such a mapping is called a classifier and it is deterministic - for a certain choice of $\mathbf{x}$ and σ it will always give the same output f.

The **actual risk**, or the expectation of the test error for such a learning machine is

$$R(\sigma) = \int \frac{1}{2}|y - f(\mathbf{x}, \sigma)|dP(\mathbf{x}, y).$$

The quantity $1/2|y - f(\mathbf{x}, \sigma)|$ is called *the loss*. Based on a finite number, l, of observations, we calculate the **empirical risk**

$$R_{emp}(\sigma) = \frac{1}{2l}\sum_{i=1}^{l}|y_i - f(\mathbf{x}_i, \sigma)|,$$

which is a fixed number for a given training set $\{(\mathbf{x}_i, y_i)\}$ and a certain choice of parameters σ.

For losses taking values 0 or 1, with probability $1-\eta$, $0 \leq \eta \leq 1$, the two risks are related in the following manner:

$$R(\sigma) \leq R_{emp}(\sigma) + \sqrt{\frac{h\log(\frac{2l}{h}) + 1 - \log(\frac{\eta}{4})}{l}}, \tag{6}$$

where h is a nonnegative integer which will play a core role in our variable selection procedure, called the *VC dimension*. The bound (6) gives an insight in one very important aspect of generalization theory of statistical learning. The term $\sqrt{\frac{hlog(\frac{2l}{h})+1-log(\frac{\eta}{4})}{l}}$, called *VC confidence* is "responsible" for the *capacity* of the learner, i.e. its ability to classify unseen data without error. The other right-hand quantity in (6) - the empirical risk, measures the *accuracy* attained on the particular training set $\{(\mathbf{x}_i, y_i)\}$. What is sought for is a function which minimizes the bound on the actual risk and thus provides a good balance between capacity and accuracy - a problem known in the literature as *capacity control*.

The presented risk bound does not depend on $P(\mathbf{x}, y)$ and it can be easily computed provided the knowledge of h. We introduce what does this parameter stand for. Let us consider the set of functions $\{f(\mathbf{x}, \sigma)\}$ with $f(\mathbf{x}, \sigma) \in \{-1, 1\}, \forall \mathbf{x}, \sigma$. In a binary classification task there are 2^l possible ways of labeling a set of l points. If for each labeling there can be found a member of $\{f(\sigma)\}$ which correctly assigns these labels, we say that the given set of points is *shattered* by the given set of functions. The VC dimension is a property of such a family of functions, which is defined as the maximum number of training points that can be shattered by that family. Although in general difficult to compute directly, an upper bound for the VC-dimension can be computed depending on the weight vector $\mathbf{w}$ and on properties of the data. In the SVM*light* implementation, which we have used for our experiments in Section 8.4, the VC dimension is estimated based on the radius of the support vectors [10].

Now, let us return to binary classification. Consider the input space $X \subseteq \mathbb{R}^n$ and the output domain $Y = \{-1, 1\}$ with a training set $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_l, y_l)\} \in (X, Y)^l$. SVM is a linear real function $f : X \to \mathbb{R}$ with

$$f(\mathbf{x}) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b,$$

where $\sigma = (\mathbf{w}, b) \in \mathbb{R}^n \times \mathbb{R}$. The separating hyperplane in the input space X is defined by the set $\{\mathbf{x} | f(\mathbf{x}) = 0\}$. The decision rule assigns an input vector $\mathbf{x}$ positive if and only if $f(\mathbf{x}) \geq 0$ and negative - otherwise. (The inclusion of 0 in the first case and not in the second is conventional.)

We are looking for the best decision function $f(\mathbf{x})$ which separates the input space and maximizes the distance between the positive and negative examples closest to the hyperplane. The parameters of the desired function are found by solving the following quadratic optimization problem:

$$\min_{\mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}} \frac{1}{2} \|\mathbf{w}\|^2$$

under the linear constraints

$$\forall i = 1, ..., l,\ y_i(\langle \mathbf{w}, \mathbf{x}_i)\rangle + b) \geq 1.$$

When data are not linearly separable in the input space, they are mapped into a (possibly higher dimensional) space, called *feature space* where a linear boundary between both classes can be found. The mapping is done implicitly by the help of a kernel function which acts as a dot product in the feature space.

When solving the above optimization problem, the weight vector $\mathbf{w}$ can be expressed as a linear combination of the input vectors. It turns out that only certain examples have a weight different from 0. These vectors are called *support vectors* for they define the separating hyperplane and can be considered the examples closest to it.

6.2 VC-Dimension-Based Variable Selection

SVM-based variable selection has already been studied in the past couple of years. Guyon *et al.* proposed the SVM-RFE algorithm [8] for selecting genes which are relevant for cancer classification. The removal criterion for a given variable is minimizing the variation of the weight vector $\| \mathbf{w} \|^2$, i.e. its sensitivity with respect to a variable. Rakotomamonjy *et al.* carried out experiments for pedestrian recognition by the help of a variable selection procedure for SVMs based on the sensitivity of the margin according to a variable [14]. A method based on finding the variables which minimize bounds on the *leave-one-out* error for classification was introduced by Weston *et al.* [17]. Bi *et al.* developed the VS-SSVM variable selection method for regression tasks applied to molecules bio-activity prediction problems [2].

The variable selection criterion that we propose is based on the sensitivity of the VC dimension of the SVM classifiers with respect to a single variable or a block of variables. As we have seen in the previous subsection, for different values of the VC dimension h, different values of the VC confidence (describing the capacity of the classifier) will be computed and thus different bounds on the actual risk (6), where from the generalization power of the classifier will change. Our main heuristics can be formulated as *"a less informative variable is one, which the VC confidence of the classifier is less sensitive to"*.

For computational reasons the evaluation function of our variable selection procedure will be formulated in terms of VC dimension directly, instead of in terms of the VC confidence. This is plausible since the VC confidence is monotonous in h. Thus, the i-th variable is evaluated by

$$eval_i = h(H) - h(H^{(i)}),\ i = 1, ..., n, \tag{7}$$

where $h(H)$ is the VC dimension of a set of SVM hypotheses H constructed over the entire data set and $h(H^{(i)})$ is the same quantity computed after the removal of the i-th variable (whose pertinence is to be evaluated) from the data set.

Tests of the performance of suggested variable evaluation criterion are presented in [16].

7 A Similarity-Based Matching Procedure

Fig. 1 shows two ontologies that intend to organize news articles in different structurally related topics. This example will help us introduce (in the current section) and evaluate (in the following section) a recursive ontology matching procedure. We abstract from the concept names being similar, for they are not taken into consideration in the concept similarity measurement. We have populated the concepts of the ontologies with documents taken from the 20 Newsgroups[1] dataset in a way that any two document sets across both ontologies are largely non intersecting.

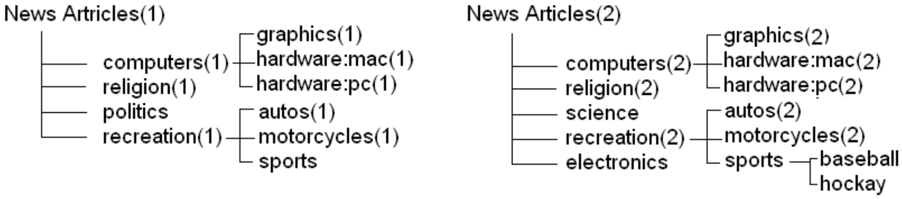


Fig. 1. Two news ontologies

One could construct an ontology matching procedure based on a concept similarity measure by producing a set of N 1-to-M similarity assertions for an ontology with N and an ontology with M concepts. However, it is likely that such an initiative turns out to be rather costly, in spite of it being semantically unjustified, for in this case *structure* is not taken into account.

We suggest that the concept similarity measure should be applied recursively on the sets of concepts found on *corresponding* levels of the two ontologies, descending down the hierarchies. In a properly designed ontology, the classes on a single level (also referred to as *unilevel* classes) are internally homogeneous and externally heterogeneous. The search of potentially similar concepts is optimized by taking the concepts intra-ontology relations into account. The proposed recursive procedure stems from a simple rule: concept similarity is tested only for those cross-ontology concepts, whose parents have already been judged similar.

Considering the ontologies in Fig. 1, we start by mapping the set of concepts $\{$*Computers(1), Religion(1), Politics(1), Recreation(1)*$\}$ against the set $\{$*Computers(2), Religion(2), Science(2), Recreation(2), Electronics(2)*$\}$. Let the mappings identified by the help of the similarity measure be $\{$*Computers(1)* $\rightarrow$ *Computers(2)*$\}$, $\{$*Religion(1)* $\rightarrow$ *Religion(2)*$\}$, and $\{$*Recreation(1)* $\rightarrow$ *Recreation(2)*$\}$. We proceed to map the sets of the children of each pair of concepts mapped in the first step (the children of the two *Computer*-classes and the children of the two *Recreation*-classes). The procedure stops when reaching the leaves of the trees.

If our source ontologies are more complex than the ones considered above and are of different granularities, we have to ensure that at each step we are

[1] http://people.csail.mit.edu/jrennie/20Newsgroups/

matching corresponding levels. To these ends, we suggest to set a threshold of the measured concept similarity: if the values found are under that threshold, the levels do not correspond and we should descend on the following level of O_2. A pseudo-code of the procedure is given in Algorithm 1, which also allows skipping of levels for the first ontology and additionally employs an anchoring technique. The procedure can be adapted for ontologies that contain other than strictly hierarchical relations in addition to a well-defined hierarchical backbone (see definition 1 and the comments thereafter) by matching the hierarchical backbones prior to the remaining concepts.

```
procedure Map($Set_1, O_1, Set_2, O_2$)
// Returns a set of mappings for the concepts of $Set_1$ and $Set_2$,
// and for their descendants in $O_1$ and $O_2$
begin
    if $Set_1 = \emptyset$ or $Set_2 = \emptyset$ then return $\emptyset$
    $\Sigma = \{\}$ // Initialize the set of mappings
    for $A \in Set_1$ do // Find matches
        for $B \in Set_2$ do
            if $sim(A, B) \geq threshold$ then
                $\Sigma := \Sigma + \{(A, B)\}$
                break// ... to enforce injectivity of mapping
    if $\Sigma \neq \emptyset$ then // Add mappings for descendants
        $Mappings = \Sigma$
        for $(A, B) \in \Sigma$ do // Recursions
            $Mappings = Mappings \cup$ Map($children[A], children[B]$)
        return $Mappings$
    // Sigma is empty: Skip one level or more in $O_2$
    $\Sigma_1$=Map($Set_1, children[Set_2]$)
    if $\Sigma_1 \neq \emptyset$ then  return $\Sigma_1$
    // None of the concepts in $Set_1$ could be mapped: skip this level of $O_1$
    return $Map(children[Set_1], Set_2)$
end

procedure Main($O_1$, $O_2$)
begin
    // Find potential anchors and compute mappings for their descendants:
    for $A$ in $O_1$ using breadth-first-search do
        for $B$ in $O_2$ using breadth-first-search do
            if $sim(A, B) \geq threshold$ then
                return $\{(A, B)\} \cup$ Map(children[$A$], $O_1$, children[$B$], $O_2$)
    return $\emptyset$
end
```

Algorithm 1. An algorithm for matching ontologies O_1 and O_2

Table 1. Matching the news ontologies: sim_{proto}

Concept Names	Comp(2)	Religion(2)	Science(2)	Recr(2)	Electr(2)
Computers(1)	*0.924*	0.188	0.457	*0.514*	*0.6*
Religion(1)	0.175	*0.972*	0.154	0.201	0.191
Politics(1)	0.414	0.246	0.441	*0.522*	0.47
Recreation(1)	*0.539*	0.218	0.369	*0.843*	*0.586*

8 Experiments

In the following, we present experimental results obtained for the news ontologies in Fig. 1. We will start with the methods presented in Section 4, including the cosine similarity for class prototypes, $sim_{proto}(A, B)$ defined in equation 1, the Jaccard similarities $sim_{jaccard}$, defined below and the methods derived from clustering, $sim_{\min}$, $sim_{\max}$, and sim_{avg}. For these methods, we will only present the results for the top-level categories of the news ontologies shown in Figure 1. For the similarity measure based on the VC-dimension, we additionally evaluate the application of the recursive matching procedure (see Section 7).

In the test ontologies, leaf nodes were assigned about 500 documents on the related topic. Parent nodes were assigned the union of the documents assigned to their children plus some additional documents on their topic, in order to account for documents annotated directly by the parent concept. Each top-level has between 1900 and 2500 instances. Each instance is described by 329 features corresponding to a selection of the words occurring in the original news articles. Note that we reduced the initial number of terms substantially by removing stop words, applying stemming, and deleting high and low frequency words. The results are presented in the form of similarity matrices, where underlined entries mark the pairs of concepts that are supposed to be mapped onto each other, like *Computers(1)* and *Computers(2)*. Numbers in italics mark values above the threshold of 0.5. If Alg. 1 establishes a mapping for a pair of classes that are not supposed to be mapped, the mapping might be considered as incorrect.

8.1 Prototype Method

The prototype method 1 based on the CAIMAN idea simply consists in first computing the class (concept) means in the usual manner, and then applying the instance based similarity measure to it. Since all vectors have non-negative feature values, the similarity lies always between 0 and 1.

The similarities for the concepts in the two news ontologies can be found in Table 1. It can be determined that the prototype approach is able to detect similar pairs of concepts. However, it fails to properly detect dissimilar pairs, provided a natural threshold of 0.5 (see e.g. the pair *Politics(1)/Recreation(2)*). This makes it difficult to use this measure for the recursive matching procedure, which relies on being able to make such decisions. We assume that the relatively high similarity values for dissimilar concept pairs result from the averaging

Table 2. Matching the news ontologies: $sim_{jaccard}$

Concept Names	Comp(2)	Religion(2)	Science(2)	Recr(2)	Electr(2)
Computers(1)	*0.597*	0.03	0.14	0.02	0.16
Religion(1)	0.0	*0.99*	0.0	0.0	0.0
Politics(1)	0.008	0.003	0.12	0.02	0.005
Recreation(1)	0.02	0.04	0.007	*0.86*	0.06

process that is used for computing the means: compared to the more "extremal" vectors in each class, the means tend to be more similar.

8.2 Jaccard Similarity-Based Method

For completeness of our proof of concept, we tested the similarities of the first levels of the two news ontologies by the help of one of the most popular measures found in the extensional OM literature, the Jaccard similarity which is in the core of the GLUE matcher, given by

$$sim_{jaccard}(A, B) = \frac{P(A \cap B)}{P(A \cup B)}. \tag{8}$$

The quantities $P(A \cap B)$ and $P(A \cup B)$ were estimated by learning an SVM on the instances of O_1 and testing it on the instances of O_2 and vice versa, as explained in [4]. The concept similarities are found in Table 2. The results by using the *corrected* Jaccard coefficient, suggested by [9], were similar to the ones presented here. Below, these results will be compared with the results achieved with the proposed approaches.

8.3 Comparing the Sets of Support Vectors (SSV)

In the following, we present the results for comparing the sets of support vectors. The similarities for the sets are based on minimizing, maximizing, and averaging the similarities of pairs of instances for the two sets to be compared. The results can be found in Table 3.

The similarity values of $sim_{\min}$ are relatively low for all considered concept pairs and it also fails to correctly map *Recreation(1)* to *Recreation(2)*. $sim_{\max}$ shows the opposite effect and judges the similarity of all concept pairs as relatively high. For instance, the similarity value of *Recreation(2)* and *Politics(1)* is equal to 0.886 and thus much higher than the natural threshold of 0.5. sim_{avg} also attains relatively low values for all concept pairs, but in contrast to $sim_{\min}$ it can at least determine the most similar concepts for each concept correctly. Our conclusion is that all three measures present problems when being used in the matching procedure, since they require the user to choose a suitable threshold different from 0.5 for discriminating between similar and dissimilar concept pairs. Note that we also applied all three similarity measures to the full instance sets instead of the sets of support vectors. The findings were quite similar, but computation times were much higher.

Table 3. Matching the news-ontologies: $sim_{\min}$, $sim_{\max}$, and sim_{avg}

Concept Names	Comp(2)	Religion(2)	Science(2)	Recr(2)	Electr(2)
Computers(1)	0.00269	$1.97 \cdot 10^{-9}$	$3.03 \cdot 10^{-9}$	$6.99 \cdot 10^{-9}$	$4.76 \cdot 10^{-9}$
Religion(1)	$8.27 \cdot 10^{-9}$	0.0181	$1.29 \cdot 10^{-6}$	$4.46 \cdot 10^{-9}$	$4.44 \cdot 10^{-9}$
Politics(1)	$7.33 \cdot 10^{-9}$	$1.93 \cdot 10^{-9}$	$8.23 \cdot 10^{-9}$	$4.35 \cdot 10^{-9}$	$7.02 \cdot 10^{-9}$
Recreation(1)	$5.98 \cdot 10^{-9}$	$2.82 \cdot 10^{-9}$	$5.37 \cdot 10^{-9}$	$5.49 \cdot 10^{-9}$	$4.46 \cdot 10^{-9}$
Computers(1)	*0.889*	*0.503*	*0.713*	*0.892*	*0.917*
Religion(1)	*0.604*	*0.874*	*0.543*	*0.679*	0.629
Politics(1)	*0.867*	*0.568*	*0.839*	*0.886*	0.885
Recreation(1)	*0.921*	*0.536*	*0.746*	*1*	*0.919*
Computers(1)	0.0175	0.008	0.0122	0.0138	0.0148
Religion(1)	0.00808	0.0216	0.00848	0.00907	0.0086
Politics(1)	0.013	0.0109	0.0133	0.0147	0.0137
Recreation(1)	0.0164	0.00962	0.0133	0.0198	0.0165

Table 4. Matching the news ontologies by selecting variables with MI

Concept Names	Computers(2)	Religion(2)	Science(2)	Recr(2)	Electr(2)
Computers(1)	*0.548*	−0.405	0.119	0.146	0.464
Religion(1)	−0.242	*0.659*	−0.105	−0.201	−0.271
Politics(1)	−0.105	0.019	0.276	0.138	−0.015
Recreation(1)	0.318	−0.301	0.001	*0.528*	0.356

8.4 Similarity Based on VC Dimension (VC-VS)

We present an evaluation of the instance-based matching procedure suggested in Section 7 as well as of the finding that the VC dimension of SVMs can provide a criterion for selecting variables in a classification task. As a measure of similarity we have used Pearson's measure of correlation (wherefrom the negative numbers), given in (5). The measure indicates high similarity for positive values and low similarity for non-positive values. The results achieved by using the novel SVM-based variable selection technique (Table 5) proved to be better than those achieved by a standard point-wise mutual information-based criterion (Table 4). The results presented below come from using the former method.

After the root nodes of O_1 and O_2 have been matched, we proceeded to match the sets of their direct descendants. The results are shown in the upper similarity matrix on Table 5. Our similarity criterion identified successfully the three pairs of similar concepts (the Computers, the Religion, and the Recreation pairs). Following the matching procedure, as a second step we matched the sets of the descendants of the concepts that were found to be similar in the first step. The obtained similarity values for the children of the computer- and recreation-classes are shown in Table 6. Finally, in order to show the effect of not respecting the rule of hierarchical matching, we have matched the first level of O_2 with the descendants of the computer class in O_1. The obtained similarity values are shown in the lower matrix in Table 5: although the potentially similar classes

Table 5. Matching the news-ontologies by selecting variables with VC-SVM: first levels vs. mixed levels

Concept Names	Comp(2)	Religion(2)	Science(2)	Recr(2)	Elec(2)
Computers(1)	*0.78*	0.08	0.04	0.40	0.06
Religion(1)	0.07	*0.94*	0.02	0.10	0.01
Politics(1)	0.08	0.12	0.08	0.14	0.01
Recreation(1)	0.29	0.09	0.06	*0.60*	0.06
Graphics(1)	0.30	-0.01	-0.1	-0.01	-0.002
HW:PC(1)	0.18	-0.03	-0.01	-0.02	-0.03
HW:Mac(1)	0.03	-0.02	-0.01	-0.02	-0.09

Table 6. Similarities of the descendants of the computer- and the recreation-classes

Concept Names	Graphics(2)	HW-Mac(2)	HW-PC(2)
Graphics(1)	*0.954*	-0.475	-0.219
HW-Mac(1)	-0.266	*0.501*	-0.073
HW-PC(1)	-0.577	0.304	*0.556*

Concept Names	Autos(2)	Motorcycles(2)	Baseball	Hockey
Autos(1)	*0.978*	0.478	0.117	0.095
Motorcycles(1)	*0.560*	*0.989*	0.121	0.452
Sports	0.452	0.491	*0.754*	*0.698*

are accorded higher similarity values, the similarity coefficients are much lower than in the previous cases and much closer to the values for the dissimilar classes.

The similarity values are computed on the sets of input variables which, in case of text, correspond to actual words. Thus, the most important words that discriminate between a pair of similar concepts and the rest of the pairs of concepts can be readily made available in contrast to related methods (e.g. Caiman or Glue). For example, our selection procedure found out that among the most important tokens that characterize the concept *Computers* are *comp*, *chip*, *graphic*, *card*, *devic*, *file*. In contrast, the features with highest scores for *Religion* were *christ*, *church*, *faith*, *bibl*, *jesu*, for *Politics* – *polit*, *govern*, *legal*, *talk* and for *Recreation* – *motorcycl*, *auto*, *speed* and *engin*. This information is useful to verify the quality and coherence of the matching results.

9 Conclusion

The paper focuses on extension-grounded approaches to identify cross-ontology concept similarities by applying machine learning techniques for classification. Four similarity criteria have been tested on two source ontologies populated with textual instances: one based on comparing the support vectors learned per concept (SSV), one based on variable selection with VC-dimension (VC-VS), the Jaccard similarity used in the Glue tool and one, prototype method based on the Caiman system. A matching procedure, using one of the proposed measures has been described and evaluated.

Our results showed that the VC-VS method outperforms the other considered measures, yielding a clearcut difference between the similarity values obtained for pairs of similar and pairs of dissimilar concepts. The method shows to respond properly to a natural similarity threshold of 0.5 on a 0-1 scale. Although the results achieved with the Jaccard similarity are competitive, the VC-VS approach makes a step further in terms of similarity verification, since the discriminant features (words) for each class are readily made available. The SSV technique, although inferior to VC-VS, tends to outperform the prototype method, provided an appropriate choice of similarity threshold from the user.

References

1. Alpaydin, E.: Introduction to Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, Cambridge (2004)
2. Bi, J., Bennett, K., Embrechts, M., Breneman, C., Song, M.: Dimensionality reduction via sparse support vector machines. JMLR 3, 1229–1243 (2003)
3. Burges, C.: A tutorial on support vector machines for pattern recognition. DMKD 2, 121–167 (1998)
4. Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. In: WWW 2002, pp. 662–673. ACM Press, New York (2002)
5. Euzenat, J., Shvaiko, P.: Ontology Matching, 1st edn. Springer, Heidelberg (2007)
6. Goldstone, R.L., Rogosky, B.J.: Using relations within conceptual systems to translate across conceptual systems. Cognition 84(3), 295–320 (2002)
7. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. JMLR 3(1), 1157–1182 (2003)
8. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. Mach. Learn. 46(1-3), 389–422 (2002)
9. Isaac, A., van der Meij, L., Schlobach, S., Wang, S.: An empirical study of instance-based ontology matching. In: Aberer, K., Choi, K.-S., Noy, N., Allemang, D., Lee, K.-I., Nixon, L.J.B., Golbeck, J., Mika, P., Maynard, D., Mizoguchi, R., Schreiber, G., Cudré-Mauroux, P. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 253–266. Springer, Heidelberg (2007)
10. Joachims, T.: SVM light (2002), http://svmlight.joachims.org
11. Lacher, M.S., Groh, G.: Facilitating the exchange of explicit knowledge through ontology mappings. In: Proceedings of the 14th FLAIRS Conf., pp. 305–309. AAAI Press, Menlo Park (2001)
12. Mitra, P., Wiederhold, G.: An ontology composition algebra. In: Staab, S., Studer, R. (eds.) Handbook on Ontologies. Springer, Heidelberg (2004)
13. Noy, N., Musen, M.: Anchor-prompt: Using non-local context for semantic matching. In: IJCAI 2001, August 2001, pp. 63–70 (2001)
14. Rakotomamonjy, A.: Variable selection using svm based criteria. JMLR 3, 1357–1370 (2003)
15. Stumme, G., Maedche, A.: Fca-merge: Bottom-up merging of ontologies. In: IJCAI, pp. 225–230 (2001)
16. Todorov, K., Geibel, P., Kühnberger, K.-U.: Extensional ontology matching with variable selection for support vector machines. In: CISIS, pp. 962–968. IEEE Computer Society Press, Los Alamitos (2010)
17. Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., Vapnik, V.: Feature Selection for SVMs. In: Advances in Neural Information Processing Systems, vol. 13, pp. 668–674. MIT Press, Cambridge (2001)

Re-mining Positive and Negative Association Mining Results

Ayhan Demiriz[1], Gurdal Ertek[2], Tankut Atan[3], and Ufuk Kula[1]

[1] Sakarya University
Sakarya, Turkey
{ademiriz,ufukkula}@gmail.com
[2] Sabanci University
Istanbul, Turkey
ertekg@sabanciuniv.edu
[3] Isik University
Istanbul, Turkey
tatan@isikun.edu.tr

Abstract. Positive and negative association mining are well-known and extensively studied data mining techniques to analyze market basket data. Efficient algorithms exist to find both types of association, separately or simultaneously. Association mining is performed by operating on the transaction data. Despite being an integral part of the transaction data, the pricing and time information has not been incorporated into market basket analysis so far, and additional attributes have been handled using quantitative association mining. In this paper, a new approach is proposed to incorporate price, time and domain related attributes into data mining by re-mining the association mining results. The underlying factors behind positive and negative relationships, as indicated by the association rules, are characterized and described through the second data mining stage *re-mining*. The applicability of the methodology is demonstrated by analyzing data coming from apparel retailing industry, where price markdown is an essential tool for promoting sales and generating increased revenue.

1 Introduction

Association mining is a data mining technique in which the goal is to find rules in the form of $X \Rightarrow Y$, where X and Y are two non-overlapping sets of items or events, depending on the domain. A rule is considered as significant if it is satisfied by at least a percentage of cases specified beforehand (`minimum support`) and its confidence is above a certain threshold (`minimum confidence`). Conventional association mining considers "positive" relations as in the rule $X \Rightarrow Y$. However negative associations such as $X \Rightarrow \neg Y$, where $\neg Y$ represents the negation (absence) of Y, might also be discovered through association mining.

Association mining has contributed to many developments in a multitude of data mining problems. Recent developments have positioned the association mining as one of the most popular tools in retail analytics, as well [1]. Traditionally,

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 101–114, 2010.

association mining generates positive association rules that reveal complementary effects. In other words, the rules suggest that purchasing an item can generate sales of other items. Association mining can also be used to find so-called "Halo effects", where reducing the price of an item can entice and increase the sales of another item. Although positive associations are an integral part of retail analytics, negative associations are not. However negative associations are highly useful to find out the substitution effects in a retail environment. Substitution means that a product is purchased instead of another one.

There have been numerous algorithms introduced to find positive and negative associations since the pioneering work of Agrawal et al. [2]. Market basket analysis is considered as a motivation and a test bed for these algorithms. Since the price data are readily available in the market basket data, one might expect to observe the usage of price data in various applications. Conceptually quantitative association mining [3,4] can handle pricing data and other attribute data. However pricing data have not been utilized before as a quantitative attribute except in [5], which explores a solution with the help of singular value decomposition. Quantitative association mining is not the only answer to analyze the attribute data by conventional association mining. Multidimensional association mining [4] is also a methodology that can be adapted in analyzing such data. Inevitably, the complexity of association mining will increase with the usage of additional attribute data where there might be both categorical and quantitative attributes in addition to the transaction data [6]. Even worse, the attribute data might be less sparse compared to transaction data.

The main objective of this paper is to develop an efficient methodology that enables incorporation of attribute data (e.g. price, category, sales timeline) to explain both positive and negative item associations. Positive and negative item associations indicate the complementarity and substitution effects respectively. To the best of our knowledge, there exists no methodological research in data mining literature that enables such a multi-faceted analysis to be executed efficiently and is proven on real world attribute data. A practical and effective methodology is developed to discover how price, item, domain and time related attributes affect both positive and negative associations by introducing a new data mining process.

As a novel and broadly applicable concept, we define *data re-mining* as mining a newly formed data from the results of an original data mining process. Aforementioned newly formed data will contain additional attributes on top of the original data mining results. These attributes in our case will be related to price, item, domain and time. Our methodology combines pricing as well as other information with the original association mining results within the framework of a new mining process. We thereby generate new rules to characterize, describe and explain the underlying factors behind positive and negative associations. Re-mining is a different process from post-mining where the latter only summarizes the data mining results. For example visualizing the association mining results [7] could be regarded as a post-mining activity. Our methodology extends and generalizes post-mining process.

Our work contributes to the field of data mining in three ways:

1. We introduce a new data mining concept and its associated process, named as *Re-Mining*, which enables an elaborate analysis of both positive and negative associations for discovering the factors and explaining the reasons for such associations.
2. We enable the efficient inclusion of price data into the mining process in addition to other attributes of the items and the application domain.
3. We illustrate that the proposed methodology is applicable to real world data from apparel retailing.

The remainder of the paper is organized as follows. In Section 2, an overview of the basic concepts in related studies is presented through a concise literature review. In Section 3, Re-Mining is motivated, defined, and framed. The methodology is put into use with apparel retail data and its applicability is demonstrated in Section 4. In Section 5, the limitations of the quantitative association regarding the retail data used in this paper are shown. Finally, Section 6 summarizes our work and discusses the future directions.

2 Related Literature

Data mining can simply be defined as extracting knowledge from large amounts of data [4]. An extended definition additionally requires that findings are meaningful, previously unknown and actionable [8]. Interpreting the results is an essential part of the data mining process and can be achieved through the post-mining analysis of multi-dimensional association mining results.

Quantitative and multi-dimensional association mining techniques can integrate attribute data into the association mining process where the associations among these attributes are also found. However, these techniques introduce significant additional complexity, since association mining is carried out with the complete set of attributes rather than just the market basket data. In the case of quantitative association mining quantitative attributes are transformed into categorical attributes through discretization, transforming the problem into multi-dimensional association mining with only categorical attributes. This is an *NP-Complete* problem as shown in [6], with the exponentially increasing running time as the number of additional attributes increases linearly.

In re-mining *single dimensional* rules are generated and then expanded with additional attributes. Multi-dimensional association mining, on the other hand, works directly towards the generation of multi-dimensional rules. It relates all the possible categorical values of all the attributes to each other. In our methodology, the attribute values are investigated only for the positively and negatively associated item pairs, with much less computational complexity.

2.1 Negative Association Mining

The term association mining is generally used to represent positive association mining. Since positive association mining has been studied extensively,

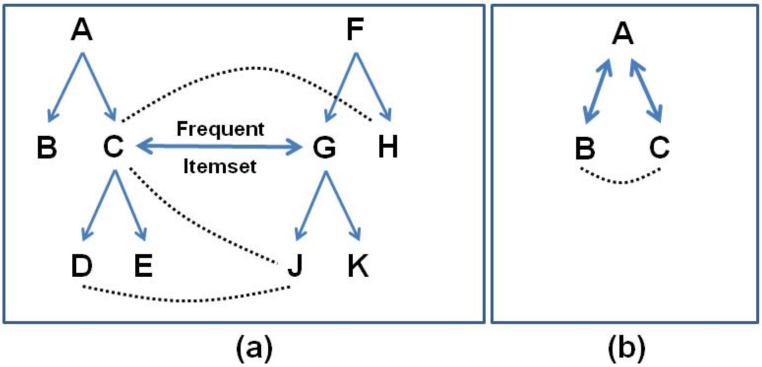


Fig. 1. (a) Taxonomy of Items and Associations [9]; (b) Indirect Association

we limit ourselves to review some of the approaches described in the literature for finding negative associations. One innovative approach utilizes the domain knowledge of item hierarchy (taxonomy), and seeks negative association between items in a pairwise way [9]. Authors in [9] propose the interestingness measure (RI) based on the difference between expected support and actual support: $RI = \frac{E[s(XY)]-s(XY)}{s(X)}$. A minimum threshold is specified for the interestingness measure RI besides the minimum support threshold for the candidate negative itemsets. Depending on the taxonomy (e.g. Figure 1(a)) and the frequent itemsets, candidate negative itemsets can be generated. For example, assuming that the itemset $\{CG\}$ is frequent in Figure 1(a), the dashed curves represent some candidate negative itemsets.

In [10], negative associations are found through indirect associations. Figure 1(b) depicts such an indirect association $\{BC\}$ via item A. In Figure 1(b) itemsets $\{AB\}$ and $\{AC\}$ are both assumed to be frequent, whereas the itemset $\{BC\}$ is not. The itemset $\{BC\}$ is said to have an *indirect association* via the item A and thus is considered as a candidate negative association. Item A in this case is called as a *mediator* for the itemset $\{BC\}$. Just like the aforementioned method in [9], indirect association mining also uses an interestingness measure -*dependency* in this case- as a threshold. Indirect mining selects as candidates the frequent itemsets that have strong dependency with their mediator.

Even though both methods discussed above are suitable for retail analytics, the approach in [10] is selected in our study to compute negative associations due to convenience of implementation.

2.2 Quantitative and Multi-dimensional Association Mining

The traditional way of incorporating quantitative data into association mining is to discretize (categorize) the continuous attributes. An early work by Srikant and Agrawal [3] proposes such an approach where the continuous attributes are first partitioned and then treated just like categorical data. For this, consecutive integer values are assigned to each adjacent partition. In case the quantitative attribute has few distinct values, consecutive integer values can be assigned to these few values to conserve the ordering of the data. When there is not enough support for a partition, the adjacent partitions are merged and the mining process is rerun. Although [3] emphasizes rules with quantitative attributes on the left hand side (antecedent) of the rules, since each partition is treated as if it were categorical, it is also possible to obtain rules with quantitative attributes on the right hand side (consequent) of the rules.

A more statistical approach is followed in [11] for finding association rules with quantitative attributes. The rules found in [11] can contain statistics (mean, variance and median) of the quantitative attributes.

As discussed earlier, re-mining does not investigate every combination of attribute values, and is much faster than quantitative association mining. In Section 5, quantitative association mining is also carried out for the sake of completeness.

Finally, in [5], where the significant ratio rules are found to summarize the expenses made on the items. An example of ratio rule would be "*Customers who buy bread:milk:butter spend 1:2:5 dollars on these items*" [5]. This is a potentially useful way of utilizing the price data for unveiling the hidden relationships among the items in sales transactions. According to this approach, one can basically form a price matrix from sales transactions and analyze it via singular value decomposition (SVD) to find positive and negative associations. Ensuring the scalability of SVD in finding the ratio rules is a significant research challenge.

2.3 Learning Association Rules

In [12,13] a framework is constructed based on the idea of learning a classifier to explain the mined results. However, the described framework considers and interprets only the positive association rules, whether they are interesting or not. The approach obligates human intervention for labeling the generated rules as interesting or not. The framework in [12,13] is the closest work in the literature to our re-mining approach. However our approach is very different in the sense that it includes negative associations and is suitable for the automated rule discovery to explain the originally mined results. Unlike in [12,13], our approach is applied to a real world dataset.

Based on correlation analysis, authors in [14] propose an algorithm to classify associations as positive and negative. The learning is only used on correlation data and the scope is narrowly determined to label the associations as positive or negative.

3 The Methodology

In this section we introduce the proposed methodology, which transforms the post-mining step into a new data mining process. The re-mining algorithm consists of following steps.

1. *Perform association mining.*
2. *Sort the items in the 2-itemsets.*
3. *Label the item associations accordingly and append them as new records.*
4. *Expand the records with additional attributes for re-mining.*
5. *Perform exploratory, descriptive, and predictive re-mining.*

Fig. 2. Re-Mining Algorithm

Potentially, re-mining algorithm can be conducted in explanatory, descriptive, and predictive manners. Re-mining can be considered as an additional data mining step of KDD process [8]. We define *re-mining* process as combining the results of an original data mining process with a new set of data and then mining the newly formed data again. Conceptually, re-mining process can be extended with many more repeating steps since each time a new set of the attributes can be introduced and a new data mining technique can be utilized. However the re-mining process is limited to only one additional data mining step in this paper. In theory, the new mining step may involve any appropriate set of data mining techniques.

Data mining does not require any pre-assumptions (hypotheses) about the data. Therefore the results of data mining may potentially be full of surprises. Making sense of such large body of results and the pressure to find surprising insights may require incorporating new attributes and subsequently executing a new data mining step, as implemented in re-mining. The goal of this re-mining process is to explain and interpret the results of the original data mining process in a different context, by generating new rules from the consolidated data. The results of re-mining need not necessarily yield an outcome in parallel with the original outcome. In other words, if the original data mining yields for example frequent itemsets, it is not expected from re-mining to output frequent itemsets again.

The main contribution of this paper is to introduce the re-mining process to discover new insights regarding the positive and negative associations. However the usage of re-mining process is not limited to this. We can potentially employ the re-mining process to bring further insights to the results of any data mining task.

The rationale behind using the re-mining process is to exploit the domain specific knowledge in a new step. One can understandably argue that such background knowledge can be integrated into the original data mining process by introducing new attributes. However, there might be certain cases that adding such information would increase the complexity of the underlying model [6], and diminish the strength of the algorithm. To be more specific, it might be necessary to find attribute associations when the item associations are also present, which requires constraint-based mining [15]. Re-mining may help with grasping the causality effects that exist in the data as well, since the input of the causality models may be an outcome of the another data mining process.

4 Case Study

In this section the applicability of the re-mining process is demonstrated through a real world case study that involves the store level retail sales data originating from an apparel retail chain. In our study, we had access to complete sales, stock and transshipment data belonging to a single merchandise group (men's clothes line) coming from the all stores of the retail chain (over 200 of them across the country) for the 2007 summer season. Throughout the various stages of the study, MS SQL Server, SAS, SPSS Clementine, Orange, and MATLAB software packages have been used as needed.

First, the retail data and its data model used in this study are described. Then the application of the re-mining methodology on the given dataset is presented. Re-mining reveals the profound effect of pricing on item associations and frequent itemsets, and provides insights that the retail company can act upon.

4.1 Retail Data

A typical product hierarchy for an apparel retailer displays the following sequence: *merchandise group*, *sub-merchandise group*, *category*, *model*, and *SKU*. The products at the Stock Keeping Unit (*SKU*) level are sold directly to the customers. At the *SKU* level, each color and size combination that belongs to a *model* is assigned a unique SKU number. A *category* is composed of similar models, e.g. long sleeve shirts. A *merchandise group* can represent the whole product line for a gender and age group, e.g. men's clothes line. A *sub-merchandise group* divides this large group. Notice that this particular product hierarchy is just a typical representation and the hierarchy may vary from one firm to another.

Since the SKU level store data exhibit high variability, the dataset is aggregated at the model level, the immediate parent of the SKU level. Sales transaction data consist of a collection of rows generated by a sale, that includes an item numbers, and their prices, a transaction identifier, and a time stamp. Positively and negatively associated item pairs can thus be found using transactional data. For apparel products, price is an important factor influencing the purchasing decisions of consumers, and markdown management (planning the schedule and price levels of discounts) is an essential activity for increasing the revenue of an apparel chain. Consequently, pricing is an important driver of the multiple-item sales transactions and is highly relevant to association mining activities in apparel retailing.

Out of the 710 models available in the dataset, the top 600 models have been selected according to sales quantities. Most of the sales consist of single-item purchases. There exist 2,753,260 transactions and 4,376,886 items sold. Technically it is hard to find positive associations in sparse data and the sparsity of the data is very high with ∼99.74%. Although single-item transactions could be removed from a traditional association mining task, they are included in the case study to observe the effect of the pricing. In other words, inclusion of single item purchases does not change the association mining statistics (support values), yet enables accurate calculation of the values of additional attributes.

For example, for calculating the average price of an item across the season, one will obtain a more accurate statistic when single item transactions are also included.

4.2 Conventional Association Mining

As the first step of re-mining methodology, conventional association mining has been conducted. Apriori algorithm was run with a minimum support count of 100 to generate the frequent itemsets. All the 600 items were found to be frequent. In re-mining, only the frequent 2-itemsets have been investigated and 3930 such pairs have been found. Thus frequent itemsets were used in the analysis, rather than association rules. The top-5 frequent pairs given in Table 1 have support counts of 22131, 17247, 17155, 14224, and 11968 respectively, within the 2,753,260 transactions. We utilize the retail data in transactional format as known in conventional association mining. Item names are replaced by the alphabet letters for brevity.

Table 1. Top-5 Frequent Pairs

Item 1	Item 2	S Count
A	B	22131
B	F	17247
A	E	17155
B	E	14224
C	B	11968

The frequent pairs were then used in finding the negatively related pairs via indirect association mining. Negative relation is an indicator of product substitution. Implementing indirect association mining resulted in 5,386 negative itemsets, including the mediator items. These itemsets were reduced to a set of 2,433 unique item pairs when the mediators were removed. This indeed shows that a considerable portion of the item pairs in the dataset are negatively related via more than one mediator item.

4.3 Re-mining the Expanded Data

Following conventional association mining, a new data set E^* was formed from the item pairs and their additional attributes A^* for performing exploratory, descriptive and predictive re-mining. In this paper, we only illustrate descriptive re-mining by using decision tree analysis and a brief exploratory re-mining example due to space considerations. As a supervised classification method, decision tree approach usually requires the input data in a table format, in which one of the attributes is the class label. The type of association, positive (‘+’) or negative (‘-’), was selected as the class label in our analysis.

An item pair can generate two distinct rows for the learning set - e.g. pairs AB and BA, but this representation ultimately yields degenerate rules out of learning process. One way of representing the pair data is to order (rank) items in the pair according to a sort criterion. In the case study, sort attribute was selected as the price, which marks the items as higher and lower priced items, respectively.

For computing price-related statistics (averages and standard deviations) a price-matrix was formed out of the transaction data. The price-matrix resembles the full-matrix format of the transaction data with the price of the item replacing the value of **1** in the full-matrix. The price-matrix was normalized by dividing each column by its maximum value, enabling comparable statistics. A price value of 0 in the price-matrix means that the item is not sold in that transaction. The full price-matrix has the dimensions $2,753,260 \times 600$.

Besides price related statistics such as minimum, maximum, average and standard deviations of item prices (MinPriceH, MaxPriceH, MinPriceL, MaxPriceL, AvgPriceH_H1_L0, ..., StdDevPriceH_H1_L0, ...), attributes related with time and product hierarchy were appended, totaling to 38 additional attributes. All the additional attributes have been computed for all sorted item-pairs through executing relational queries.

Once the new dataset is available for the re-mining step, any decision tree algorithm can be run to generate the descriptive rules. Decision tree methods such as C5.0, CHAID, CART are readily available within data mining software in interactive and automated modes. An interactive decision tree is an essential tool for descriptive discovery, since it grows with user interaction and enables the verification of the user hypotheses on the fly.

If decision tree analysis is conducted with all the attributes, support count related attributes would appear as the most significant ones, and data would be perfectly classified. Therefore it is necessary to exclude the support item attributes from the analysis to arrive a conclusive description of the data.

The C5.0 algorithm was executed in automated mode with the default settings, discovering 53 rules for the '-' class, and 11 rules for the '+' class (the default class). One example of the rule describing the '-' class is as follows: **If** StartWeekH > 11 **and** AvgPriceL_H0_L1 > 0.844 **and** CategoryL = 0208 **and** CorrNormPrice_HL ≤ 0.016 **Then** '-'. This rule reveals that when

- the higher priced is sold after 11^{th} calendar week *and*
- average (normalized) price of lower priced item is greater than 0.844 (that corresponds to roughly maximum 15% discount on average) when it is sold and the higher priced item is not sold *and*
- category of lower priced item is "0208" *and*
- correlation coefficient between normalize prices of higher and lower price items is less than or equal to 0.016 then the target class is '-'.

An example of '+' class rule is that **If** MaxPriceH ≤ 14.43 **and** StdDevPriceH_H1_L0 ≤ 0.05 **Then** '+'. Another example of the '+' class rule is the rule **If** LifeTimeH > 23 **and** LifeTimeL <= 21 **and** MaxPriceH > 12.21 **and** CorrNormPrice_HL > 0.003 **Then** '+'. This rule basically emphasizes on the lifetime of the higher priced item and its maximum price with respect to item association.

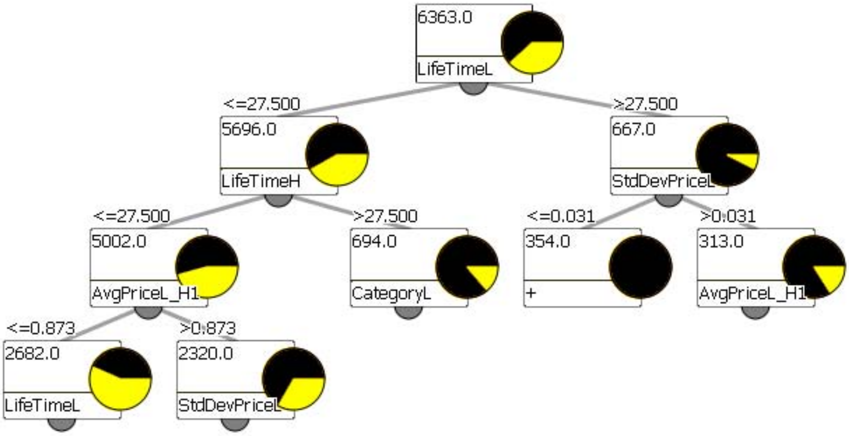


Fig. 3. An Illustrative Decision Tree Model in Re-Mining

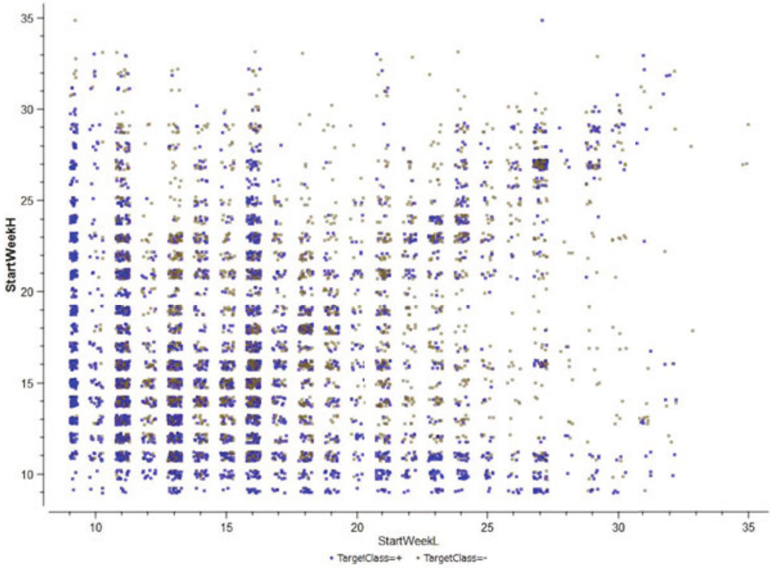


Fig. 4. Exploratory Re-Mining Example: Analyzing Item Introduction in Season

The decision tree in Figure 3 is obtained by interactively pruning the full-tree. It can be observed that LifeTimeL, the lifetime of the lowered price item (in weeks) is the most significant attribute to classify the dataset. If LifeTimeL greater than or equal to 27.5 weeks,the item pair has much higher chance of having positive association.

As seen from Figure 4, positive ("Target Class=+") and negative ("Target-Class=-") associations can be separated well at some regions of the plot. For example, if StartWeekL is around 5 (calendar week) then the associations are mainly positive regardless of the starting week of the higher priced item. In addition, a body of positive associations up to 'StartWeekL=15' can be seen in Figure 4. It means that there is a high possibility of having a positive association between two items when the lower priced item is introduced early in the season. We can maybe conclude that since basic items are usually introduced early in the season, there is a big chance of having a positive association between two items when the lower priced item is a basic one but not a fashion one.

Similar conclusions can be attained when the maximum prices of items considered are compared in Figure 5. It can easily be seen that negative associations usually occur when the maximum prices of items are higher than 25. It should be noted that the basic items may be priced below 20-25 range. Thus it is very likely that many of the fashion items might have negative associations between themselves.

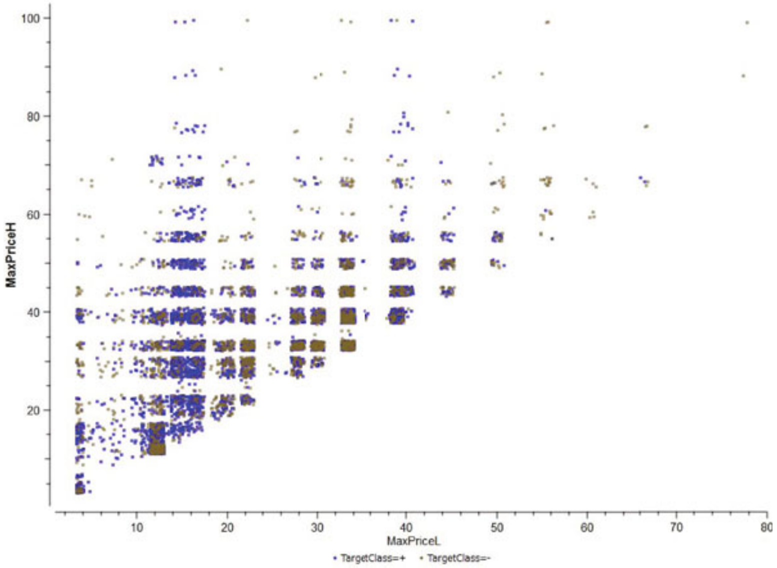


Fig. 5. Exploratory Re-Mining Example: Effect of the Maximum Item Price

5 Comparison with Quantitative Association Mining

Additional attribute data used can also be incorporated through quantitative association mining, as an alternative to re-mining. This type of analysis has also been conducted to illustrate its limitations on analyzing retail data. Quantitative association mining has been studied extensively in the literature and some of the major applications like [3] were reviewed in Section 2.2. After an initial analysis of the price data, it was observed that there are not too many price levels for the products. Therefore a straight-forward discretization, which does not require a complex transformation, exists.

Notice that one of the most important steps in the quantitative association mining is the discretization step. Instead of utilizing a more complex quantitative association mining, we take the liberty to conjoin the item and the price information into a new entity to conduct our quantitative association analysis in a simplified way.

Two main seasons in apparel retailing, winter and summer have approximately equal length. As a common business rule, prices are not marked down too often. Usually, at least two weeks pass by between subsequent markdowns. Thus, there exist a countable set of price levels within each season. There might be temporary sales during the season, but marked down prices remain the same until the next price markdown. Prices are set at the highest level in the beginning of each season, falling down by each markdown. If the price data of a product is normalized by dividing by the highest price, normalized prices will be less than or equal to 1. When two significant digits are used after the decimal point, prices can be easily discretized. For example, after an initial markdown of 10%, the normalized price will be 0.90. Markdowns are usually computed on the original highest price. In other words, if the second markdown is 30% then the normalized price is computed as 0.70.

Table 2. Top-10 Frequent Pairs for the Quantitative Data

Pair ID	Item 1	Item 2	Sup. Count
1	A_0.90	B_0.63	14312
2	A_0.90	E_0.90	10732
3	B_0.63	E_0.90	8861
4	C_0.90	B_0.63	7821
5	A_1.00	B_0.70	7816
6	A_1.00	E_1.00	6377
7	B_0.63	F_0.90	5997
8	B_0.70	E_1.00	5344
9	B_0.63	F_0.78	5318
10	D_0.90	B_0.63	4402

After using this discretization scheme, 3,851 unique product-normalized price pairs have been obtained for the 600 unique products of the original transaction data. Each product has 6 price levels on the average. The highest number of price levels for a product is 14 and the lowest number of price levels is 2. This shows that markdowns were applied to all the products and no product has been sold at its original price throughout the whole season. Technically, a discretized price can be appended to a corresponding product name to create a new unique entity for the quantitative association mining. For example appending 0.90 to the product name 'A' after an underscore will create the new entity 'A_0.90'. One can easily utilize the conventional association mining to conduct a discretized quantitative association mining after this data transformation (discretization).

The top-10 frequent pairs and their support counts are depicted in Table 2 where item IDs are masked. As can be observed from Table 2, a large portion of the frequent purchases occurs at the discounted prices. Among the top-10 frequent item pairs, only the sixth one has full prices for both of the items. The remaining pairs are purchased at marked down (discounted) prices.

The retail company does not allow price differentiations by locations at any give time. In other words, an item will have the same price across all the stores at any given time. Quantitative association mining can identify negative associations between items for value combinations that actually never occur. For example, even though an item A is sold only at the normalized price of 0.70 in the time interval that B is sold, quantitative association mining can still suggest other price combinations of A and B, such as (A_1.00, B_1.00) as negative item pairs. Thus, many item-price combinations will have negative associations due to the nature of the business and the way quantitative association mining operates, yielding misleading results.

Even though both positive and negative quantitative association mining can be run on any given dataset conceptually, it is not guaranteed to yield useful outcomes. Alternatively, re-mining operates only on the confirmed positive and negative quantitative associations and does not exhibit the discussed problem.

6 Conclusion

A framework, namely re-mining, has been proposed to enrich the original data mining process with a new set of data and an additional data mining step. The goal is to describe and explore the factors behind positive and negative association mining, and to predict the type of associations based on attribute data. It is shown that not only categorical attributes (e.g. category of the product) but also quantitative attributes such as price, lifetime of the products in weeks and some derived statistics, can be included in the study while avoiding NP-completeness.

The framework has been demonstrated through a case study in apparel retail industry. Only descriptive and a brief exploratory re-mining have been performed in this paper. Predictive re-mining can also be conducted, and this is planned as a future study. Our case study has revealed some interesting outcomes such as the negative associations are usually seen between fashion items and the price of an item is an important factor for the item associations in apparel retailing. The scope of the current study has been limited to the proof of concept and the future work is planned towards extending the retail analytics to include re-mining as an important component.

Acknowledgement

This work is financially supported by the Turkish Scientific Research Council under Grant TUBITAK 107M257.

References

1. Brijs, T., Swinnen, G., Vanhoof, K., Wets, G.: Building an association rules framework to improve product assortment decisions. Data Min. Knowl. Discov. 8(1), 7–23 (2004)
2. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Buneman, P., Jajodia, S. (eds.) SIGMOD Conference, pp. 207–216. ACM Press, New York (1993)
3. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In: Jagadish, H.V., Mumick, I.S. (eds.) SIGMOD Conference, pp. 1–12. ACM Press, New York (1996)
4. Han, J., Kamber, M.: Data Mining Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
5. Korn, F., Labrinidis, A., Kotidis, Y., Faloutsos, C.: Quantifiable data mining using ratio rules. VLDB J. 8(3-4), 254–266 (2000)
6. Angiulli, F., Ianni, G., Palopoli, L.: On the complexity of inducing categorical and quantitative association rules. Theoretical Computer Science 314(1-2), 217–249 (2004)
7. Ertek, G., Demiriz, A.: A framework for visualizing association mining results. In: Levi, A., Savaş, E., Yenigün, H., Balcısoy, S., Saygın, Y. (eds.) ISCIS 2006. LNCS, vol. 4263, pp. 593–602. Springer, Heidelberg (2006)
8. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery: An overview. In: Advances in Knowledge Discovery and Data Mining, pp. 1–34. AAAI Press, Menlo Park (1996)
9. Savasere, A., Omiecinski, E., Navathe, S.: Mining for strong negative associations in a large database of customer transactions. In: Proceedings of the 14th International Conference on Data Engineering, pp. 494–502 (1998)
10. Tan, P.N., Kumar, V., Kuno, H.: Using sas for mining indirect associations in data. In: Western Users of SAS Software Conference (2001)
11. Aumann, Y., Lindell, Y.: A statistical theory for quantitative association rules. J. Intell. Inf. Syst. 20(3), 255–283 (2003)
12. Yao, Y., Zhao, Y., Maguire, R.B.: Explanation-oriented association mining using a combination of unsupervised and supervised learning algorithms. In: Xiang, Y., Chaib-draa, B. (eds.) Canadian AI 2003. LNCS (LNAI), vol. 2671, pp. 527–531. Springer, Heidelberg (2003)
13. Yao, Y., Zhao, Y.: Explanation-oriented data mining. In: Wang, J. (ed.) Encyclopedia of Data Warehousing and Mining. Idea Group Inc., USA (2005)
14. Antonie, M.L., Zaïane, O.R.: An associative classifier based on positive and negative rules. In: Das, G., Liu, B., Yu, P.S. (eds.) DMKD, pp. 64–69. ACM, New York (2004)
15. Ng, R.T., Lakshmanan, L.V.S., Han, J., Pang, A.: Exploratory mining and pruning optimizations of constrained associations rules. In: SIGMOD 1998: Proceedings of the 1998 ACM SIGMOD international conference on Management of data, pp. 13–24. ACM, New York (1998)

Multi-Agent Based Clustering: Towards Generic Multi-Agent Data Mining

Santhana Chaimontree, Katie Atkinson, and Frans Coenen

Department of Computer Science
University of Liverpool, UK
{S.Chaimontree,katie,Coenen}@liverpool.ac.uk

Abstract. A framework for Multi Agent Data Mining (MADM) is described. The framework comprises a collection of agents cooperating to address given data mining tasks. The fundamental concept underpinning the framework is that it should support generic data mining. The vision is that of a system that grows in an organic manner. The central issue to facilitating this growth is the communication medium required to support agent interaction. This issue is partly addressed by the nature of the proposed architecture and partly through an extendable ontology; both are described. The advantages offered by the framework are illustrated in this paper by considering a clustering application. The motivation for the latter is that no "best" clustering algorithm has been identified, and consequently an agent-based approach can be adopted to identify "best" clusters. The application serves to demonstrates the full potential of MADM.

Keywords: Multi Agent Data Mining, Agent-Based Clustering.

1 Introduction

The advantages offered by Multi-Agent Systems (MAS) with respect to distributed cooperative computing are well understood. Broadly the work presented in this paper seeks to demonstrate how the general advantages offered by MAS may be applied to data mining, i.e. Multi Agent Data Mining (or MADM). MADM can provide support to address a number of general data mining issues, such as:

1. **The size of the data sets to be mined:** Ultimately data miners wish to mine everything: text, images, video, multi-media as well as simple tabular data. Data mining techniques to mine tabular data sets are well established, however ever larger data sets, more complex data (images, video), and more sophisticated data formats (graphs, networks, trees, etc.) are required to be mined. The resources to process these data sets are significant; an MADM approach may therefore provide a solution.
2. **Data security and protection:** The legal and commercial issues associated with the security and protection of data are becoming of increasing

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 115–127, 2010.

significance in the context of data mining. The idea of sharing data for data mining by first compiling it into a single data warehouse is often not viable, or only viable if suitable preprocessing and annonimization is first undertaken. MADM provides a mechanism to support data protection.

3. **Appropriateness of Data mining Algorithms:** An interesting observation that can be drawn from the data mining research conducted to date is that for many data mining tasks (for example clustering) there is little evidence of a "best" algorithm suited to all data. Even when considering relatively straightforward tabular data, in the context of clustering, there is no single algorithm that produces the best (most representative) clusters in all cases. An agent-based process of negotiation/interaction, to agree upon the best result, seems desirable.

The vision of MADM suggested in this paper is that of a generic framework that provides the infrastructure to allow communities of data Mining Agents to collectively perform specific data mining tasks. However, although this MADM vision offers a potential solution to the above, there are a number of issues to be resolved if this vision is to be achieved:

1. **The disparate nature of data mining:** The nature of the potential data mining tasks that we might wish the envisioned MADM to perform is extensive.
2. **Organic growth:** For the MADM vision to be truly useful it must be allowed to grow "organically".

Thus the envisioned MADM must support facilities to allow simple inclusion of additional agents into the framework in an "ad hoc" manner. The communication mechanism that supports the MADM is therefore a key issue. The mechanism must support appropriate agent interaction; so that agents may undertake many different data mining tasks, and so that more and more agents can be included into the system by a variety of end users.

This paper describes an operational generic MADM framework that supports communication through means of an extendable data mining ontology. To provide a focus for the work described a clustering scenario is addressed. The motivation for the scenario is to employ a number of clustering algorithms and select the result that has produced the "best" (most cohesive) set of clusters. However, the scenario features many of the more general MADM issues identified above.

The rest of this paper is organised as follows. In Section 2 some previous work in the field of MADM, and some background to the clustrering application used to illustrate this paper, is described. The broad architecture for the MADM framework is described in Section 3. Section 4 presents a discussion of the communication framework adopted. In Section 5 the proposed MADM mechanism is illustrated and evaluated in the context of data clustering. Some conclusions are presented in Section 6.

2 Previous Work

This previous work section provides some necessary background information for the work described. The section is divided into two sub-sections. The first gives a "state-of-the-art" review of current work on MADM. The second provides background information regarding the data clustering application used to illustrate the work described in this paper.

2.1 Multi-Agent Data Mining

A number of agent-based approaches to data mining have been reported in the literature, including several directed at clustering. These include PADMA [1], PAPYRUS [2] and JABAT [3]. PADMA is used to generate hierarchical clusters in the context of document categorisation. *Local clusters* are generated at local sites, which are then used to generate *global cluster* at the central site. PAPYRUS is clustering MAS where both data and results are moved between agents according to given MAS strategies. JABAT is a MAS for both distributed and non-distributed clustering (founded on the K-means algorithm). JABAT is of note in the context of this paper because it uses ontologies to define the vocabularies and semantics for the content of message exchange among agents. None of these MAS support the concept of using an MADM approach to identify "best" clusters.

Another example of a MADM system is that of Baazaoui Zghal et al. [4] who proposed a MAS, directed at geographically dispersed data, that uses pattern mining techniques to populate a Knowledge Base (KB). This KB was then used to support decision making by end users.

There is very little reported work on generic MADM. One example is EMADS (the Extendible Multi-Agent Data Mining Framework) [5]. EMADS has been evaluated using two data mining scenarios: Association Rule Mining (ARM) and Classification. EMADS can find the best classifier providing the highest accuracy with respect to a particular data set. A disadvantage of EMADS is that fixed protocols are used, whereas the proposed MADM uses a more accessible ontlogy based approach. However, the work described here borrows some of the ideas featured in EMADS.

2.2 Data Clustering

A clustering scenario is used in this paper to illustrate the operation of the proposed generic MADM. The objective is to identify the best set of clusters represented in a given data set. The demonstration comprises three clustering agents each possessing a different clustering algorithm: (i) K-means [6], K-Nearest Neighbour (KNN) [7], and (iii) DBSCAN [8].

The K-means algorithm is a partitional clustering technique that takes a parameter K (the desired number of clusters) and then partitions a given set of data into K clusters. Cluster similarity is measured with regard to the mean value of the objects in a cluster. The K-means algorithm operates as follows: (i) selects K random points as centres, called centroids, (ii) assigns objects to the closest of centroids, based on the distance between the object and centroids, (iii) when all objects have been assigned to a cluster compute a new centroid for each cluster and repeat from step two until the clusters converge. A disadvantage of K-means is the need to specify the number of clusters in advance.

The KNN algorithm uses a threshold, t, to determine the nearest neighbour. If the distance between an item and the closet object is less than the threshold, this item should be put into the same cluster with the closest object. A new cluster is created when the distance is more than the threshold. The value of t thus significantly affects the number of clusters.

DBSCAN is a density-based clustering algorithm that generates clusters with a given minimum size ($minPts$) and density threshold (ϵ). This feature allows the algorithm to handle the "outlier problem" by ensuring individual outliers will not be include in a cluster. The overall number of clusters, K, is determined by the algorithm.

For evaluation purposes the F-Measure has been adopted, in this paper, to compare the "fitness" of clustering results. The F-Measure is popularly used in the domain of Information Retrieval [9]. The technique measures how well cluster labels match externally supplied class labels. The F-measure combines the probability that a member of a cluster belongs to a specific partition (*precision*), and the extent to which a cluster contains all objects of a specific partition (*recall*) [10]. Let $C = \{C1, C2, ..., Ck\}$ be a clustering result, $P = \{P1, P2, ..., Pk\}$ is the ground-truth partition of data set. The precision of cluster i with respect to partition j is $precision(i, j) = |Ci \cap Pj|/|Ci|$. The recall of cluster i with respect to partition j is defined as $recall(i, j) = |Ci \cap Pj|/|Pj|$. The F-measure of cluster i with respect to partition j is then defined as:

$$F_{ij} = \frac{2 \times precision(i,j) \times recall(i,j)}{precision(i,j) + recall(i,j)} \tag{1}$$

and the overall F-measure of the cluster is calculated using:

$$F = \sum_{i=1}^{m} \frac{|P_i|}{|P|} \times \max(F_{ij}) \tag{2}$$

In the context of the clustering scenario used to illustrate this paper, the "best" clustering algorithm is defined as the algorithm providing a cluster result with the highest overall F-measure in the context of a given data set.

3 MADM Framework

The proposed generic MADM framework comprises 5 basic types of agent, each agent type may have several sub-types associated with it, as follows:

1. **User Agents:** Provide a graphical user interface between the User and Task Agents.

2. **Task Agents:** Facilitate the performance of data mining tasks. Three distinct sub-types of Task Agent are identified corresponding to three high level data ming tasks: Association Rule Mining (ARM), Classification, and Clustering. Task Agents are responsible for managing and scheduling a mining request. The Task Agents identify suitable agents required to complete a data mining task through reference to a "yellow pages" service.
3. **Data Agents:** Agents that possess meta-data about a specific data set that allows the agent to access that data. There is a one-to-one relationship between Data Agents and data sets.
4. **Mining Agents:** Responsible for performing data mining activity and generating results. The result is then passed back to a Task Agent.
5. **Validation Agents:** Agents responsible for evaluating and assessing the "validity" of data mining results. Several different sub-types of Validation Agent are identified, in a similar manner to the sub-types identified for Task Agents, associated with different generic data mining tasks: Association Rule Mining (ARM), Classification, and Clustering

In some cases a specific agent may have secondary *sub-agents* associated with it to refine particular MADM operations.

The MADM framework was implemented using JADE (Java Agent Development Environment) [11], a well established, FIPA[1] compliant, agent development toolkit. JADE provides a number of additional "house keeping" agents that are utilised by the MADM framework. These include: (i) The AMS (Agent Management System) agent responsible for managing and controlling the lifecycle of other agents in the platform, and (ii) The DF (Directory Facilitator) agent that provides the "yelllow pages" service to allow agents to register their capabilities and to allow Task Agents to identify appropriate agents when scheduling data mining tasks.

Figure 1 shows an example agent configuration for the proposed generic MADM. The configuration given in the figure includes examples of the different types of agent identified above. The figure actually illustrates an agent configuration to achieve data clustering, the MADM application used for demonstration purposes in this paper (see Section 5). The "flow of control" starts with the User Agent that creates a specific Task Agent, in this case a clustering Task Agent. The created Task Agent interacts with the House Keeping Agents to determine which agents are available to complete the desired task and then selects from these agents. In the example given in Figure 1 a total of five agents are selected: three Mining Agents ($C1$, $C2$ and $C3$), a Validation Agent and a Data Agent. In the example the Task Agent communicates with the three data Mining Agents that interact with a single Data Agent (but could equally well interact with multiple Data Agents). The Mining Agents pass results to the Validating Agent, which (with the assistance of its secondary agent) processes the results and passes the final result back to the User Agent via the Task Agent. Note that the Task Agent ceases to exist when the task is complete while the other

[1] Foundation for Intelligent Physical Agents, the international association responsible for multi-agent system protocols to support agent interoperability.

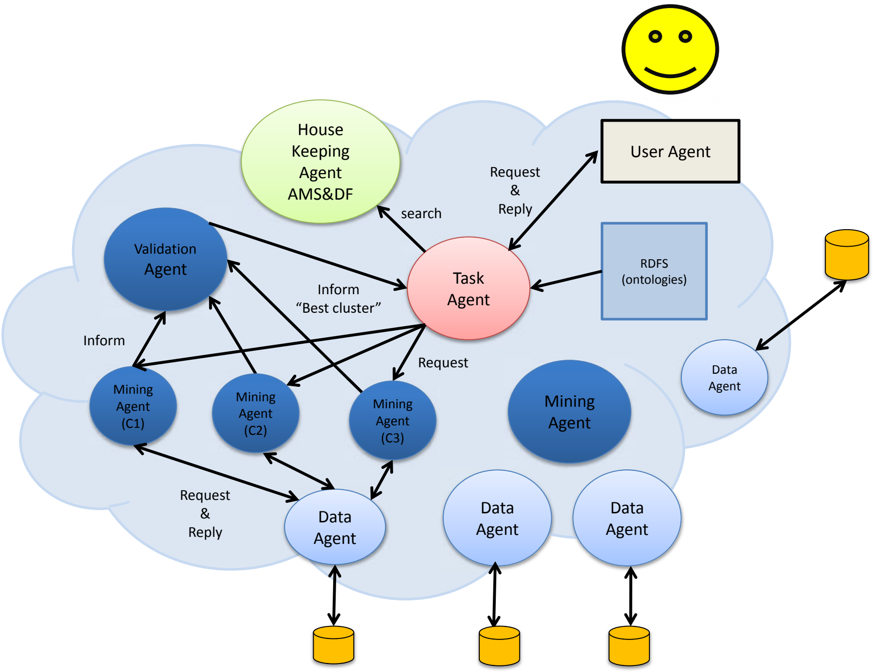


Fig. 1. Architecture of the proposed MADM system for clustering

agents persist. The interaction and communication between agents occurs using the structure, vocabularies and properties defined in the ontology (see Section 4). Figure 1 also includes a number of additional Data and Mining Agents that are not used to resolve the given task.

4 Intra Agent Communication

An essential aspect of interactions within a multi-agent system is the ability for the agents to be able to communicate effectively. The development of agent communication languages has been influenced by work from philosophy on speech act theory, most notably [12] and [13], yet there is no one definitive agent communication lanaguage appropriate to all applications. There are, however, some notable examples of popular languages that have been developed for use in multi-agent systems, with two of the most prominent proposals being: KQML (Knowledge Query and Manipulation Language) and the associated Knowledge Interchange Format (KIF) [14]; and FIPA ACL (Foundation for Intelligent Physical Agents Agent Communication Langauge) [15].

Firstly, regarding KQML and KIF, KQML is defined as the message-based "outer" language of the communication, classifying messages into particular groups, called *performatives*, to establish a common format for the exchanges.

Conversely, KIF is concerned only with providing a representation for the "inner" content of the communication, i.e. the knowledge applicable in the particular domain. Although KQML proved to be influential amongst agent developers and has formed the basis of a number of implementations, numerous criticisms have been directed at it on a number of grounds including its interoperability, lack of semantics and the omission of certain classes of messages to handle particular expressions.

The criticisms of KQML led to the development of a separate though similar agent communication language, FIPA ACL, which was aimed at establishing a standard communication language for use by autonomous agents. FIPA ACL is similar to KQML in terms of syntax; and also, like KQML, the FIPA ACL uses an outer language to enable message passing of the separate inner content, which may be expressed in any suitable logical language. For the outer language, FIPA ACL provides twenty two *performatives* to distinguish between the different kinds of messages that can be passed between agents. Examples of FIPA ACL performatives are *inform*, to pass information from one agent to another, and *request*, to ask for a particular action to be performed. The full specification of FIPA ACL performatives can be found in [15]. In order to avoid some of the criticisms that were directed against KQML the developers of FIPA ACL provided a comprehensive formal semantics for their language. These semantics made use of the work on speech act theory, as mentioned earlier, through the definition of a formal language called Semantic Language (SL). SL enables the representation of agents' beliefs, uncertain beliefs, desires, intentions and actions available for performance. To ensure that agents using the language are conforming to it, SL contains constraints (pre-conditions) in terms of formulae mapped to each ACL message that must be satisfied in order for compliance to hold e.g., agents must be sincere, and they must themselves believe the information they pass on to others. Additionally, SL enables the rational effects of actions (post-conditions) to be modelled, which state the intended effect of sending the message e.g., that one agent wishes another to believe some information passed from the first to the second.

Despite the enhancements that the FIPA ACL provided over KQML it has not escaped criticism itself and we return to consider this point later on in section 6. For now we discuss the communication mechanism used in our MADM framework. As noted above, our implementation was done using the JADE toolkit, which promotes the use of the standard FIPA ACL to facilitate the interaction among agents. Agents apply an asynchronous message passing mechanism, ACLMessage, to exchange messages through the computer infrastructure. Through the content language a particular expression is communicated from a sender to a recipient. The content expression might be encoded in several ways; JADE provides three types of content language encoding as follows:

1. **SL:** The SL (Semantic Language) content language is a human-readable string encoded content language and suitable for open-based applications where agents come from different developers, running on different platforms and have to communicate.

2. **LEAP:** The LEAP (Lightweight Extensible Agent Platform) content language is a non-human readable byte-codec content language. Therefore, LEAP is lighter than SL and adopted for agent-based applications that have a memory constraint.
3. **User Defined:** User-defined content language is consistent with the languages handled by the resources, e.g. SQL, XML, RDF, etc.

FIPA does not define a specific content language but recommends using the SL language when communicating with the AMS and DF agents. In the proposed MADM framework the SL language was adopted because the MADM framework is an open-agent based application where agents could come from different developers, running on different platforms and have to communicate.

In our MADM system the agents communicate by making and responding to requests. As noted above, the communicative exchanges proceed in accordance with the FIPA ACL. Each FIPA ACL message comprises a number of different elements. For example, consider Table 1 which shows an excerpt of the communication with respect to a particular "mining request" message sent by a Task Agent to a Mining Agent. In this example the performative being used is "request" and details are given of which agent is the sender, which is the receiver, the language being used, and the conversation id. Regarding the actual content of the messages, this is taken from the ontology, which is defined in the form of a Resource Description Framework Schema (RDFS). Within the ontology are defined the objects, atttibutes and relationships that are of concern to the task at hand and thus need to be referred to within the communication that takes place. Thus, the RDFS ontology enables a vocabulary to be defined, and the RDF message content that can be generated given this onotology provides the semantics for the content of messages exchanged among agents. As can be seen from the example in Table 1, the content of the message is expressed in RDF, as taken from the ontology, where the type of task is stated (clustering), the data set to be used is stated (userd1), and the values for *minPts* and ϵ parameters (to be used by the DBSCAN algorithm) are supplied. The Mining Agent that received this message can subsequently reply as appropriate to the Task Agent that sent the request.

Figure 2 gives details of the generic MADM ontology in the context of the data clustering application. In this framework, the instance of ontology is represented in RDF format.

5 Data Clustering Demonstration

To illustrate the operation of the proposed MADM framework a clustering application is described and evaluated in this section. The scenario was one where an end user wishes to identify the "best" set of clusters within a given data set. Three Mining Agents were provided with different clustering algorithms: (i) K-means, (ii) KNN and (iii) DBSCAN (brief overviews for each were given in Section 2).

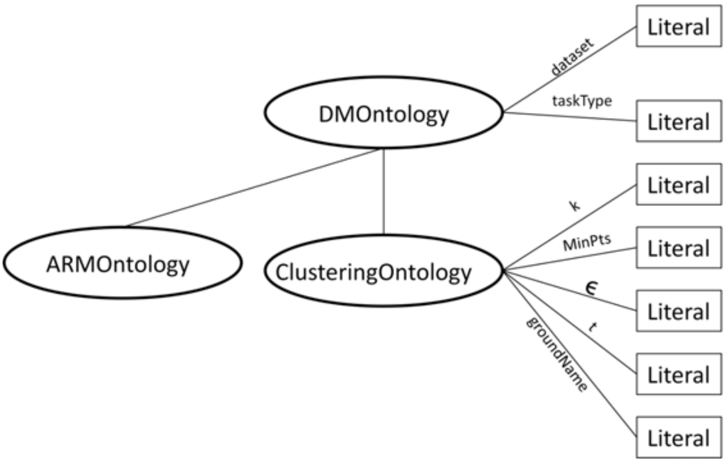


Fig. 2. MADM ontology for clustering task

The F-measure, also described in Section 2, was adopted as the criteria for comparing clusters and identifying a best clustering (however any other system of measuring the quality of clusters could equally well have been used and discussed amongst agents). A Validation Agent was provided with the ability to calculate FI measures given specific clusterings. For this purpose a further, secondary Validation Agent, which had access to an appropriate "ground truth partition" (a set of records identified, apriori, with predefined clusters), used to calculate the F-measure, was included. Recall that a cluster result providing a large overall F-Measure value is better than one with a small value.

Within the context of the proposed MADM framework the clustering scenario is resolved as follows (the reader may find it useful to refer to Figure 1). The process commences with an end user request. The user supplies details of the data set to be processed: a ground-truth partition together and the necessary parameters used for the three clustering algorithms. The parameters were as follows: the desired number of K-means clusters (k), the KNN t threshold, and the DBSCAN minimum size ($minPts$) and density (ϵ) thresholds. The User Agent then creates an appropriate Task Agent. Once the Task Agent is created it interacts with the DF agent so as to identify appropriate Mining, Data and Validation Agents. The original mining request message is then sent to each identified (clustering) Mining Agent. Data is accessed locally agent-based using the identified Data Agent which interacts with the Mining Agents. The clustering results, from the Mining Agents, are sent to the Validation Agent where (employing the F-measure and the secondary "ground-truth" Validation Agent) the best set of clusters is identified. This result is then returned to the user via the Task Agent and User Agent. On completion of the process the Task Agent is destroyed.

Table 1. Example of request message

```
(request
:sender usert1
:receiver DBSCANAgent1
:content
<rdf:RDF
  xmlns:j.0="http://protege.stanford.edu/"
  xmlns:rdf="http://www.w3.org/1999/02/22−rdf−syntax−ns#">
  <rdf:Description rdf:about="http://somewhere/ClusteringOntology">
    <j.0:Eps>5.0</j.0:Eps>
    <j.0:MinPts>3</j.0:MinPts>
    <j.0:dataset>userd1</j.0:dataset>
    <j.0:taskType>clustering</j.0:taskType>
  < /rdf:Description>
< /rdf:RDF>
:language SL
:conversation-id data mining
)
```

For evaluation purposes ten data sets taken from the UCI machine learning data repository [16] were used. Table 2 lists the results obtained using the MADM approach for these ten data sets. The table gives the size of the data set (in terms of the number of data attributes and records), the number of clusters produced and the associated F-measure for each of the three clustering algorithms used and the values of the required parameters (K, t, $minPts$ and ϵ). Note that with respect to the K-means algorithm, for the number of desired clusters to be pre-specified we have used the number of classes given in the UCI repository. The KNN algorithm makes use of the threshold, t, to determine whether items will be added to a cluster or not. The t value which provided the closest number of classes given in the UCI repository was selected in this exercise so as the give KNN the best opportunity to correctly cluster the given data. The values for the $minPts$ and ϵ parameters used by DBSCAN was determined in a similar manner to the KNN t parameter.

Table 3 gives the best result, returned to the end user in each case. The results were generated by the Validation Agent. Table 3 supports the observation that there is no single best clustering algorithm consequently supporting the motivation for the scenario. From the table it can be seen that there is no obvious link between particular clustering algorithms and the features associate with individual data sets. The only thing that can be said is that DBSCAN and K-means tend (in many cases) to outperform KNN.

The demonstration presented above indicates firstly the versatility of the MADM approach. New agents can be added to the framework and operate within it provided that they subscribe to the defined ontology. A new data mining technique, DM evaluation technique, or a data set can be shared to other users in the system by using the existing agent templates. The intention is that through

Table 2. The clustering results as produced by the MADM framework

No.	Data sets	No. Attr	No. Recs.	K-means		KNN			DBSCAN			
				Num Classes	F-Measure	t	Num Classes	F-Measure	$MinPts$	ϵ	Num Classes	F-Measure
1.	Lenses	4	24	3	4.69	1.00	1	14.40	1	1.0	4	3.70
2.	Iris Plants	4	150	3	44.26	1.00	4	29.41	1	5.0	4	32.48
3.	Zoo	18	101	7	10.72	2.00	9	8.10	2	4.0	7	11.06
4.	Wine	13	178	3	40.53	135.00	4	30.05	10	2700.0	7	8.36
5.	Heart	13	270	2	78.97	94.00	3	59.94	3	55.0	3	3.22
6.	Ecoli	7	336	8	38.39	0.45	6	35.23	1	0.4	7	44.89
7.	Blood Transfusion	4	748	2	265.36	1100.00	6	86.17	15	120.0	8	18.35
8.	Pima Indians Diabetes	8	768	2	246.84	135.00	4	128.41	10	300.0	5	4.63
9.	Yeast	8	1484	10	58.84	0.35	9	66.95	2	0.5	9	89.40
10.	Car	6	1782	4	176.70	1.45	5	195.95	2	35.0	4	226.93

Table 3. The "best" cluster result provided by the MADM framework

No.	**Data sets**	**Overall F-Measure**	**Best clustering alg.**
1	Lenses	14.40	KNN
2	Iris Plants	44.26	K-means
3	Zoo	11.06	DBSCAN
4	Wine	40.53	K-means
5	Heart	78.97	K-means
6	Ecoli	44.89	DBSCAN
7	Blood Transfusion	265.36	K-means
8	Pima Indians Diabetes	246.84	K-means
9	Yeast	89.40	DBSCAN
10	Car	226.93	DBSCAN

the adoption of the ontology the system will be allowed to grow organically. The clustering scenario also indicates how MADM can be used to select the most appropriate data mining algorithm for a particular application (clustering in the case of this paper). The privacy and security advantages, although not specifically discussed in this paper, are self evident.

6 Conclusions

In this paper a proposed generic MADM framework was described. The framework is intended to support generic multi-agent data mining by providing mechanisms for the multi-agent system to grow organically. This is facilitated partly by the architecture of the proposed MADM framework and partly by the adoption of the advocated ontology. The utility of the framework was illustrated using a data clustering scenario. The scenario demonstrated the effectiveness of the proposed MADM approach.

The data mining ontology, a segment of which was presented in this paper, is not yet complete. Indeed it can be argue that it will never be fully complete. However, the current ontology is extendible and the research team are currently working towards increasing the breadth (scope) of the ontology. There are also aspects of the communication mechanism that we intend to develop in future work. As noted in section 4, despite the enhancements that the FIPA ACL provides over earlier ACLs, a number of criticisms have been directed against it, e.g. in [17] a comprehensive analysis and critique has been given. Some of the main points of contention are that the semantics are difficult to verify, the language and rules provide little structure as there are no rules to avoid disruptive behaviour (which can be an issue in open MASs), and the ACL was intended for purchase negotiation dialogues and as such is not fully relevant for alternative types of application (such as data mining). However, a particularly noteworthy criticism of the FIPA ACL relevant for our considerations is that its argumentative capabilities are severely limited, with an under-provision of locutions to question and contest information. In future work we hope to allow our agents to engage in more expressive types of dialogue where they argue about what constitutes the 'best' cluster and why (for example, agents could argue about which clustering algorithm is the most appropriate to use in a particular scenario and why). We intend to explore this aspect in future work by making use of argumentation approaches (see [18] for an overview of this topic) that give rise to richer, more expressive forms of communication between autonomous agents.

References

1. Kargupta, H., Hamzaoglu, I., Stafford, B.: Scalable, distributed data mining using an agent based architecture. In: Proceedings the Third International Conference on the Knowledge Discovery and Data Mining, pp. 211–214. AAAI Press, Menlo Park (1997)
2. Bailey, S., Grossman, R., Sivakumar, H., Turinsky, A.: Papyrus: A system for data mining over local and wide area clusters and super-clusters. In: Proceedings of Supercomputing. IEEE, Los Alamitos (1999)
3. Czarnowski, I., Jędrzejowicz, P.: Agent-based non-distributed and distributed clustering. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition. LNCS (LNAI), vol. 5632, pp. 347–360. Springer, Heidelberg (2009)
4. Baazaoui Zghal, H., Faiz, S., Ben Ghezala, H.: A framework for data mining based multi-agent: An application to spatial data. In: Proceedings - WEC 2005: 3rd World Enformatika Conference, vol. 5, pp. 22–26 (2005)
5. Albashiri, K., Coenen, F., Leng, P.: Emads: An extendible multi-agent data miner. Knowledge-Based Systems 22(7), 523–528 (2009)
6. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
7. Dasarathy, B.V.: Nearest neighbor (NN) norms: NN pattern classification techniques. IEEE Computer Society Press, Los Alamitos (1991)

8. 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 (KDD 1996), pp. 226–231 (1996)
9. van Rijsbergen, C.: Information Retrieval, 2nd edn. Butterworths, London (1979)
10. Tan, P.N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Addison-Wesley, Reading (2005)
11. Bellifemine, F., Bergenti, F., Caire, G., Poggi, A.: Jade: a java agent development framework. In: Bordini, R.H. (ed.) Multi-agent programming: languages, platforms, and applications, p. 295. Springer, New York (2005)
12. Austin, J.L. (ed.): How to do Things with Words. Oxford University Press, Oxford (1962)
13. Searle, J.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)
14. Patil, R., Fikes, R.F., Patel-Schneider, P.F., McKay, D., Finin, T., Gruber, T., Neches, R.: The DARPA knowledge sharing effort: Progress report. In: Nebel, B., Rich, C., Swartout, W. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference, pp. 777–788. Morgan Kaufmann, USA (1992)
15. FIPA: Communicative Act Library Specification. Technical Report XC00037H, Foundation for Intelligent Physical Agents (2001), http://www.fipa.org
16. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
17. McBurney, P., Parsons, S., Wooldridge, M.: Desiderata for agent argumentation protocols. In: Castelfranchi, C., Johnson, W.L. (eds.) Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002), Bologna, Italy, pp. 402–409. ACM Press, New York (2002)
18. Bench-Capon, T., Dunne, P.E.: Argumentation in artificial intelligence. Artificial Intelligence 171(10-15), 619–641 (2007)

Describing Data with the Support Vector Shell in Distributed Environments

Peng Wang[1] and Guojun Mao[2]

[1] College of Computer Science, Beijing University of Technology,
No. 100 Pingleyuan, Chaoyang District, Beijing, China, 100124
commology@gmail.com

[2] School of Information, Central University of Finance and Economics,
No. 39 South College Road, Haidian District, Beijing, China, 100081
maoguojun@bjut.edu.cn

Abstract. Distributed data streams mining is increasingly demanded in most extensive application domains, like web traffic analysis and financial transactions. In distributed environments, it is impractical to transmit all data to one node for global model. It is reasonable to extract the essential parts of local models of subsidiary nodes, thereby integrating into the global model. In this paper we proposed an approach SVDDS to do this model integration in distributed environments. It is based on SVM theory, and trades off between the risk of the global model and the total transmission load. Our analysis and experiments show that SVDDS obviously lowers the total transmission load while the global accuracy drops comparatively little.

Keywords: Support Vector Machines, Shell, Model Integration.

1 Introduction

1.1 Background

In the present practical environment of data mining, most data are or can be regarded as data streams. With the rapid development of Internet, it is increasingly required to process the massive input data in real-time. Such cases exist extensively in most application domains (like web traffic analysis, intrusion detection, financial transactions analysis, etc.). Data stream mining has gained many intensive studies [1-3], including adapting most traditional algorithms to single data stream mining [4, 5], as well as mining in distributed environments [6, 7]. However, few studies focus on integrating local models into global model.

A *(single) data stream* is a data sequence that is *temporally ordered*, *continuous*, *massive*, *potentially infinite*, and *fast changing with varying update rates* [8], of which each item is consistently defined beforehand. In contrast with traditional static, finite data sets, accompanied by their corresponding mining algorithms, data stream consume exponentially more time and space to process. Despite modern hardware that has especially powerful computation speed and storage capacity, it is still impossible or impractical to store data streams for future offline analysis.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 128–142, 2010.

Due to the above characteristics of data stream, online algorithm is the only practical way of processing data streams. The term "online" does not only denote processing data stream in real-time, instead of putting them away into storage systems of massive capacity, but also indicates processing data stream by single-pass (or single-scan), in incremental manner.

A group of parallel single data streams comprise *multiple data streams*. All data streams can be synchronous or asynchronous. They can be merged (conditionally) into one single data stream.

The multiple data streams in distributed system are *distributed data streams*. They are correlated and thus must be analyzed globally. Similar to the time-series theory, it is helpful to introduce *auto-correlation* and *cross-correlation*. Auto-correlation describes the correlation between subsequences of same data stream at different phases. Cross-correlation describes the correlation between different data streams of same distributed system. In distributed environments, the parallel data streams may have similar patterns, but the patterns may arrive at different time. The repeated patterns of the same data stream can be discovered by auto-correlation, and the similar patterns between some subsequences of different data streams at same or different time discovered by auto-correlation.

Centralized and decentralized architectures are typical architectures of distributed system. In this paper, *centralized architecture* refers to the multi-level topology with one or more centers, *decentralized architecture* the topology with no centers, like ring network.

The goal of mining distributed data streams is to discover the global model among the distributed streams: they are not independent ones, but related. Discovering the global model requires integrating distributed models, rather than simply merging them. However, because those data streams and their models are distributed, integrating them, or even transmitting them is rather expensive.

1.2 SVM Algorithms

Support Vector Machines (SVMs) and related kernel methods are well-performed algorithms optimizing generalization performance. SVM theory is based on statistical learning theory and *structural risk minimization (SRM)* thereof. It optimizes the tradeoff between empirical risk of generalization that only replies on subset of input data and the complexity of approximating function (structural risk of the machine, such as classification, regression, etc.), and targets *the function that for the fixed amount (essential subset) of data achieves the minimum of the guaranteed risk*. [9]

The essential subset of data, named *support vectors (SVs)*, is determined by actual distribution, but with least priori knowledge of the distribution. Other data than support vectors do not affect the final decision hyperplane.

1.2.1 SVM Preliminary

The goal of SVM is to maximize the margin around the decision hyperplane between two data sets for maximal generalization performance. This optimization is a quadratic programming (QP) problem. The classical SVM model, named *soft-margin method*, can be applied to both linearly separable and inseparable cases (in mapped RKHS space).

For a given kernel mapping $\Phi(\cdot)$ from input space into a dot product feature space, in the feature space the dot product of images can be computed by $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ satisfying

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j) \tag{1}$$

This (*kernel trick*) makes the computation of dot product in feature space available without ever explicitly knowing the mapping. See [10, 11] for more details.

The optimization model is tunable via presetting different regularization constant C, which determines the tradeoff between the structural risk and the empirical risk. By setting larger C, the empirical errors (ξ) get more punished i.e. less data are allowed to be in the margin; and vice versa.

The outline of particular data set varies in different contexts. To adapt to a specific case, certain suitable kernel function has to be chosen in advance, usually from *polynomial functions*, *radial basis functions (RBF)*, *hyperbolic tangent functions*.

In geometric viewpoint [12-14], SVM (including classification, outlier detection, regression, etc.) can be regarded as convex hull problem, in which every class of the data corresponds to a convex hull. In particular, for SVM classification, maximizing the margin is equivalent to finding the maximum margin between the closest points between the convex hulls (or reduced convex hulls, RCH, for inseparable cases). “SVs lie on the boundary of the convex hulls of the two classes, thus they possess supporting hyperplanes. The SV optimal hyperplane is the hyperplane which lies in the middle of the two parallel supporting hyperplanes (of the two classes) with maximum distance.” [10].

1.2.2 SVDD Preliminary

By applying the same methodology of SVM to a data set without labels, support vector data description (SVDD) [15] algorithm provides an efficient approach to outlier or novelty detection, which is named one-class classification [16].

The decision hyperplane between the outliers and the normal data outlines the boundary of the data set. Through kernel mapping, every data set can be transformed to a hypersphere-like shape in feature space, whose boundary is a hypersphere.

In geometric viewpoint, every class is contained in a convex hull, which corresponds to the boundary hypersphere produced by SVDD. SVDD actually finds the smallest enclosing hypersphere [10, 11, 16].

1.3 Integrating SVMs in Distributed Environments

The SVM family provides an approach to reducing the massive input data to a fairly small subset – the support vectors. The decision hyperplane can be recovered from the support vectors without any losses, provided that the settings (kernel function and parameter values) are identical. As those essential data account for a small portion of the model, and in turn the labeled data (which need learning) a small portion of the input data, SVM family is suitable for mining data stream.

Since the decision hyperplane is determined by SVs but independent of non-SVs, it is reasonable to integrate different models by their SVs. That is, for each SVM, the SVs accompanied with the respective settings comprise the SVM model, whereas the decision hyperplane is optional. For example, given two SVM 1 and 2 (**Fig. 1**) in same 2-dimensional space, which classify respective data of same two classes, respectively marked by “X” and “+”, **Fig. 2** shows the decision hyperplane and the support vectors trained by batch SVM based on the combined data of SVM 1 and SVM 2; **Fig. 3** shows two possible decision hyperplanes trained by the batch SVM only based on the combined SVs of SVM 1 and 2. Note that the points with circles and squares are not the new support vectors trained by the batch SVM, but the original support vectors of SVM 1 and 2. Other data points are shown in order to check the performance of the classification only based on the combined SVs of SVM 1 and 2. It can be seen that the classification performance of both classifiers are well, compared to that of **Fig. 2**.

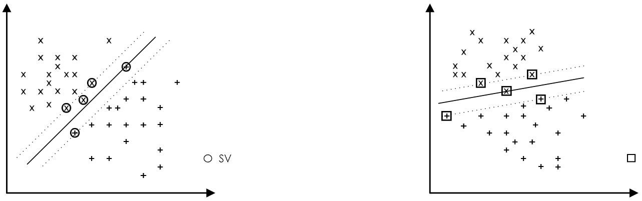


Fig. 1. SVM 1 and 2

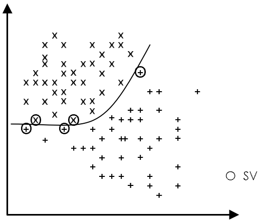


Fig. 2. The decision hyperplane and the support vectors trained by the batch SVM based on the combined data of SVM 1 and SVM 2. The solid curve shows the decision hyperplane and the points with circles the support vectors.

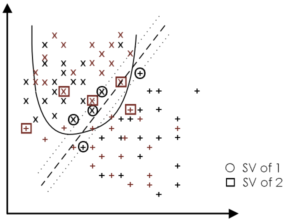


Fig. 3. Two possible decision hyperplanes trained by the batch SVM only based on the combined SVs of SVM 1 and 2. For a linear classifier, the solid line shows the decision hyperplane and the dotted line the margin. For a non-linear classifier, the solid curve shows the decision hyperplane.

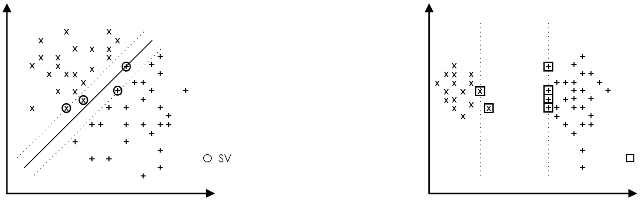


Fig. 4. SVM 3 and 4

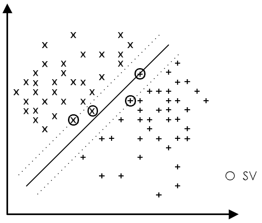


Fig. 5. The decision hyperplane and the support vectors trained by the batch SVM based on the combined data of SVM 3 and SVM 4. The solid line shows the decision hyperplane, the dotted line the margin, and the points with circles the support vectors.

Nevertheless, the support vectors are the vulnerable points for SVMs, because any loss or even any unintentional deviation of the support vectors (e.g. transmission errors) will damage the final decision hyperplane. Furthermore, different settings of kernel function and parameter values, such as decreasing the value of *C*, can cause more data points of non-SVs to enter into the margin (join SVs), which in turn ruins the independency of non-SVs.

For model integration, the SVs of different models are quite limited and at risk. For example, one of the risky cases is as follows. Given two SVM 3 and 4 (**Fig. 4**) similar to SVM 1 and 2, **Fig. 5** shows the decision hyperplane and the support vectors trained by batch SVM based on the combined data of SVM 3 and SVM 4; shows the decision hyperplane trained by the batch SVM only based on the combined SVs of SVM 3 and 4. Note that the points with circles and squares are not the new support vectors trained by the batch SVM, but the original support vectors of SVM 3 and 4. It can be seen that the decision hyperplane may be almost perpendicular to the decision hyperplane (the dashed line) of **Fig. 5**.

Moreover, for model integration in the global scope, without the combined original data, the few SVs (like **Fig. 6**) are so sparse that no non-SVs can support the wider margin, which means the settings would be forced to be more limited in the global model, so that the result would be meaningless. In geometric viewpoint, some probable reduced convex hulls inside are discarded without foresight.

It even gets worse when integrating different models to a global model in distributed environments. It would be very risky to submit the mere support vectors of local model to the central node. The data of different nodes will probably have different

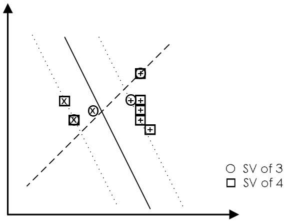


Fig. 6. The decision hyperplane trained by the batch SVM only based on the combined SVs of SVM 3 and 4. The solid line shows the decision hyperplane, the dotted line the margin, and the dashed line the hyperplane of **Fig. 5**.

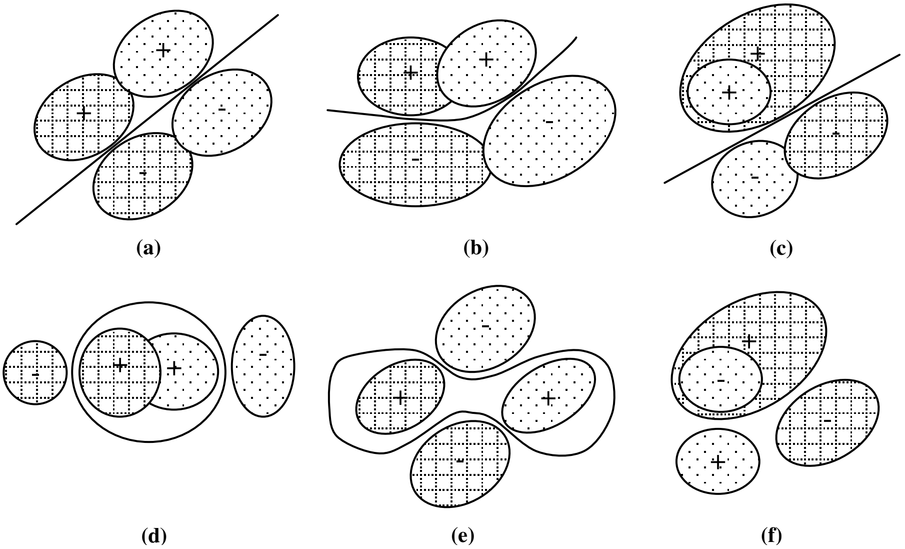


Fig. 7. Some cases of model integration of two SVMs. The areas with “+” indicate the positive data, and the areas with “-” the negative data. The dotted pair of areas is of SVM A, and the gridded pair of SVM B.

distributions, and different from the global distribution. For instance, **Fig. 7** shows several cases of model integrations. The difference between global hyperplane and the local hyperplanes suggests that the local hyperplanes are useless for the global model. Note that the chart (f) of **Fig. 7** shows a case of contrary overlap.

The support vectors of both classes produced by a SVM classifier lie in the margin. However, because it is not possible to predict where the support vectors of other classes may lie, like chart (d), it is necessary to assume that the support vectors around the data of one class must get equal treatment.

The support vectors of every class produced by a SVDD surround the data. Even though SVDD is originally designed to detect outliers, its produced SVs describe the data very well: they "shelter" the non-SVs inside. The description is omni-directional such that it can adapt to the SVM anywhere around in the global scope.

In distributed environments, the transmission load is one of the crucial factors. It is impractical to transmit every local data to the central node, since it makes no significant difference from the centralized architecture. The mere support vectors, on the other hand, is at risk and harmful to the model integration. It is necessary to trade off between them.

2 SVDD Shell Algorithm

2.1 Basic Intention

For model integration in distributed environments, it is critical to trade off between the risk of sparse support vectors produced by SVDD and the transmission load.

The mere sparse support vectors are inadequate to support the margin (or shelter the non-SVs), so the non-SVs inside are necessary for SVM model integration. SVDD is not completely independent of SVM. They work out the support vectors following the same structural risk minimization (SRM) principle as well as the same regularization methodology.

In geometric viewpoint, the convex hulls (including the reduced convex hulls, RCH) establish the relation between SVM classification and SVDD. The support vectors of SVDD are major candidates for classifications, but they are incomplete for the lack of the potential convex hulls inside.

In addition to the boundary made up of the support vectors (the outermost convex hull), the layer widened inward is reasonably appropriate. This shell-like widened boundary layer not only provides the fundamental SVs and the potential non-SVs, but also avoids transmitting the entire data set.

2.2 Model

The target is to minimize the overall risk of both describing and transmitting the data: the tight boundary and the transmission load. The tight boundary for every class is obtained by SVDD, while the transmission load relies on the amount of non-SVs outside the widened boundary. The shell-like widened boundary layer for one class (abbreviated to shell), analogous to SVDD, can be viewed in 2-dimensions, as illustrated in **Fig. 8**. Although strictly speaking the convex hulls are not certainly concentric, assuming them concentric can simplify the model, according to SRM principle. This model can be extended to higher dimensions. Because in distributed environments the non-SVs in the hollow largely increase the transmission load (in fact, the outliers detected by SVDD account for the rest) but provide less information, the hollow should be maximized.

Given a data set $\{\boldsymbol{x}_i\}$ in N-dimensional input space $\mathbb{R}^N$, and a kernel function $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the target is to minimize both the outer boundary (according to SVDD) and the thickness of the shell δ. According to trigonometry, for particular R, $\varepsilon = R^2 - r^2 = \delta\ (2R - \delta)$ is determined. Therefore, the model of the shell (primal form) is as follows:

$$\text{minimize } \Omega(R,\varepsilon,\boldsymbol{\phi};\xi,\zeta) = R^2 + \varepsilon + C_R\sum_{i=1}^{N}\xi_i + C_r\sum_{i=1}^{N}\zeta_i \tag{2}$$

$$s.t.\begin{cases} \|\boldsymbol{x}_i - \boldsymbol{\phi}\|^2 \le R^2 + \xi_i \\ \|\boldsymbol{x}_i - \boldsymbol{\phi}\|^2 \ge R^2 - \varepsilon - \zeta_i \\ 0 \le \varepsilon \le R^2 \\ \xi_i \ge 0, \zeta_i \ge 0, \quad i = 1,\ldots,N \end{cases} \tag{3}$$

where Ω is the target function (overall risk), C_R and C_r is the regularization constants corresponding to the outer and inner boundaries, and $\boldsymbol{\xi}$ and $\boldsymbol{\zeta}$ are the slack variables, respectively. The slack variable ξ_i reflects the state of $\boldsymbol{x}_i$: for support vectors, $\xi_i > 0$, which indicates its distance to the center is greater than R – a contribution to the overall risk; otherwise $\xi_i = 0$. The slack variable ζ_i reflects the state of $\boldsymbol{x}_i$: for support vectors, $\zeta_i > 0$, which indicates its distance to the center is less than r ($r^2 = R^2 - \varepsilon$) – a contribution to the overall risk; otherwise $\zeta_i = 0$.

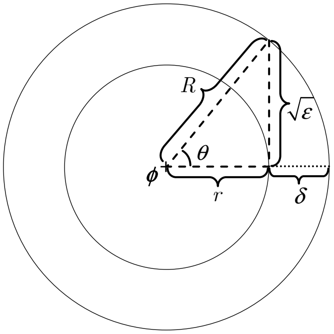


Fig. 8. The shell-like widened boundary layer (abbrev. to shell), analogous to SVDD, can be viewed in 2-dimensions. The shell consists of two concentric circles, where ϕ is the center, R is the outer radius, r the inner radius, δ the thickness, and from trigonometry, $\varepsilon = R^2 - r^2$, $\cos\theta = r / R$.

Setting derivatives of the Lagrangian to zero, respectively, and after substitution, the result is shown as the following (Wolfe) dual form:

$$\begin{aligned} &\text{maximize } L_D(\boldsymbol{\alpha}) = \sum_{i=1}^{N}(\alpha_i - \beta_i)\|\boldsymbol{x}_i - \boldsymbol{\phi}\|^2 \\ &= \sum_{i=1}^{N}(\alpha_i - \beta_i)K(\boldsymbol{x}_i,\boldsymbol{x}_i) - \frac{1}{\sum_{i=1}^{N}(\alpha_i - \beta_i)}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \beta_i)(\alpha_j - \beta_j)K(\boldsymbol{x}_i,\boldsymbol{x}_j) \end{aligned} \tag{4}$$

$$s.t.\begin{cases}\sum_{i=1}^{N}(\alpha_i-\beta_i)\le 1\\ \sum_{i=1}^{N}\alpha_i\le 2\\ 0\le\alpha_i\le C_R\\ 0\le\beta_i\le C_r\\ i=1,\ldots,N\end{cases} \tag{5}$$

Then analogous to SVDD, α_i and β_i respectively correspond to outer and inner hyperspheres. The data $\boldsymbol{x}_i$ with $\alpha_i > 0$ are *outer support vectors*. Specifically, those with $0 < \alpha_i < C_R$ [$\xi_i = 0$] are *outer margin support vectors*, which are on the outer boundary ($\|\boldsymbol{x}_k-\boldsymbol{\phi}\|^2=R^2$); those with $\alpha_i = C_R$ [$\xi_i > 0$] *outer error support vectors*, which are outside the outer boundary ($\|\boldsymbol{x}_k-\boldsymbol{\phi}\|^2>R^2$), named *outliers*. The data $\boldsymbol{x}_i$ with $\beta_i > 0$ are *inner support vectors*. Specifically, those with $0 < \beta_i < C_r$ [$\zeta_i = 0$] are *inner margin support vectors*, which are on the inner boundary ($\|\boldsymbol{x}_i-\boldsymbol{\phi}\|^2=r^2$); those with $\beta_i = C_r$ [$\zeta_i > 0$] *inner error support vectors*, which are inside the inner boundary ($\|\boldsymbol{x}_i-\boldsymbol{\phi}\|^2<r^2$), named *inliers*. The non-SVs contained in the shell, between the outer boundary and the inner boundary ($r^2<\|\boldsymbol{x}_k-\boldsymbol{\phi}\|^2<R^2$) are candidates for model integration, named *candidate support vectors*. Since the outer and the inner margin support vectors are as well suitable for model integration, they and the candidate support vectors are named *extended candidate support vectors*.

(4) is not a quadratic programming model, due to the denominator. However, it can be simplified to a QP model if the denominator is set to a constant. The KKT conditions states that the denominator equals to 1 if $R^2 > \varepsilon$, which implies that the hollow in the shell can not be empty. This excludes only the trivial case that the shell degrades to SVDD.

So, by assuming $R^2 > \varepsilon$, the QP model has the following (Wolf) dual form:

$$\text{maximize } L'_D(\boldsymbol{\alpha})=\sum_{i=1}^{N}(\alpha_i-\beta_i)K(\boldsymbol{x}_i,\boldsymbol{x}_i) \\ -\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i-\beta_i)(\alpha_j-\beta_j)K(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{6}$$

$$s.t.\begin{cases}\sum_{i=1}^{N}(\alpha_i-\beta_i)=1\\ \sum_{i=1}^{N}\alpha_i\le 2\\ 0\le\alpha_i\le C_R\\ 0\le\beta_i\le C_r\\ i=1,\ldots,N\end{cases} \tag{7}$$

The shell can be recovered by

$$\boldsymbol{\phi} = \sum_{i=1}^{N} (\alpha_i - \beta_i)\boldsymbol{x}_i \tag{8}$$

$$R^2 = \left\| \boldsymbol{x}_p - \boldsymbol{\phi} \right\|^2 \tag{9}$$

$$r^2 = \left\| \boldsymbol{x}_q - \boldsymbol{\phi} \right\|^2 \tag{10}$$

where $\boldsymbol{x}_p$ and $\boldsymbol{x}_q$ are any support vectors on the outer and inner boundaries, respectively.

The name "inlier" does not denote the negative examples considered by [16], which are referred to as the "objects which should be rejected" and "should be outside" the hypersphere. Albeit the mathematical model and the derived expressions are in the same form, they have completely different purposes and meanings. The algorithm in this paper attempts to find out those "positive" examples (in terms of SVDD) inside the SVDD boundary that increase the transmission load instead of errors. The inner support vectors in this paper are the "positive" examples of the same class, whereas in SVDD the negative examples are of other classes (so they must have labels y_i and can be generalized with $\alpha_i' = y_i\alpha_i$). The concept of inlier here and the concept of the negative examples can work together without conflicts. The intention of the name "inlier" here is to contrast the inner error support vectors with the outer error support vectors, or outliers.

2.3 Analysis

According to the principle of SVDD, SVDD shell (or briefly SVDDS) model is able as well to be independent of the shape of the data [16]. As suggested by SVDD, the Gaussian radial basis function (GRBF) kernel enables the shell to be independent of the position of the data set with respect to the origin [15]. In the above kernel function, σ determines the width (or extent) of each support vector. The smaller σ is, the more SVs are (respectively for outer and inner), the tighter the boundaries are; and vice versa. In the following, the kernel is assumed to be the Gaussian RBF as above.

The regularization constants C_R and C_r respectively determine the tradeoffs between the structural risk (first two terms in (2)) and the empirical errors of the outer support vectors as well as the inner support vectors. By setting the C_R larger, less data are allowed to be outside the outer boundary, and vice versa; by setting the C_r larger, less data are allowed to be inside the inner boundary (in the hollow), and vice versa. Usually, to obtain significant effect of the hollow, the C_r is set relatively small, as 0.02; while the C_R can be set relatively large, as 0.4.

The shell can adapt to both the outer and inner shapes and be independent of the center. The outliers are also detected as SVDD. The inliers account for a considerable proportion, and their amount can be controlled by adjusting parameters. The shell becomes thick so that the shell contains adequate extended candidate support vectors for model integration with other models, while trading off the transmission load. Inversely, to reduce the transmission load, the shell is forced to be thin and hence the risk of integration gets high.

The shell actually excludes those high density areas, in contrast to, in SVDD, only the low density areas. The data in the high density areas are regarded as inliers, while those in the low density areas are regarded as outliers. The high density areas are marked out by the inner boundary. The parameters C_R and C_r respectively determine the thresholds among high, medium, and low.

3 Implementation Issues

SVDDS is primarily applied to global model integration in distributed environments. As discussed in the background section, centralized and decentralized are the typical distributed architectures.

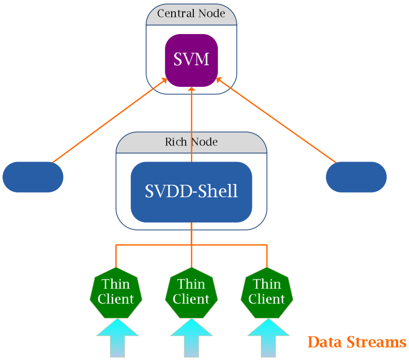


Fig. 9. Centralized architecture of model integration deployed with SVDDS

The centralized architecture can have two or more levels. Here, each level can be the following types: Rich node analyzes the data collected by subsidiary thin clients; after receiving the data, they try to discover local models using SVDDS, and then submit to central node for model integration. Central node analyzes the data collected by subsidiary rich nodes, integrating the separate subsidiary models into global model. It may play the role of rich node, relative to even upper level central node, for further integration, as illustrated in **Fig. 9**.

The decentralized architecture does not have levels, but rather is composed of rich nodes as peers. The rich nodes are identical to rich nodes of centralized architecture except that they have no upper level central nodes. To have the global model, the peers duplicate their local models to others, so that every peer respectively conducts model integration. This architecture does not depend on central nodes, but distributes the work load among the peers, so it is easier to integrate further.

SVDDS is deployed on rich node, which has to duplicate its local model for model integration. In the nodes to conduct model integration (e.g. central node), the received data are of the subsidiary models and regarded as normal input to the global working SVM, as well as a global SVDDS if deployed for further integration. Note that if no further integration is needed the global SVDDS is unnecessary. The working SVM is neither aware of the subsidiary nodes nor modified to adapt to the distributed environment.

SVDDS is only deployed on rich node that needs to duplicate local models for model integration. In star network, as illustrated in **Fig. 9**, the central node need not deploy an instance of SVDDS, since no further integration is required. In decentralized architecture, the shell is also deployed on every peer for their respective model integrations.

4 Experiments

In experiments, the SVDDS algorithm is implemented in R [17] language and based on library *kernlab* [18]. The library provides several SVM algorithms (classification, novelty detection, and regression, etc.), as well as *ipop* for solving common QP problems. The SVM classification of kernlab is used as the fundamental working SVM, and the SVDDS shell on each subsidiary node is generated by ipop solution. The Gaussian RBF kernel is only applied. The parameters are not the best, but for comparison and fairly good performance, such as C for classification is fixed to 10, C_R for SVDDS to 0.9, .The accuracy is based on 10-fold cross validation. The total transmission load is the summation of the numbers of every SVDDS extended candidate support vectors, which means the duplicate data points among different nodes are regarded separately, not as one data point. The corresponding total transmission load ratio is with respect to the training set (9/10, since 10-fold cross validation). The data is equally assigned to every node.

The experiments are made upon some data sets of UCI machine learning repository [19]. The experiments and their results are briefly shown in **Table 1**. In the experiments, effects of typical settings variations are shown. The "sub nodes" column shows the numbers of subsidiary nodes. The "training set size" column shows the numbers of training set size by 10-fold cross validation. The "parameters" column shows the settings of variable parameters, where σ_{SVC} is the σ of Gaussian RBF kernel function for global classifier, σ_{SVDDS} the σ of Gaussian RBF kernel function for SVDDS of subsidiary nodes, C_r is the regularization parameter of the inner boundary. The "SVC accuracy wholly" column shows the 10-fold cross validation accuracy upon the whole dataset (batch training). The "SVC accuracy with SVDDS nodes" column shows the 10-fold cross validation accuracy upon the combined data of the SVDDS shells of subsidiary nodes. The "Total Trans. Load # (%)" column shows total transmission load and the ratio with respect to training set. The "Trans Load per Node" column shows the average load for every node.

The experiments 3, 4 and 5 show the effect of different numbers of subsidiary nodes on the same other context. Their comparison shows that with more subsidiary nodes the accuracy does not drop dramatically, but the total transmission load decreases well. However, the number of subsidiary nodes can not increase at will, because the data assigned to each node would be so small that no adequate data are for SVDDS.

Table 1. Experiments results

ID	Dataset	Sub Nodes	Training Set Size	Parameters (Gaussian RBF)	SVC Accuracy wholly	SVC Accuracy with SVDDS nodes	Total Trans. Load # (%)	Trans. Load per Node
1	Wine	2	160	$C_r = 0.075$ $\sigma_{\text{SVC}} = 10$ $\sigma_{\text{SVDDS}} = 10$	98.33%	95.00%	87.1 (54.44%)	43.6
2				$C_r = 0.06$ $\sigma_{\text{SVC}} = 10$ $\sigma_{\text{SVDDS}} = 10$		95.56%	70.7 (44.19%)	35.4
3	Wisconsin Diagnostic Breast Cancer	2	512	$C_r = 0.06$ $\sigma_{\text{SVC}} = 15$ $\sigma_{\text{SVDDS}} = 15$	97.72%	97.19%	451.6 (88.20%)	225.8
4		5				95.96%	356.7 (69.67%)	71.3
5		8				95.96%	269.8 (52.70%)	33.7
6	Musk (V2)	13	5938	$C_r = 0.005$ $\sigma_{\text{SVC}} = 10$ $\sigma_{\text{SVDDS}} = 15$	99.42%	89.80%	3369.5 (56.74%)	259.2
7				$C_r = 0.005$ $\sigma_{\text{SVC}} = 10$ $\sigma_{\text{SVDDS}} = 15$		98.29%	4229.6 (71.23%)	325.4
8	Nursery	12	11664	$C_r = 0.01$ $\sigma_{\text{SVC}} = 5$ $\sigma_{\text{SVDDS}} = 10$	93.83%	87.21%	8088.0 (69.34%)	674.0
9				$C_r = 0.005$ $\sigma_{\text{SVC}} = 5$ $\sigma_{\text{SVDDS}} = 10$		83.09%	4477.9 (38.39%)	373.2
10	Covertype	50	10000	$C_r = 0.015$ $\sigma_{\text{SVC}} = 5$ $\sigma_{\text{SVDDS}} = 10$	74.61%	67.91%	4018.7 (40.19%)	80.4

The experiment 6 is made on uniformly sampled data from the whole dataset, while the experiment 7 is made on exclusive subspaces, i.e. if one data is assigned to a node, it will not be assigned to other nodes. The results show that the concentrated (like experiment 7) case is less risky, since its data for SVDDS is more concentrated.

The comparison between experiments 8 and 9 shows the effect of the regularization parameter C_r of the inner boundary. As C_r doubles (possibly intending to raise the accuracy), the accuracy does not rise obviously, but the total transmission load nearly doubles.

The experiment 10 shows the performance of SVDDS upon uniformly sampled data. Since the total Covertype dataset is very huge, uniformly sampled data can be regarded as actual environment. For every run of this experiment, 5 times individual sampling are made. For every sampling, only 10000 data are sampled, 200 data for each node. The parameter C_r indicates that about less than 67 inliers are allowed, so the size of the shell is about 133. It can be seen that with so many subsidiary nodes, the accuracy does not drop dramatically, but the total transmission load decreases quite well.

5 Conclusion

In this paper we proposed an approach SVDDS to integrate the models of subsidiary nodes in distributed data stream mining into a global model. It is based on SVM theory, and trades off between the risk of the global model and the total transmission load. In distributed environments, it is impractical to transmit all data (assembly) to the global node for integration. Consequently, our proposed SVDDS is suitable for the above model integration in distributed environments. Our analysis and experiments show that accuracy with SVDDS does not drop dramatically, while obviously lowers the total transmission load. Resulting from the SVM theory, the performance of SVDDS is controllable and so it can be applied to a wide range of practical problems. We will further work on incremental SVDDS algorithm based on C&P [20] series of incremental SVM algorithms.

References

1. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and Issues in Data Stream Systems. In: Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. ACM, Madison (2002)
2. Domingos, P., Hulten, G.: Mining High-Speed Data Streams. In: Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, Boston (2000)
3. Street, W.N., Kim, Y.: A Streaming Ensemble Algorithm (SEA) for Large-Scale Classification. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 377–382. ACM, San Francisco (2001)
4. Syed, N.A., Liu, H., Sung, K.K.: Handling Concept Drifts in Incremental Learning with Support Vector Machines. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 317–321. ACM, New York (1999)
5. Guha, S., Meyerson, A., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering Data Streams: Theory and Practice. IEEE Transactions on Knowledge and Data Engineering 15, 515–528 (2003)
6. Chen, L., Reddy, K., Agrawal, G.: GATES: A Grid-Based Middleware for Distributed Processing of Data Streams. In: Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing, pp. 192–201. IEEE, Honolulu (2004)
7. Beringer, J., Hüllermeier, E.: Online Clustering of Parallel Data Streams. Data & Knowledge Engineering 58, 180–204 (2006)
8. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Elsevier, Amsterdam (2006)
9. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, Inc., Chichester (1998)

10. Schölkopf, B., Smola, A.J.: Learning with Kernels - Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)
11. Shawe-Taylor, J., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, New York (2004)
12. Bennett, K.P., Bredensteiner, E.J.: Duality and Geometry in SVM Classifiers. In: Proceedings of the 17th International Conference on Machine Learning, pp. 57–64. Morgan Kaufmann Publishers Inc., Standord (2000)
13. Mavroforakis, M.E., Theodoridis, S.: A Geometric Approach to Support Vector Machine (SVM) Classification. IEEE Transactions on Neural Networks 17, 671–682 (2006)
14. Crisp, D.J., Burges, C.J.C.: A Geometric Interpretation of v-SVM Classifiers. Advances in Neural Information Processing Systems 12, 244–251 (1999)
15. Tax, D.M.J., Duin, R.P.W.: Support Vector Data Description. Machine Learning 54, 45–66 (2004)
16. Tax, D.M.J.: One-Class Classification. Vol. Doctor. Delft University of Technology, p. 198 (2001)
17. R Development Core Team: R: A Language and Environment for Statistical Computing (2008)
18. Karatzoglou, A., Smola, A., Hornik, K., Zeileis, A.: kernlab – An S4 Package for Kernel Methods in R. Journal of Statistical 11, 1–20 (2004)
19. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine (2007)
20. Cauwenberghs, G., Poggio, T.: Incremental and Decremental Support Vector Machine Learning. In: Proceedings of the 14th Conference on Neural Information Processing Systems, pp. 409–415. MIT Press, Cambridge (2000)

Robust Clustering Using Discriminant Analysis

Vasudha Bhatnagar and Sangeeta Ahuja

[1] Department of Computer Science,
University of Delhi, India
vbhatnagar@cs.du.ac.in
[2] IASRI, New Delhi, India
sangeeta@iasri.res.in

Abstract. Cluster ensemble technique has attracted serious attention in the area of unsupervised learning. It aims at improving robustness and quality of clustering scheme, particularly in scenarios where either randomization or sampling is the part of the clustering algorithm.

In this paper, we address the problem of instability and non robustness in K-means clusterings. These problems arise naturally because of random seed selection by the algorithm, order sensitivity of the algorithm and presence of noise and outliers in data. We propose a cluster ensemble method based on Discriminant Analysis to obtain robust clustering using K-means clusterer. The proposed algorithm operates in three phases. The first phase is preparatory in which multiple clustering schemes generated and the cluster correspondence is obtained. The second phase uses discriminant analysis and constructs a label matrix. In the final stage, consensus partition is generated and noise, if any, is segregated. Experimental analysis using standard public data sets provides strong empirical evidence of the high quality of resultant clustering scheme.

Keywords: K-means, Cluster Ensemble, Discriminant Analysis.

1 Introduction

Obtaining high quality clustering results is a challenging task because of several reasons including randomization inherent in the algorithm [1], sampling (to improve scalability) [2] and idiosyncracies of clustering algorithms [1]. In such situations, different solutions may appear equally acceptable in absence of apriori knowledge of the underlying data distribution [1]. Unfortunately in most real life applications data do not follow *nice* distributions documented in literature. Hence inherent assumptions (idiosyncracies) of the algorithm are often violated producing results that are far from reality, leading to errorenous decisions. Thus the choice of right clustering algorithm, which will reveal natural structures in the data is a difficult task.

Clustering ensemble technique aims to improve the clustering scheme by intelligently combining multiple schemes to yield a robust and stable clustering [1, 3, 4, 5, 2, 6, 7]. The technique has been recognized as an important method of information fusion as it improves robustness, stability and accuracy of the

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 143–157, 2010.

unsupervised learning methods. The technique is naturally amenable for parallelization and application in distributed environment [1]. Combining multiple partitions is the core task in cluster ensemble problem, which is accomplished by design of consensus function Γ.

K-means is one of the most common clustering algorithm used for data analysis in statistics [8], machine learning [9] and data mining [10]. The algorithm, proposed by MacQueen [11], is a center based clustering algorithm which iteratively partitions data till the specified quality criterion (minimum mean squared distances of each data point from centroids) is met. The popularity of the algorithm hinges on its simplicity and efficiency. After more than fifty years of extensive usage for data analysis in the fields of social, physical and biological sciences, it still interests data mining and machine learning community[1]. Bock [8] presents a historical view of K-means algorithm showing the importance and usefulness of the approach.

Interestingly, there are several known limitations of K-means algorithm. Random seed selection ensures that multiple execution of the algorithm on the same data set results into clustering schemes, which may sometimes be significantly different. Consequently user is confronted with the problem of scheme selection, since two different schemes may assign the same object in two different clusters with different properties. Thus there is a possibility of making a wrong decision if the selected scheme does not represent true structures in data. Sensitivity of the algorithm to the order in which the data is presented also contributes to the instability of the algorithm [10]. Presence of noise and outliers in data is a well known and understood cause of non-robustness of K-means clustering algorithm. Since it is not guaranteed to achieve global minimum, the number of iterations for convergence may be very large. Specification of the number of iterations by the user may result into variation of results.

In order to overcome the known weakness causing in stability and consequent non-robustness, a series of extensions and generalizations of K-means algorithm have been proposed [8]. Kanungo et al. [12] propose an effective implementation of K-means which uses a pre-computed kd-tree like structure. Use of this structure avoids reading original data at each iteration, and speeds up the execution. To overcome the effect of random initialization wrapper methods are practiced where the algorithm is executed multiple times and the best clustering is selected. The method has marked computational expense. Bradley et al. [13] propose a refinement scheme for choice of initial seed points. This strategy is particularly useful for large data sets where wrapper approach is infeasible. Since K-means algorithm can identify structures in linear data spaces, kernel K-means has been proposed to identify clusters in non-linearly separable input space [14].

Kuncheva and Vetrov examine the stability of K-means cluster ensemble with respect to random initialization [3]. They give empirical evidence of the stability of the ensemble using both pairwise and non pairwise stability metric. In

[1] On the day of writing this paper, google scholar search for ”K-means clustering” yielded more than 76K hits for articles (excluding patents).

other work, Kuncheva and Hadjitodorov propose a variant of the generic pairwise ensemble approach which enforces diversity in the ensemble [15]. A fusion procedure has been proposed in [16] to handle initialization dependency and selection of the number of clusters.

1.1 Problem Definition

Let D denote a data set of N, d-dimensional vectors $x =< x_1, x_2, \ldots, x_d >$, each representing an object. D is subjected to a clustering algorithm which delivers a clustering scheme π consisting of K clusters. ($\pi = \{C_1, C_2, \ldots, C_K\}$). Let $\{\pi_1, \pi_2, \ldots, \pi_H\}$ be H schemes of D obtained by applying either same clustering algorithm on D or by applying H different clustering algorithms. Further, let $\lambda_\pi : D \rightarrow \{1, K\}$ be a function that yields labeling for each of the N objects in D. Let $\{\lambda_1, \lambda_2, \ldots, \lambda_H\}$ be the set of corresponding labelings of D. The problem of cluster ensemble is to derive a consensus function Γ, which combines the H partitions (via labelings) and delivers a clustering π_f, with a promise that π_f is more robust than any of constituent H partitions and *best* captures the natural structures in D. Figure 1 shows the process of construction of cluster ensemble.

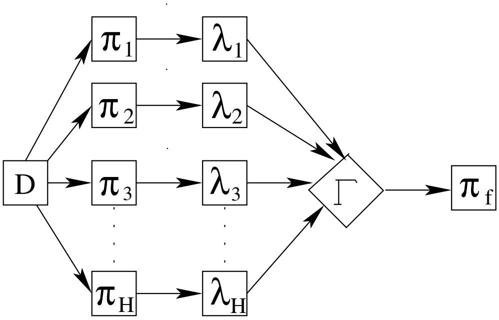


Fig. 1. The process of Cluster Ensemble

Combining the multiple partition is the core task in cluster ensemble problem which is accomplished by design of consensus function Γ. It is the design of Γ that distinguishes different cluster ensemble algorithms to a large extent. Hypergraph partitioning [1], voting approach [7], mutual information [17, 1], co-associations [16, 4, 18] are some of the well established approaches for design of consensus functions.

1.2 Our Approach

We propose to design a K-means cluster ensemble using a well known multivariate statistical analysis technique of Discriminant Analysis (DA). The motivation for using discriminant analysis comes from the ability of the technique to identify observed differences in multivariate data divided into two or more groups. The identification is carried out by finding 'discriminants' whose numerical values are such that the groups are separated as much as possible [19]. The technique was first introduced by R.A.Fisher and is widely used in statistics for classification in situations where information is either incomplete or expensive [19].

Given H clustering schemes, first the cluster labels are rearranged so as to set correspondance among the clusters. Discriminant function is computed for each scheme and is used to predict the cluster labels of the tuples in D. This process yields NXH label matrix, which essentially consists of predicted labels by each of the partitions for data tuples in D. Based on the user specified threshold, consistent predictions form the part of final clusterings. Tuples with low consistency predictions are iteratively refined for membership, to the best extent possible. If no further refinement is possible they are reported as noise to the user.

Robust Clustering Using Discriminant Analysis (RCDA) algorithm has the following salient features
(i) The algorithm requires two scans of data after clustering.
(ii) Discriminant analysis, a non parametric statistical technique has been utilized for consensus.
(iii) The noisy data is filtered out.
(iv) Experimental analysis of several UCI Machine learning data sets show that the consensus clustering has improved the accuracy, quality, purity, stability and consistency of clustering as compared to the original clusterings.

The paper is organized as follows. Section 2 describes the recent works in the area of cluster ensemble. Section 3 describes the proposed algorithm in detail. Section 4 briefly describes the quality criteria used to evaluate the cluster ensembles. Section 5 describes experimental analysis and finally Section 6 concludes the paper.

2 Related Work

Cluster ensemble technique has been widely studied by machine learning and data mining research community. An informative survey of various cluster ensemble techniques can be found in [1]. We describe some of the well known approaches followed in design of consensus functions, from recent works in cluster ensemble.

In CESG [20], the authors propose a cluster ensemble framework for gene expression analysis to generate high quality and robust clustering results. This clustering has been based upon the concept of distance matrix and weighted graph. In this framework, the clustering results of the individual clustering algorithm are converted into the distance matrix, these distance matrices are combined and a weighted graph is constructed according to the combined matrix. Then a graph partitioning approach is used to cluster the graph to generate the final clusters.

The adaptive clustering ensemble proposed in [2] is inspired by the success of sampling techniques. Clustering is based upon the consistency indices and sampling probability. Individual partitions in the ensemble are sequentially generated by clustering specially selected subsamples of the given data set. The sampling probability of each data point dynamically depends upon the consistency of its previous assignment in the ensemble.

Probabilistic model of finite mixture of multinomial distribution has been used in [5] for design of consensus function. In [21], authors investigate the commonalities and differences between the categorical clustering and cluster ensemble approaches. They propose a novel algorithm for designing cluster ensemble using concepts from categorical clustering. In ([4],[22]), authors proposed the data resampling approach for building cluster ensembles that are both robust and stable.

In [23], authors give the concept of cluster ensemble based upon multi-clustering fusion algorithm in which different runs of a clustering algorithm are appropriately combined to obtain a partition of the data that is not affected by initialization and overcomes the instabilities of clustering methods. Improvement in the performance of clustering analysis by using Cluster based Similarity Partitioning Algorithm (CSPA), Hypergraph Partitioning Algorithm (HGPA) and Meta Clustering Algorithm (MCLA) cluster ensemble approach has been claimed in [17].

3 Robust Clustering Using Discriminant Analysis

RCDA algorithm operates in three phases. In the first phase, it creates H clustering schemes from data set by applying K-means clustering algorithm as many number of times. Relabeling of the clusters in the partitions is also done during this phase. In the second phase the algorithm constructs discriminant functions corresponding to each partition. This is a compute intensive phase of the algorithm and needs no user parameter. Label of each tuple in D is predicted by each of the H discriminant functions and a NXH label matrix (L) is constructed.

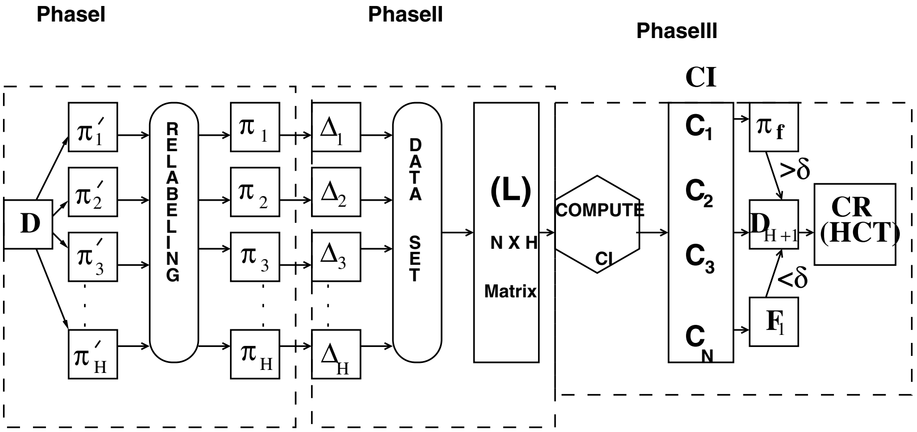


Fig. 2. Architecture of RCDA Algorithm

Finally, in the third phase tuples with consistent labels in L are assigned to clusters in the final partition, and the tuples with low consistency are iteratively refined.

Figure 2 describes the algorithm pictorially. $\pi'_1, \pi'_2, \ldots, \pi'_H$ are the H partitions generated during phase I. Using one partition as the reference partition, relabeling is performed resulting into schemes $\pi_1, \pi_2, \ldots, \pi_H$. Discriminant function Δ_i is constructed for scheme π_i $(i = 1, \ldots, H)$, to predict the cluster label of each tuple in D. The NXH label matrix L is constructed in which l_{ij} is the label of the cluster in which tuple i falls (a member of), according to π_j. Finally, for each object consistency of prediction is assessed. Tuples which are found to be consistent are used to refine the remaining inconsistent tuples.

3.1 Initialization Phase

Phase I of the algorithm is preparatory in the sense that during this phase H partitions are obtained by as many applications of K-means algorithm on D.

```
Input  : $K$ :number of cluster, $D$ :data set, $H$ :number of clustering schemes
         (partitions)
Output: $H$ clustering schemes $\pi_1, \pi_2, \ldots, \pi_H$ with corresponding labeled clusters
begin
    for $i = 1$ to $H$ do
        Apply K-means algorithm on $D$ to deliver partition $\pi_i$
    end
    Arbitrarily select $\pi_i, (1 = 1, \ldots, H)$ as $\pi_{ref}$
    for $i = 1$ to $H$ do
        if $(\pi_i <> \pi_{ref})$ then
            Relabel clusters in $\pi_i$ using distance from centroids.
        end
    end
end
```

Algorithm 1. Algorithm for Initialization Phase

Once H partitions have been obtained, the cluster labels need to be coordinated. Since there is no explicit correspondence between the clusters of different partitions, label correspondence problem is solved by taking arbitrarily one partition to be the reference partition π_{ref}. For relabeling partition π_i, the distances of centroids of the clusters in π_i are computed from those in reference partition. The cluster with the centroids closest to those in π_{ref} are assigned labels as in π_{ref}.

This method of relabeling has been chosen because of its efficiency. A more expensive bipartite graph matching based approach has been used in [4, 6] for this purpose. Algorithm 1 shows the steps for phase I.

Let $O(NKt)$ be the complexity of K-means algorithm, K being the number of clusters, t the number of iterations and N the number of tuples in D. The time complexity of phase I is $O(NKtH) + O(K^2)$. The former component is the cost of H runs of K-means algorithm and the latter is the cost of relabeling.

3.2 Predicting Labels Using Discriminant Analysis

Having set the correspondence between the clusters in H partitions, the algorithm proceeds to its core phase. For each partition, a discriminant function is constructed which is used to predict its membership of all records in the data set. One data scan is required for this purpose.

3.2.1 Applying Discriminant Analysis

Discriminant Analysis is a statistical method to separate between distinct classes in multivariate data. It establishes relationships between attributes for classifying objects into one of the several populations, by identifying attributes that best discriminate between the members of the group. In this method, one can judge the maximum discrimination of the tuple to the specific cluster through the discriminant score.

Given p-variate data from K populations $P_1, P_2, \ldots, P_K$ which map to K clusters $(C_1, C_2, \ldots, C_K)$. Cluster C_i contains n_i members, which are similar to each other. Let $X_1, X_2, \ldots, X_p$ denote the p attributes of data objects. The thrust in discriminant analysis is to form a linear function of these variables (Eqn. 1) for each of the K populations.

$$L = \beta_1 * X_1 + \beta_2 * X_2 + \ldots + \beta_p * X_p \tag{1}$$

L is subsequently used to assign a new object to one of the K populations. Computing the discriminant function for each population is the core task in discriminant analysis. This is done by taking into account the variability and correlations between the attributes for each population. We describe the method in detail adapting notation from [19].

Let x_{ijk} denote the value of attribute X_j, for k^{th} object $(1 \leq k \leq n_i)$ of the i^{th} cluster C_i. Thus $< x_{i1k}, x_{i2k}, \ldots, x_{ipk} >$ denotes the attribute vector of the k^{th} object in C_i. In order to determine the discriminant function (L), $\beta' s$ need to be determined in such a way that they provide maximum discriminating capabilities among the clusters. It is important to note that the focus of the estimation is the precision with which the discriminant function correctly classifies sets of observations, rather than the methods for optimization [19, 24].

Let $\overline{x_{ij}} = \frac{1}{n_i} \sum_{k=1}^{n_i} X_{ijk}$ be the observed mean of attribute j for the cluster C_i. Let $\overline{x}_i$ denote the centroid of the cluster C_i. Let $\sum_{k=1}^{n_i} (x_{ijk} - \overline{x_{ij}})^2$ be the squared sum of differences of values of the j^{th} attribute from the mean value for C_i.

Thus $s^i_{jj} = \frac{1}{n_i} \sum_{k=1}^{n_i} (x_{ijk} - \overline{x_{ij}})^2$ is the variance component estimation of the attribute j and $s^i_{jj'} = \frac{1}{n_i} \sum_{k=1}^{n_i} (x_{ijk} - \overline{x_{ij}})(x_{ij'k} - \overline{x_{ij'}})$ is the covariance component where $j \neq j'$ for C_i.

S_i gives the variance-covariance matrix of the cluster C_i.

$$S_i = \begin{pmatrix} s^i_{11} & s^i_{12} & \cdots & s^i_{1p} \\ s^i_{21} & s^i_{22} & \cdots & s^i_{2p} \\ \cdots & \cdots & \cdots & \cdots \\ s^i_{p1} & s^i_{p2} & \cdots & s^i_{pp} \end{pmatrix}$$

```
Input  : $\pi_1, \pi_2, \ldots, \pi_H$
Output: $NXH$ label matrix $L$
begin
    for $j = 1$ to $H$ do
        for $i = 1$ to $K$ do
            Compute variance covariance matrix $S_i$
        end
        Compute pooled variance-covariance matrix $Spooled$ for $\pi_j$
    end
    for $x = 1$ to $N$ do
        for $j = 1$ to $H$ do
            for $i = 1$ to $K$ do
                Compute $DScore_i(x)$ for tuple $x$ using discriminant function $D_i$
                $l_{xj} \longleftarrow$ label of the cluster with maximum DScore
            end
        end
    end
end
```

Algorithm 2. Algorithm for predicting cluster labels using Discriminant Analysis

Define $nm = (n_1 - 1)S_1 + (n_2 - 1)S_2 + \cdots + (n_K - 1)S_K$

$S_{pooled} = \frac{1}{n_1 + n_2 + \cdots + n_K - K} * nm$ is the pooled variance covariance matrix of the clustering schemes. The D Score of tuple x for the i^{th} cluster of the partition is computed as follows $DScore_i(x) = (\overline{x_i})' \ S_{pooled}^{-1}$ x - $\frac{1}{2}$ $(\overline{x_i})' \ S_{pooled}^{-1} \ \overline{x_i}$ + ln p_i, where p_i is the prior probability of cluster C_i and $\overline{x_i}'$ is the transpose of the centroid using the discriminant functions for each of the H clustering scheme, is class label of each record of D is predicted resulting into label matrix L . Detailed algorithm of this phase is presented in Algorithm 2.

This phase of RCDA is most compute intensive since $(H * K)$ variance-covariance matrices need to be computed. However, this phase does not require any user parameter and is purely data driven.

3.3 Refinement of Clustering Scheme

The label matrix L is processed in the final phase of the algorithm. The user specifies consistency threshold (δ) which is used to segregate consistent and non- consistent tuples. Non-consistent tuples are iteratively refined, till it is not possible to do so. The ones that are left are designated as noisy tuples or outliers. Consistency of the tuple is quantified by using an intuitive method suggested in [2].

Since the cluster correspondence is already established among the H partitions, for a stable clustering algorithm, each tuple must have same cluster label in all the partitions. Thus consistency of the label prediction can be estimated by $CI = Max_{(i=1,K)}(p_i)$, where $p_i = \frac{\text{Number of predictions of label } i}{H}$, $1 \leq i \leq K$ being the cluster label. Thus CI quantifies the maximum confidence with which

```
    Input  : $NXH$ label matrix , $\delta$ where $\delta$ is the consistency threshold
    Output: $\pi_f$ and Noise
 1  begin
 2      for $i = 1$ to $N$ do
 3          Compute consistency score $CI$
 4          if $CI \geq \delta$
 5          Assign tuple to corresponding cluster in $\pi_f$
 6      end
 7      if ($K$ cluster in $\pi_f$) then
 8          Compute discriminant function $D^f$ from $\pi_f$.
 9          Predict remaining (inconsistent) tuples using $D^f$
10          Assign to corresponding cluster in $\pi_f$.
11      end
12      else
13          Recompute discriminant functions from
14          the remaining tuples in $\pi_1, \pi_2, \ldots, \pi_H$
15          Predict the remaining tuples
16          Compute consistency score of tuple $i$
17          Assign high score tuples to corresponding cluster in $\pi_f$.
18      end
19      Repeat Step 7 to 18 until no change in tuple status.
20      Report remaining tuples as noise.
21  end
```

Algorithm 3. Algorithm for Phase III(Refinement Phase)

cluster label i can be assigned to the tuple. Tuple with consistency above the user specified threshold δ are assigned to the corresponding cluster of the final partition π_f. The tuples that remain are the ones which do not have the desired level of consensus among their labels. If the number is very small and acceptable to the user, these can be discarded (or investigated) as noise, otherwise a refinement step is carried out as described below. If all K clusters are represented in π_f, a new discriminant function D^f is constructed from π_f. The labels of low consistency tuples are predicted by D^f and the tuples are added to the appropriate clusters in π_f.

In case there are outliers in the data, it is possible that all K clusters are not represented in π_f in the first iteration. In such a situation, all the tuples that belong to the missing cluster have low consistency score. Thus there is a need to iteratively improve the cluster quality of π_f. For the remaining tuples in the partitions $(\pi_1, \ldots, \pi_H)$, discriminant functions are recomputed and the tuples are predicted. This process is repeated till the consistency scores of the tuples do not improve beyond the threshold δ. The tuples whose consistency does not improve are reported as noise to the user. Detailed algorithm of this phase is described in the Algorithm 3.

3.4 Discussion

Though RCDA is targeted to overcome the instability of K-means algorithm, the approach is general enough to be applied in other cluster ensemble problems. Initial partitions $\pi^{'}_1, \pi^{'}_2, \ldots, \pi^{'}_H$ can be obtained in multiple ways depending on the environment and application at hand. The H folds of the data can be created by random sampling without replacement. Each fold may be clustered independently yielding H partitions $\pi_1^{'}, \pi_2^{'}, \ldots, \pi_H^{'}$. In case the data is voluminous ($\geq 100K$ tuples) then in order to achieve scalability H random samples of same size may be drawn from data with replacement, to create $\pi_1^{'}, \pi_2^{'}, \ldots, \pi_H^{'}$.

RCDA algorithm is suitable for data with linearly separable clusters. Discriminant analysis technique captures linear relationship between attributes in a cluster. For non linear groupings in the data Kernel K-means is used [14]. However it is non-trivial to adapt discriminant analysis for this purpose. Further, since discriminant analysis requires computation of variance covariance matrices for computation of discriminant function, its algorithm does not scale well with increasing data dimensionality. The algorithm gives the best results when the number of natural clusters (K) in data is known.

4 Assessing Quality of Ensemble

The cluster ensemble is computationally expensive proposition and hence must deliver reasonable benefit to the user in terms of cluster quality. There is no *best* measure for evaluating the cluster quality. However a mix of internal and external quality criteria can be employed to empirically establish the superiority of the proposed method. We employ the following measures for this purpose as defined in [25].

1. Purity: Purity of a clustering scheme is an external quality criterion and is used when classes in the data are known. A class then corresponds to a cluster, and a cluster with all the objects belonging to one class is considered pure.
   Let there be K clusters in the data set D and size of cluster C_j be $|C_j|$. Let $|C_j|class = i$ denote number of objects of class i assigned to C_j. Purity of C_j is given by

$$Purity(C_j) = Max_{(i=1,K)} \frac{|C_j|class = i}{(|C_j|)} \tag{2}$$

   The overall purity of a clustering solution is expressed as a weighted sum of individual cluster purities

$$Purity = \sum_{j=1,K} \frac{|C_j| * Purity(C_j)}{|D|} \tag{3}$$

   In general, larger value of purity indicates better quality of the solution.
2. Normalized Mutual Information (NMI) : The optimal combined clustering should share the most information with the original clusterings [6, 17]. Normalized Mutual Information (NMI) captures the commonality between two clustering schemes as described below.

Let A and B be the random variables described by the cluster labellings $\lambda(a)$ and $\lambda(b)$ with $k(a)$ and $k(b)$ groups respectively. Let $I(A,B)$ denote the mutual information between A and B, and $H(A), H(B)$ denote the entropy of A and B respectively. It is known that $I(A,B) \leq \frac{H(A)+H(B)}{2}$. Normalized mutual information (NMI)[26] between the two clustering schemes is defined as

$$NMI(A,B) = 2I(A,B)/H(A) + H(B) \tag{4}$$

Naturally $NMI(A,A) = 1$. Eqn 4 is estimated by the labels provided by the clustering. Let $n^{(h)}$ be the number of objects in cluster c_h according to $\lambda(a)$ and let n_g be the number of objects in cluster c_g according to $\lambda(b)$. Let n_h^g be denote the number of objects in cluster c_h according to $\lambda(a)$ as well as cluster c_g according to $\lambda(b)$. The normalized mutual information criteria $\phi(NMI)$ is computed as follows

$$\phi^{(NMI)}(\lambda(a), \lambda(b)) = \frac{2}{n}(\sum_{h=1}^{k^{(a)}} \sum_{g=1}^{k^{(b)}} (n_g^{(h)}) \log_{k^{(a)} k^{(b)}} \frac{n_g^{(h)} * n}{n^h * n_g}) \tag{5}$$

3. Adjusted Rand Index (ARI): The Adjusted Rand Index is an external measure of clustering quality which takes into account biases introduced due to distribution sizes and differences in the number of clusters. The quality of clustering $R(U,V)$ can be evaluated by using the Adjusted Rand Index as

$$R(U,V) = \frac{\sum_{(l,k)}(n_{lk}C2) - [\sum_l(n_lC2) * \sum_k(n_kC2)]}{(1/2) * [\sum_l(n_lC2) + \sum_k(n_kC2)] - [\sum_l(n_lC2) * \sum_k(n_kC2)]} \tag{6}$$

where $l, k =$ clusters representation. $n_{lk} =$ number of data items that have been assigned to both cluster l and cluster k. $n_l =$ number of data items that have been assigned to cluster l. $n_k =$ number of data items that have been assigned to cluster k. $n =$ Total number of data items. The Adjusted Rand Index return values in the interval [0,1] and is to be maximized.

5 Experimental Analysis

RCDA (Robust Clustering Using Discriminant Analysis) algorithm was implemented as a multithreaded C++ program and tests for cluster quality were carried out using synthetic, standard UCI machine learning [28] and CLBME repository data sets [29].

Preliminary investigations were carried out on synthetic data generated using ENCLUS data generator [27]. Use of synthetic data allows validating the algorithm. Data set D was generated consisting of 1000 records, distributed in 4 clusters. The cluster sizes were 300, 300, 200, 200 respectively. RCDA was applied on D with $K = 4$ and H varying as 4, 8, 12, 16, 20. $H = 16$ was found to be partition giving best measures (Purity, Mutual Information and Adjusted Rand Index) as computed using Eqns 3 and 6. Results obtained for $H = 16$ are

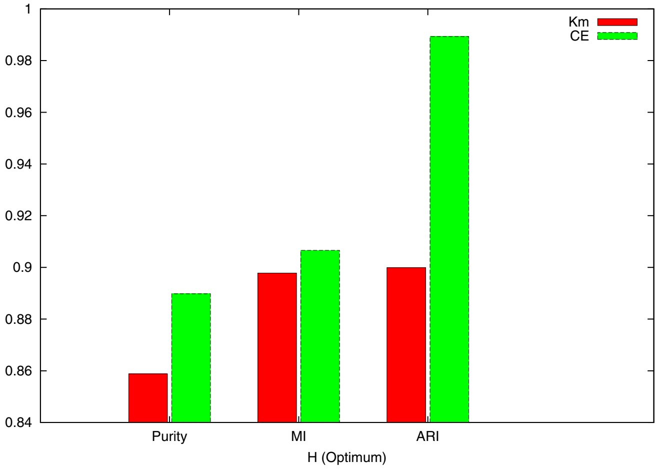


Fig. 3. Comparison of quality measures for synthetic data set. Km is the value of the best metric among all the 16 clustering schemes generated to create the ensemble. CE is the corresponding metric for RCDA ensemble.

Table 1. Details of the data sets; (1) from UCI repository; (2) from CLMBE repository; H: optimum number of partitions

Data Set	**Tuples**	**Dimensions**	**Classes**	H
Wine (1)	178	13	3	4
Winconsion Breast Cancer (1)	683	11	2	8
Respiratory (1)	85	17	2	4
Lymphography (1)	148	18	4	10
Iris (1)	150	4	3	8
Laryngeal (2)	353	16	3	6
Voice3 (2)	238	10	3	4
Voice9 (2)	428	10	9	12

plotted against the corresponding best measures among the sixteen partitions. It was further observed that the best measure values for purity, NMI and ARI come from different partition among the 16 clustering schemes (Figure3).

Five data sets from UCI [28] (Wine, Winconsion Breast cancer, Respiratory, Lymphography, Iris) and 3 data sets from CLBME [29] (Laryngeal, Voice3 and Voice9) were used for evaluation of RCDA algorithm. The characteristics of these data sets are shown in the Table 1. For each of these data sets, experimentation was made by varying the number of partitions in the ensemble and the value of H for which the best combination of purity, NMI and ARI was noted. The value of H that appears in the last column of the Table 1, was used for the evaluating the cluster quality. For each data set, an ensemble partition was constructed using

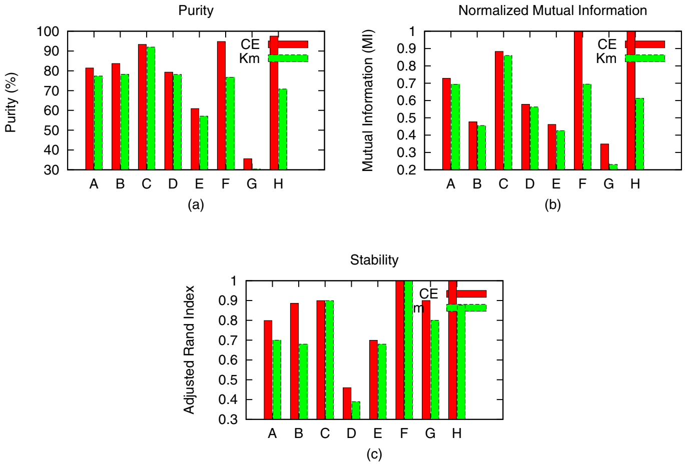


Fig. 4. Comparison of the three quality measures for UCI and CLBME data sets. Km is the value of the best metric among all the H clustering schemes and CE is the corresponding metric for RCDA ensemble. Data sets A:Wine, B:Winconsion Breast Cancer, C:Iris, D:Laryngeal, E:Voice3, F:Lymphography, G:Respiratory and H:Voice9.

the corresponding optimum H value. The three metrics were computed for each of the H partitions individually and the ensemble (π_f). Then for each metric the best value among H clustering schemes was ploted for comparison with RCDA ensemble (Figure 4). It is evident that all three measures are improved in RCDA ensemble for each of the datasets. However the extent of improvement varies for each dataset.

6 Conclusion and Future Work

We propose a novel algorithm Robust Clustering using Discriminant Analysis (RCDA) for designing a cluster ensemble using a well known statistical technique of discriminant analysis. The algorithm aims to overcome the instability of K-means algorithm that arise because of random initialization and data order sensitivity. The motivation for using discriminant analysis arises because of the non-parametric and parameterless nature of the method. The algorithm operates in three phases and requires two scans after the initial clusterings have been done (in phase I). During phase II discriminant functions are computed and cluster labels of all tuples in the data set are predicted. This is the compute intensive phase of the algorithm. In the final phase, the predictions are combined using consistency index and iterative refinement is carried out. The tuples that can not be refined are designated as noise. Preliminary experimentation on synthetic and publically available data sets demonstrates definite improvement in the cluster quality.

References

[1] Reza Ghaemi, M., Nasir Sulaiman, H.I., Mustapha, N.: A survey: Clustering ensembles techniques. In: Proceedings of World academy of science, Engineering and Technology 38, 2070–3740 (2070)

[2] Topchy, A., Behrouz Minaei-Bidgoli, A., Punch, W.F.: Adaptive clustering ensembles. In: ICPR, pp. 272–275 (2004)

[3] Kuncheva, L., et al.: Evaluation of stability of k-means cluster ensembles with respect to random initialization. IEEE Transcations on pattern analysis and machine intelligence 11(28), 1798–1808 (2006)

[4] Fred, A.L.N., Jain, A.K.: Data clustering using evidence accumulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 835–850 (2002)

[5] Topchy, A., Jain, A.K., Punch, W.F.: A mixture model for clustering ensembles. In: SDM (2004)

[6] Strehl, A., Ghosh, J.: Relationship-based clustering and cluster ensembles for high-dim. data. PhD thesis (May 2002)

[7] Fischer, B., Buhmann, J.M.: Path-based clustering for grouping of smooth curves and texture segmentation. Transaction on Pattern Analysis and Machine Intelligence 25(4) (April 2003)

[8] Bock, H.H.: Origins and extensions of the k-means algorithm in cluster analysis. Electronic Journal for History of Probability and Statistics 4(2) (2008)

[9] Anderson, J., et al.: Machine Learning: An Artificial Intelligence Approach. Morgan Kaufmann, San Francisco (1983)

[10] Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn., Morgan Kaufmann Publishers, San Diego (August 2006)

[11] MacQueen, J.: Some methods for classification and analysis of mulivariate observations (2008)

[12] Tapas, K., et al.: An efficient k-means clustering algorithm:analysis and implementation. CIKM, Mcleen, Virginia, USA, vol. 24(7) (July 2002)

[13] Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: ICML 1998, May 1998, vol. 24, pp. 91–99 (1998)

[14] Dhillon, I.S., Yuqiang Guan, B.K.: Kernel k-means, spectral clustering and normalized cuts. In: KDD, Seattle, Washigton, USA (August 2004)

[15] I, K.L., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: Proceedings IEEE International Conference on Systems, Man and Cybernatics, The Netherlands, pp. 1214–1219 (2004)

[16] Fred, A.L.N.: Finding consistent cluster in data partitions. MCS 19(9), 309–318 (2001)

[17] Strehl, A., Ghosh, J.: Cluster ensemble knowledge reuse framework for combining partitions (2002)

[18] Topchy, A., Jain, A.K., Punch, W.: Combining multiple weak clusterings. In: Proceedings of the Third IEEE International Conference on Data Mining (2003)

[19] Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice-Hall, Upper Saddle River (August 1979)

[20] Hu, X., Yoo, I.: Cluster ensemble and its applications in gene expression analysis. In: 2nd Asia-pacific Bioinformatics Conference, Dunedin, New Zealand

[21] He, Z., Xiaofei, X., Deng, S.: A cluster ensemble method for clustering categorical data. In: Department of Computer Science and Engineering, Harbin Institute of Technology, China, August, vol. (2), pp. 153–172 (2002)

[22] Minaei-Bidgoli, B., Topchy, A., Punch, W.F.: Ensembles of partitions via data resampling, Michigan State University, East Lancing, MI, USA

[23] Frossyniotis, D., Stafylopatis, M.A.: A multi-clustering fusion algorithm. Journal of Computer Science and Technology 17(2), 118–128 (2002)

[24] Narain, Malhotra, P.: Handbook of statistical genetics. IASRI, New Delhi-12 and Printed at S.C.Printers (1979)

[25] Maimon, O., Rokech, L.: Data Mining and Knowledge discovery Handbook. Springer, Heidelberg (2004)

[26] Ankerst, M., Breuig, M.M., Kriegel, H.P., Sander, J.: Optics: Ordering points to identify the clustering structure. In: ACM SIGMOD 1999 Int. Conf. on Management of Data, Philadelphia, PA (1999)

[27] Chang, C.H., Fu, A.W., Zhang, Y.: Entropy based subspace clustering for mining numerical data. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 1999), San Diego (August 1999)

[28] Uci repository, http://www.ics.uci.edu

[29] http://www.clbme.bas.bg

New Approach in Data Stream Association Rule Mining Based on Graph Structure

Samad Gahderi Mojaveri[1], Esmaeil Mirzaeian[1], Zarrintaj Bornaee[2], and Saeed Ayat[3]

[1] M.Sc. Student of Payame Noor University, Nejatollahi Street, Tehran, Iran
{ghaderi.meisam,emirzaeian}@gmail.com
[2] Assistant Professor of Iranian Research Organization for Science & Technology, Tehran, Iran
bornaee@irost.ir
[3] Assistant Professor, Payame Noor University, Iran
ayat@ce.sharif.edu

Abstract. Discovery of useful information and valuable knowledge from transactions has attracted many researchers due to increasing use of very large databases and data warehouses. Furthermore most of proposed methods are designed to work on traditional databases in which re-scanning the transactions is allowed. These methods are not useful for mining in data streams (DS) because it is not possible to re-scan the transactions duo to huge and continues data in DS. In this paper, we proposed an effective approach to mining frequent itemsets used for association rule mining in DS named GRM[1]. Unlike other semi-graph methods, our method is based on graph structure and has the ability to maintain and update the graph in one pass of transactions. In this method data storing is optimized by memory usage criteria and mining the rules is done in a linear processing time.

Efficiency of our implemented method is compared with other proposed method and the result is presented.

Keywords: Data mining, Association rule mining, Data streams, frequent itemsets, Transaction, Occurrence list.

1 Introduction

Today, mining huge and continues data, which comes into databases with high rate, has caused lots of new challenges in association rule mining and knowledge discovery so that old algorithms which were applied in traditional databases, cannot be completely useable in mining of data streams (DS). Two such algorithms are Apriori [1] and FP-tree[2] [2] that has been proposed by Han and Yiwen. The first one can be known as the basic algorithm in knowledge discovery and it belongs to dynamic programming algorithms, because the results of lower levels are used in higher levels. The basic rule of this algorithm is this fact that each subset of frequent itemsets, is frequent itself. This two algorithms with their different characteristics were designed

[1] Graph based rule mining.

[2] Frequent pattern tree.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 158–164, 2010.

for traditional databases, therefore they were not useful for fast and huge cases like DS. Many algorithms were proposed to increase the efficiency of association rule mining based on Aapriori and all of them were more efficient than a version of Apriory which needed to scan the data in multiple pass.

In another method called FP-tree which was proposed for traditional databases by Han and Yiwen [2], by using the frequent pattern tree data structure, it was possible to mine frequent itemsets without making frequent candidate itemsets. This model of frequent pattern is stored in a tree structure called frequent pattern tree, in which new transactions are always updated. Frequent pattern tree is used to store transactions in current timed-window and for past timed-window a similar structure called pattern tree is used .in order to analyze data stream, this model uses two parameters, one was pattern tree and the other was tiled timed-window. FP-tree is more efficient than algorithm based on Apriori which are essentially a version of Apriori designed to be compatible to huge and continuous data streams as proposed by Miller [3], and its higher efficiency is because of prevention in producing of frequent candidate itemsets. But this algorithm is not suitable for Association rule mining in data streams either, because it needs to analyze data in two pass to make frequent pattern tree, and duo to mass and real time data streams, it is not acceptable because of being fast and huge DS most be scanned in one pass.

In order to mining association rules in DS new algorithm such as VFDT[3] (proposed by Maron and Moore [4]) and CFI-Stream[4] (proposed by Jiang and Gruenwald [5]) were introduced. In VFDT the decision tree method and in CFI-Stream DUI[5] tree is used.

This paper is organized as follows: Section 2 introduces our proposed graph. Section 3 deals with evaluation and results. Conclusion and future works are given in section 4.

2 Proposed Graph Structure for Data Streams

Our graph structure is designed for mining frequent itemsets in data stream and dynamically adds new transaction into current graph as soon as it comes into system in one pass. Moreover, to compute frequent itemsets, it acts same as dynamic programming algorithms, because it uses pair frequent items to compute long frequent items. The structure of this graph is designed so that it can maintain all entered transactions without any sliding window or time-window up to an unlimited time. Other semi-graph methods use multiple symbols in their graph structure, but GRM uses unique symbols in graph structure and by aid of optimized memory allocation tries to control large amount of continuous transactions. Moreover other proposed methods may use a single symbol for multiple nodes. This matter causes high redundancy and processing a single symbol will be time-consuming because one must search all nodes to find all occurrence of that node.

To form the proposed graph in Figure 1, each transaction should be separated to single symbols and new node should be create for each of them. If one node has been

[3] Very fast decision tree.

[4] Closed Frequent Itemsets in Data Streams.

[5] Direct update.

crated before, we just need to increase its counter. In next step we extract all possible edges according to sequence of symbols in each transaction. For example consider first transaction of Table 1 the extraction leads to set P= {AC, AT, AW, CT, CW, TW}. Each item of set P should be stored as an edge in graph. If such an edge has been placed before, we increase its counter and set starting symbol in its starters list. This information helps us to keep track and detachment of transactions in extraction phase, especially when they are overlapped in multiple edges.

It's very important to note that set P contains all possible edges between two symbols that holds the starter symbol. All other information needed to extraction phase is stored in our auxiliary proposed data structure called O-list[6] shown in figure 2. More detail about constructing o-list is introduced later in this section.

One of the main problems to design a method based on graph structure is finding all possible passes for each frequent item because a standalone graph structure cannot extract all different passes due to overlapping between transactions. To resolve this problem, a technique must store more information about transactions or use semi-graph

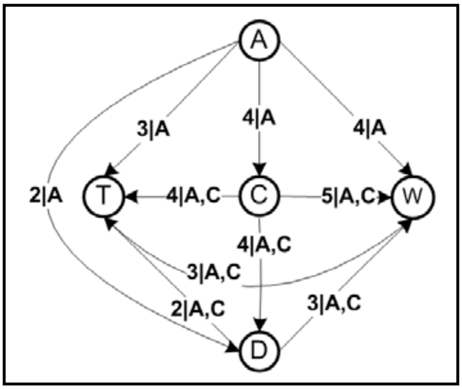


Fig. 1. Final graph from transactions in table 1

Table 1. Entered transactions in time order

Transaction No	Itemsets
1	ACTW
2	CDW
3	ACTW
4	ACDW
5	ACDTW
6	CDT

[6] Occurrence list.

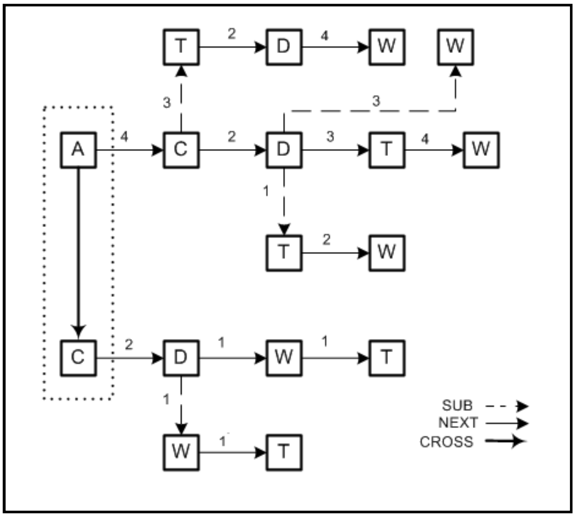


Fig. 2. O-list from transactions in Table 1

structure with more redundancy (as it is shown in Naganathan and Rsmesh [6]). As it said before in this section, we proposed another data structure to keep track of overlapped transaction and their occurrence. In the following more details about O-list are given.

Algorithms 1. Insert transaction

```
Step 1. extract all possible pairs as $T_i$ , $T_j$ into set P where i<j
Step 2. for each  pair p ∈ P as uv
            if edge uv already exists in graph
                        uv[count]=uv[count]+1
            else
                        add edge uv between vertex u and v in graph
                        uv[count]= 1
Step 3. add starter of set P in uv[starters]
Step 4. for each  pair p ∈ P as uv
            add edge uv into Olist[Starter of P]
```

Algorithms 2. Rules extraction

```
Step 1. for each  vertex v ∈ V in graph
            add v[count] into result set
Step 2. for each  edge uv ∈ E in graph
            add uv[count] into result set
Step 3. for subtransactions of all inserted transaction as stran
            search graph to find and set stran[starters]
            for each starter ∈ stran[starters]
                        add min-chain stran by Olist[starter]into result set
```

O-list consists of collection of nodes that are linked with three different links between them. First, cross link refers to nodes which were as starter at least once. Second, NEXT links refer to occurrence of a symbol after another symbol in which the first one is starter symbol. Third, all other links are SUB links.

Table 2 shows final extracted frequent itemsets related to Transactions in table 1, along with their corresponding confidence ratio.

Table 2. Extracted frequent itemsets

Confidence	Itemsets
100%(6)	C
83%(5)	W,CW
67%(4)	A,D,T,AC,AW,CD,CT,ACW
50%(3)	AT,DW,TW,ACT,ATW,CDW, CTW,ACTW

3 Results

The implementation of the GRM is able to mine rules with 100% accuracy and extracting all frequent itemsets. In order to evaluate GRM, we used both random and real dataset.

Random dataset is used in other similar methods such as [6] and [7]. In figure 3 the Evaluation result of GRM method in compare to method in [6] on random datasets is reported and it shows that GRM has more performance.

Faster processing time, less redundancy and optimized structure has resulted Figure 3.

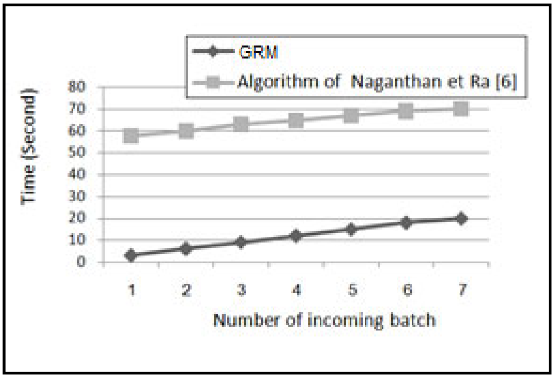


Fig. 3. Time comparison of GRM with the method offered in [6]

The BMS-WebView-1is used as Real dataset to Evaluate GRM method. The BMS-WebView-1 datasets contain several months' worth of clickstream data from e-commerce web sites. Each transaction in this datasets is a web session consisting of all the product detail pages viewed. it contains 59602 transaction with 497 itemsets and is known as standard dataset that has been used by ARMOR method [8] to evaluate the performance and efficiency. ARMOR focuses on the question of how much space remains for performance improvement over Oracle algorithm. in many association rules mining methods like ARMOR the performance depend on reducing minimum support but GRM method is independent from minimum support ratio and even the performance increases when transaction length become shorter.

Figure 4 shows the comparison result between GRM and ARMOR methods by applying two different transaction lengths on BMS-WebView-1 datasets.

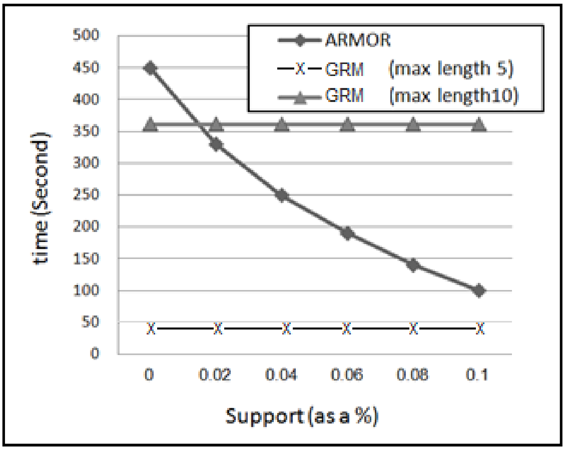


Fig. 4. Time comparison between GRM and ARNOR method

4 Conclusions and Future Work

This paper proposed a new approach to mining frequent itemsets used for association ule mining in data streams named GRM. GRM unlike other semi-graph methods, is based on graph structure and has ability to maintain and update graph in one pass of transactions. This method is able to handle huge and massive streamed data with complete accuracy, beside it is able to mine rules faster than other proposed method designed to work on data streams. The only limitation is related to making all possible sub transaction for long chains of transactions, therefore GRM method takes unacceptable time for chains of size 10 or above. Our future work will be applying more optimization, so that, processing time for long chain of transaction will be more scalable.

References

1. Agrawal, R., Imiliniski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Proc. of the ACM SIGMOD conference on Management of Data, Washington, D.C. (May 1993)
2. Han, J., Jian, P., Yin, Y.: Mining Frequent pattern without Candidate Generation. In: Int'l. Conf. on Management of Data (May 2000)

3. Miller, R.G.: Simultaneous Statistical Inference, 2nd edn. Springer, New York (1981)
4. Maron, O., Moore, A.: Hoe_ding races: Accelerating model selection search for classi_cation and function approximation. In: Cowan, J.D., Tesauro, G., Alspector, J. (eds.) Advances in Neural Information Processing Systems, vol. 6, Morgan Kaufmann, San Mateo (1994)
5. Jiang, N., Gruenwald, L.: CFI-Stream: Mining Closed Frequent Itemsets in Data Streams. In: ACM SIGKDD intl. conf. on knowledge discovery and data mining (2006)
6. Naganthan, E.R., Rsmesh Dhanaseelan, F.: Efficient Graph Structure for the Mining of Frequent Itemsets from Data Streams. International Journal of Soft Computing 3(2), 144–146 (2008)
7. Leung, C.K.-S., Kkan, Q.l.: DSTree: A Tree Structure for the Mining of Frequent Set from Data Streams. In: IEEE Sixth Int'l. Conf. on Data Mining (2006)
8. Pudi, V., Haritsa, J.R.: ARMOR: Association Rule Mining based on Oracle. In: Proc. of ICDM Workshop on Frequent Itemset Mining Implementations, Florida, USA (2003)

Fast Training of Neural Networks for Image Compression

Yevgeniy Bodyanskiy[1], Paul Grimm[2], Sergey Mashtalir[1], and Vladimir Vinarski[3]

[1] Kharkiv National University of Radio Electronics,
Computer Science faculty,
pr. Lenina 14, Kharkiv 61166, Ukraine
{bodya,mashtalir_s}@kture.kharkov.ua
[2] University of Applied Sciences,
Altonaer st. 25, Erfurt, 99085, Germany
grimm@fh-erfurt.de
[3] University of Applied Sciences and Arts,
Ricklinger Stadtweg 120, Hannover 30459, Germany
vladimir.vinarski@fh-hannover.de

Abstract. The paper considers the problem of image compression by using artificial neural networks (ANN). The main concept of this approach is the reduction of the original feature spaces, what allows us to eliminate the image redundancy and accordingly leads to their compression. Two variants of the neural networks: two layers ANN with the self-learning algorithm based on the weighted informational criterion and auto-associative four-layers feedforward network have been proposed and analyzed.

Keywords: image compression, artificial neural networks, self-learning algorithm.

1 Introduction

When solving various tasks related to processing signals of high dimension, prevalently in the field of image processing, rather often one faces the problem of reducing the dimension of original image vectors with the minimal loss of information. For this purpose most often the principal component and principal subspaces analysis tools based on the Karhunen-Loeve linear transformation are used. At the same time the linear technique is not always able to detect more complex relations which are common for real images. At that in the number of cases a non-linear approaches based on neural networks technologies can be more effective [1].

Thus in [2] the task of information compression using a two layers neural network with direct information transmission was considered and it was shown that regardless of the used activation functions type, such architecture performs the standard principal component analysis (PCA). In [1] a three-layers auto-associative neural networks with error backpropagation learning algorithm was proposed and experiments connected with image processing were carried out. However the comparative analysis with other technologies was not mentioned. In [3-5] it is shown that the optimal

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 165–173, 2010.

compression can be provided by using four-layers auto-associative neural networks with interchange of the linear and non-linear layers and examples of image processing are given. This network performs the non-linear principal component analysis (NLPCA) and is trained using standard error backpropogation gradient algorithm. This algorithm however has law convergence rate, does not consider lack of networks architecture uniformity and it is rather sensitive to the distortions of different kinds.

All the aforesaid brings enough urgency and significance to the task of construction and analysis of real time fast-operating parallel artificial neural networks (ANN), which provide finding the principal components of correlation matrix where data are sequentially fed for the processing.

2 Architecture of Two Layers ANN for Finding Principal Components and Its Optimal Learning

It is assumed that the original information is given as a fixed data array formed by N n-dimensional vectors $x(1), x(2), \dots, x(k), \dots, x(N)$, where $k = 1, 2, \dots$ are numbers of observations in the original data array, and the result is a set of eigen values $\lambda_1 > \dots > \lambda_j > \dots > \lambda_m$ and corresponding eigen vectors $w_1, w_2, \dots, w_j, \dots, w_m$, $w_j = (w_{j1}, w_{j2}, \dots, w_{jn})^{\mathrm{T}}$ of the original data correlation $(n \times n)$ matrix

$$\begin{cases} R(N) = \dfrac{1}{N} \sum_{k=1}^{N} (x(k) - \bar{x}(N))(x(k) - \bar{x}(N))^{\mathrm{T}}, \\ \bar{x}(N) = \dfrac{1}{N} \sum_{k=1}^{N} x(k). \end{cases} \tag{1}$$

The compression itself (dimensionality reduction) of the original space is done by the transformation

$$y(k) = Wx(k) \tag{2}$$

where $y(k) = (y_1(k), y_2(k), \dots, y_m(k))^{\mathrm{T}}$ and $W = (w_1, w_2, \dots\ w_{m-1}, w_m)^{\mathrm{T}}$ is a $(m \times n)$ projective matrix formed by dominant eigenvectors of correlation matrix $R(N)$.

Architecture of the ANN solving this problem is shown on figure 1. The network has two layers formed by m (in first hidden layer) and n (in output layer) adaptive linear associators. In first hidden layer which synaptic weights form $(m \times n)$ matrix $W = \{w_{ji}\}$, the compression of information is fulfilled, at that on its output the values of principal components $y_1, y_2, \dots, y_m$ are calculated. The output layer is used for restoring the input signal with the help of $(n \times m)$ synaptic weights matrix $W^{\mathrm{T}} = \{w_{ij}\}$. The result of it are output values $\hat{x}_1(k), \hat{x}_2(k), \dots, \hat{x}_n(k)$, which are the estimates of the input signal

$$\hat{x}(k) = W^{\mathrm{T}}(k-1)y(k) = W^{\mathrm{T}}(k-1)W(k-1)x(k). \tag{3}$$

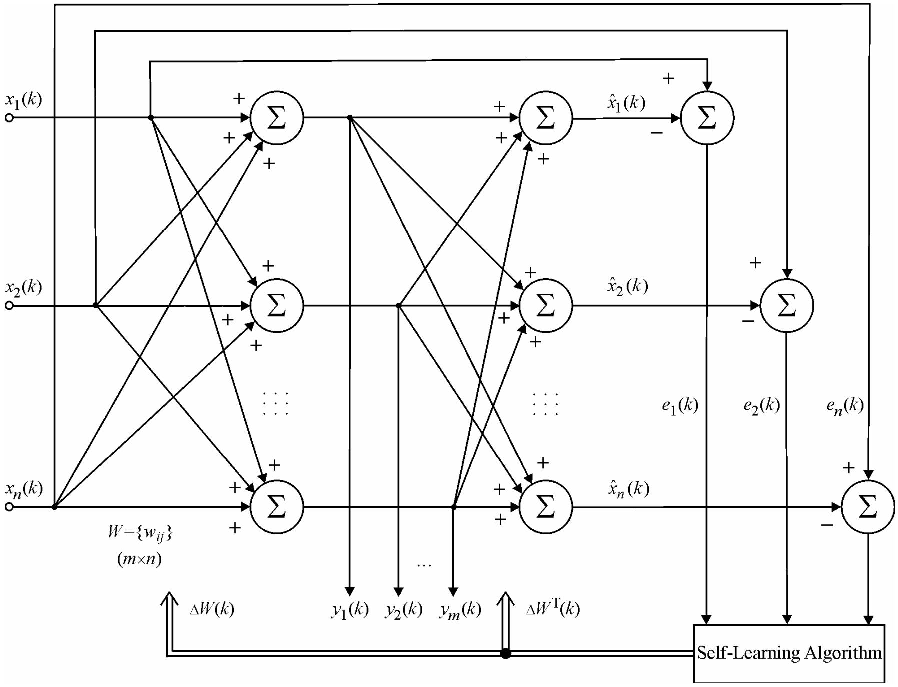


Fig. 1. Parallel ANN for the principal components search

It is obvious that such restoring is possible under $m=n$, but under $m<n$ the restoring with maximal possible accuracy in terms of criterion

$$E(k)=\left\|\tilde{e}(k)\right\|^2=\left\|x(k)-W^{\mathrm{T}}(k)y(k)\right\|^2=\left\|x(k)-W^{\mathrm{T}}(k)W(k)x(k)\right\|^2. \qquad (4)$$

can be obtained.

The proposed optimal self-learning algorithm based on weighted information criterion can be formalized as following

$$W(k)=W(k-1)+\frac{\left(x(k)-W^{\mathrm{T}}(k-1)y(k)\right)^{\mathrm{T}}G(k)y(k)}{\left\|G(k)y(k)\right\|^2}G(k) \qquad (5)$$

where $G(k)=-((A^{-1}W(k-1)R(k)W^{\mathrm{T}}(k-1)A)^{-1}W(k-1)(k)-W(k-1))$.

The given algorithm is a generalization of procedures proposed in [6-8] having at that a high operating speed due to its projective properties.

3 Architecture of Four Layers ANN and Its Optimal Learning

The architecture considered above implements the idea of linear PCA. To increase the compression quality it is reasonable to implement NLPCA, realized by more complicated architecture.

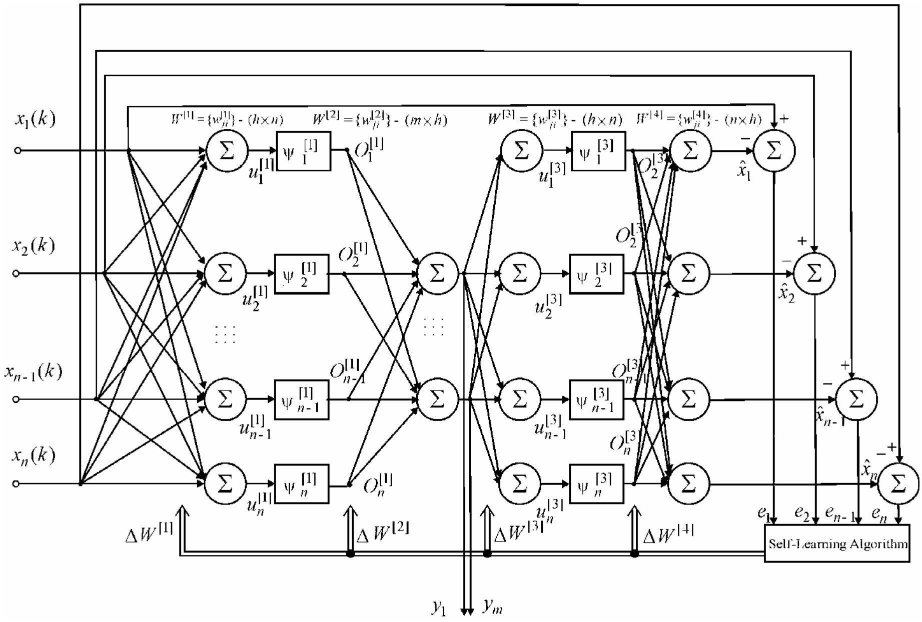


Fig. 2. Auto-associative multi-layers ANN for image compressing

Architecture of the auto-associative neural network designed to reduce dimensionality of original image space is shown on figure 2 and contains four sequentially connected layers of neurons.

To the network input (receptive zero layer) a sequence of image vectors $x(1), x(2), ..., x(k), ..., x(N)$, $x(k) = (x_1(k), ..., x_i(k), ..., x_n(k))^T \in R^N$ is given, being preliminary encoded on hypercube so that $x_i(k) \in [-1,1], i = 1,2,...,n$.

First hidden layer includes $h \geq n$ neurons as elementary Rosenblatt perceptrons with sigmoidal activation function. If the bias signal is absent the first hidden layer contains hn tuned synaptic weights and described by relations

$$o_j^{[1]}(k) = \psi_j^{[1]}(u_j^{[1]}(k)) = \psi_j^{[1]}(\sum_{i=1}^{n} w_{ji}^{[1]} x_i(k)), j = 1,2,...,h,$$

$$o^{[1]}(k) = \Psi^{[1]}(\mathrm{W}^{[1]} x(k)), \tag{6}$$

where $o_j^{[1]}(k)$ is an output signal of j-th neuron of the first hidden layer, $\psi_j^{[1]}(\circ)$ is an activation function of the first hidden layer's j-th neuron, $u_j^{[1]}(k)$ is the first hidden layer's j-th neuron inner activation signal, $w_{ji}^{[1]}$ is the synaptic weight on i-th input of the first hidden layer's j-th neuron, $o^{[1]}(k) = (o_1^{[1]}(k), ..., o_j^{[1]}(k), ..., o_h^{[1]}(k))^{\mathrm{T}}$, $\Psi^{[1]} = diag(\psi_1^{[1]}, ..., \psi_j^{[1]}, ..., \psi_h^{[1]})$ is a $(h \times h)$

diagonal operator of activation functions, $\mathrm{W}^{[1]}=\{w_{ji}^{[1]}\}$ is a $(h\times n)$ synaptic weights matrix.

If the first hidden layer neurons include threshold signal the relations stated above take form

$$o_j^{[1]}(k)=\psi_j^{[1]}(u_j^{[1]}(k))=\psi_j^{[1]}(\sum_{i=1}^{n} w_{ji}^{[1]}x_i(k)+\theta_j^{[1]})=\psi_j^{[1]}(\sum_{i=0}^{n} w_{ji}^{[1]}x_i(k)),$$

$$o^{[1]}(k)=\Psi^{[1]}(\mathrm{W}^{[1]}x(k)) \tag{7}$$

where $\theta_j^{[1]}$ is a bias signal of the first hidden layer's j-th neuron, $w_{j0}^{[1]}=\theta_j^{[1]}$, $x_0(k)=1, W^{[1]}$ is a $(h\times n+1)$ synaptic weights matrix. $x(k)=(1,x_1(k),...,x_n(k))^{\mathrm{T}}$. In this case the layer includes $h(n+1)$ tuned parameters.

As activation functions we can use either logistic function

$$\psi_j^{[1]}(u_j^{[1]})=\frac{1}{1-e^{-\gamma_j u_j^{[1]}}} \tag{8}$$

or hyperbolic tangent

$$\psi_j^{[1]}(u_j^{[1]})=\tanh(\gamma_j u_j^{[1]})=\frac{1-e^{-2\gamma_j u_j^{[1]}}}{1+e^{-2\gamma_j u_j^{[1]}}} \tag{9}$$

where γ_j is a positive parameter (possibly tuning one) which defines the «slope» of the function. It is also useful to take into consideration derivatives of logistic function

$$\frac{\partial\psi_j^{[1]}(u_j^{[1]})}{\partial u_j^{[1]}}=\gamma_j\psi_j^{[1]}(u_j^{[1]})(1-\psi_j^{[1]}(u_j^{[1]})) \tag{10}$$

and hyperbolic tangent

$$\frac{\partial\psi_j^{[1]}(u_j^{[1]})}{\partial u_j^{[1]}}=\gamma_j(1-(\psi_j^{[1]}(u_j^{[1]}))^2) \tag{11}$$

accordingly.

Second hidden layer contains $m<n$ neurons as adaptive linear associators, and its output signal $y=(y_1,...,y_m)^{\mathrm{T}}$ is an output of the neural network in whole and it represents «compressed» input image x.

In the absence of bias the second hidden layer is described by relations

$$\begin{cases} y_j(k)=\sum_{i=1}^{h} w_{ji}^{[2]}o_i^{[1]}(k),\ j=1,2,...,m, \\ y(k)=\mathrm{W}^{[2]}o^{[1]}(k) \end{cases} \tag{12}$$

where $\mathrm{W}^{[2]}=\{w_{ji}^{[2]}\}$ is a $(m\times h)$ synaptic weights matrix.

If the second hidden layer neurons include the threshold signal, the stated above relations take form

$$y_j(k) = \sum_{i=1}^{h} w_{ji}^{[2]} o_i^{[1]}(k) + o_j^{[2]} = \sum_{i=0}^{h} w_{ji}^{[2]} o_i^{[1]}(k)\,,$$
$$y(k) = \mathrm{W}^{[2]} o^{[1]}(k), \tag{13}$$

where $\mathrm{W}^{[2]} = \{w_{ji}^{[2]}\}$ is a $(m\times(h+1))$ synaptic weights matrix, $o^{[1]}(k) = (1, o_1^{[1]}(k),\ldots,o_h^{[1]}(k))^{\mathrm{T}}$.

The third hidden layer is similar to the first one and also contains h elementary Rosenblatt perceptrons. If there are no thresholds, this layer is described by relations

$$o_j^{[3]}(k) = \psi_j^{[3]}(u_j^{[3]}(k)) = \psi_j^{[3]}(\sum_{i=1}^{m} w_{ji}^{[3]} y_i(k)),\ j = 1,2,\ldots,h,$$
$$o^{[3]}(k) = \Psi^{[3]}(\mathrm{W}^{[3]} y(k)) \tag{14}$$

where $\mathrm{W}^{[3]}$ is a $(h\times m)$ synaptic weights matrix. If the neurons of third hidden layer include the threshold signal then stated above relations have form

$$o_j^{[3]}(k) = \psi_j^{[3]}(u_j^{[3]}(k)) = \psi_j^{[3]}(\sum_{i=1}^{m} w_{ji}^{[3]} y_i(k) + \theta_j^{[3]}),\ j = 1,2,\ldots,h,$$
$$o^{[3]}(k) = \Psi^{[3]}(\mathrm{W}^{[3]} y(k)) \tag{15}$$

where $\mathrm{W}^{[3]}$ is a $(h\times(m+1))$ synaptic weights matrix.

The fourth output layer is similar to the second hidden layer and contains n neurons as adaptive linear associators. If there is no threshold, the fourth layer is described by relations

$$\begin{cases} \hat{x}_j(k) = \sum_{i=1}^{h} w_{ji}^{[4]} o_i^{[3]}(k),\ j = 1,2,\ldots,n, \\ \hat{x}(k) = \mathrm{W}^{[4]} o^{[3]}(k) \end{cases} \tag{16}$$

where $\hat{x}(k)$ is a $(n\times 1)$ vector which is also the estimate of input signal $x(k)$, restored after compression, $\mathrm{W}^{[4]}$ is $(n\times h)$ matrix of synaptic weights.

If adaptive linear associators of the forth layer include threshold signal, then stated above relations take form

$$\hat{x}_j(k) = \sum_{i=0}^{h} w_{ji}^{[4]} o_i^{[3]}(k) + \theta_j^{[4]} = \sum_{i=1}^{h} w_{ji}^{[4]} o_i^{[3]}(k), o_0^{[3]}(k) = 1,\ \hat{x}(k) = \mathrm{W}^{[4]} o^{[3]}(k) \tag{17}$$

where $\mathrm{W}^{[4]}$ is a $(n\times(h+1))$ synaptic weights matrix.

Thus the mapping formed by the four-layers auto-associative neural network has the form

$$\hat{x}(k) = \mathrm{W}^{[4]}(\Psi^{[3]}(\mathrm{W}^{[3]}\mathrm{W}^{[2]}\Psi^{[1]}(\mathrm{W}^{[1]}x(k)))). \tag{18}$$

The neural network learning process is based on the gradient minimization of the learning criterion

$$\begin{aligned} \mathrm{E}(k) &= \sum_{j=1}^{n} \mathrm{E}_j(k) = \frac{1}{2}\sum_{j=1}^{n}\left(x_j(k)-\hat{x}_j(k)\right)^2 = \frac{1}{2}\sum_{j=1}^{n} e_j^2(t) = \\ &= \frac{1}{2}\left\|e(k)\right\|^2 = \frac{1}{2}\left\|x(k)-\hat{x}(t)\right\|^2 \end{aligned} \tag{19}$$

and in general case can be written in form

$$w_{ji}^{[s]}(k+1) = w_{ji}^{[s]}(k) - \eta_{(k)}^{[s]} + \frac{\partial \mathrm{E}(k)}{\partial w_{ji}^{[s]}}, \quad s = 1,2,3,4 \tag{20}$$

or in a vector form

$$w_j^{[s]}(k+1) = w_j^{[s]}(t) - \eta^{[s]}(k)\nabla w_j \mathrm{E}(k), \quad s = 1,2,3,4 \tag{21}$$

where $\eta^{[s]}(k)$ are learning rate parameters, defining the speed of learning.

Following the backpropagation error concept and using learning algorithms speed optimization technique introduced in [9], we can write an optimal on speed learning procedure in form

$$\begin{cases} w_{ji}^{[4]}(k+1) = w_j^{[4]}(k) + \dfrac{e_j(k)o^{[3]}(k)}{\left\|o^{[3]}(k)\right\|^2}, \quad j = 1,2,\ldots,n, \\[2ex] w_{ji}^{[3]}(k+1) = w_j^{[3]}(k) + \dfrac{\delta_j^{[3]}(k)y(k)}{\left\|y(k)\right\|^2}, \quad j = 1,2,\ldots,h, \\[2ex] w_{ji}^{[2]}(k+1) = w_j^{[2]}(k) + \dfrac{\delta_j^{[2]}(k)o^{[1]}(k)}{\left\|o^{[1]}(k)\right\|^2}, \quad j = 1,2,\ldots,m, \\[2ex] w_{ji}^{[1]}(k+1) = w_j^{[1]}(k) + \dfrac{\delta_j^{[1]}(k)x(k)}{\left\|x(k)\right\|^2}, \quad j = 1,2,\ldots,h \end{cases} \tag{22}$$

where

$$\delta_j^{[s]}(k) = -\frac{\partial \mathrm{E}(k)}{\partial u_j^{[s]}}, \quad s = 1,2,3 \tag{23}$$

is a local error (δ -error) of the corresponding layer.

Procedure (22) is a generalization of Widrow-Hoff algorithm onto multilayer feed-forward neural networks and provides optimal speed in class of gradient learning methods.

4 Comparison of the Results of Using Proposed Neural Networks for the Image Compression

In order to detect how effective the proposed neural networks are for the image compression it is necessary to carry out a research of the obtained compression ratio for the images of different kinds. The compression ratio by default is a ratio of the original file size to the obtained one. The experimental results for the proposed neural networks are shown in figure 3. The vertexes of the four-layers auto-associative neural network graph are marked with a square, those of two-layers artificial neural network are marked with a triangular. Thus, we can say that for practically all of the images from the experimental set in general the four-layers auto-associative neural network gives a bit better results (except the images 6 and 10, where the two-layers network showed better results). At that it should be noted that the compression ratio was in a range from 3 to 11, though for most of the images from the test set it varied from 6 to 8 (11 images) for the two-layers neural network, and from 7.5 to 9.5 (11 images) for the four-layers one. This dependency got especially strong for the images 14-19.

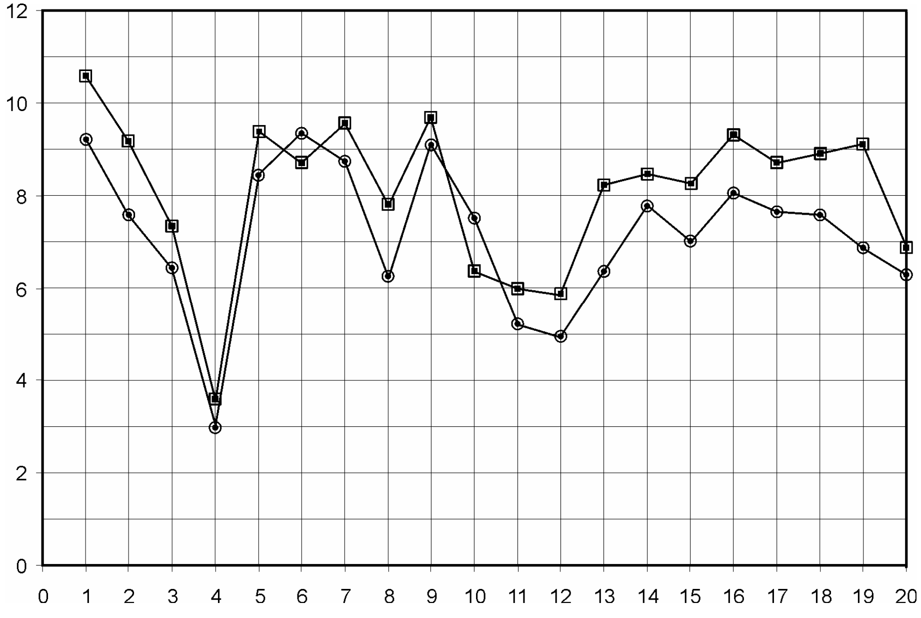


Fig. 3. Images compression ratio for the proposed neural networks

Moreover to analyze the quality of the image compression performed by the given neural networks the criterion based on the pixels values mean square error (MSE)

$$d(x,y)=\sqrt{\sum_{i,j=1}^{n}((x_{i,j}\text{-}y_{i,j})^2\ /\ n^2\ )} \tag{24}$$

was used.

On the base of this criterion we can defined the loss of the information under using each of the networks. In the result we obtain approximately same loss values (in other words approximately same redundancy eliminating) for both networks, and the difference between four and two layers networks did not exceed 1%. At that the loss value was not also critical. Thus, we can make a conclusion as for the possibility of using the given neural networks for reducing the main subspaces dimensionality, i.e. using them for the visual information compression.

5 Conclusions

Two neural networks and their learning algorithms for image compression were considered and analyzed. The results show that the four-layers neural network performs some better results under compression, however in some cases using more simple network gives the same good result. The distinctive feature of the considered neural networks learning processes is their maximal possible speed in the gradient procedures class. Moreover it is necessary to carry further investigation to define classes of images for which these approaches are most effective and also to allocate maximal, minimal and average compression ratios.

References

1. Kohonen, T.: Self-organizing Maps. Springer, Berlin (1995)
2. Dony, R.D., Haykin, S.: Neural network approaches to image compression. Proceedings of IEEE 83, 288–303 (1995)
3. Bishop, C.M.: Neural Network for Pattern Recognition. Clarandon Press, Oxford (1995)
4. Kramer, M.L.: Nonlinear principal component analysis using autoassociative neural networks. AIChE J. 32, 233–243 (2006)
5. Tan, S., Mavrovouniotis, M.: Reducing data dimensionality through optimizing neural-network inputs. AIChE J. 41(6), 1471–1480 (1995)
6. Oja, E., Ogawa, H., Wangviwattana, J.: Principal component analysis by homogeneous neural networks – Part 1: Weghted subspace criterion. IEICE Trans. Inform. Syst. E75-D(3), 366–375 (1992)
7. Miao, Y.F., Hua, Y.B.: Fast subspace tracking and neural networks learning by a novel information criterion. IEEE Trans. on Signal Processing 46, 1962–1979 (1998)
8. Ouyang, S., Bao, Z.: Fast principal component extraction by a weighted information criterion. IEEE Trans. on Signal Processing 50, 1994–(2002)
9. Otto, P., Bodyanskiy, Y., Kolodyazhniy, V.: A new learning algorithm for a forecasting neuro-fuzzy network. Integrated Computer-Aided Engineering 10(4), 399–409 (2003)

Processing Handwritten Words by Intelligent Use of OCR Results

Benjamin Mund and Karl-Heinz Steinke

University of Applied Sciences and Arts, Hanover
Hanover, Germany
karl-heinz.steinke@fh-hannover.de

Abstract. About 3.5 million dried plants on paper sheets are deposited in the Botanical Museum Berlin in Germany. Frequently they have handwritten annotations (see figure 1). So a procedure had to be developed in order to process the handwriting on the sheet. In the present work an approach tries to identify the writer by handwritten words and to read handwritten keywords. Therefore the word is cut out and transformed into a 6-dimensional time series and compared e.g. by means of DTW-method. A recognition rate of 98.6% is achieved with 12 different words (1200 samples). All herbar documents contain several printed tokens which indicate more information about the plant. With the token it is possible to get information who has found this plant, where this plant was found (country and sometimes the town), what kind of plant it is and so on. By using the local connections of the text it is possible to get more information from the herbar document, e.g. to find and recognize handwritten text in a defined area.

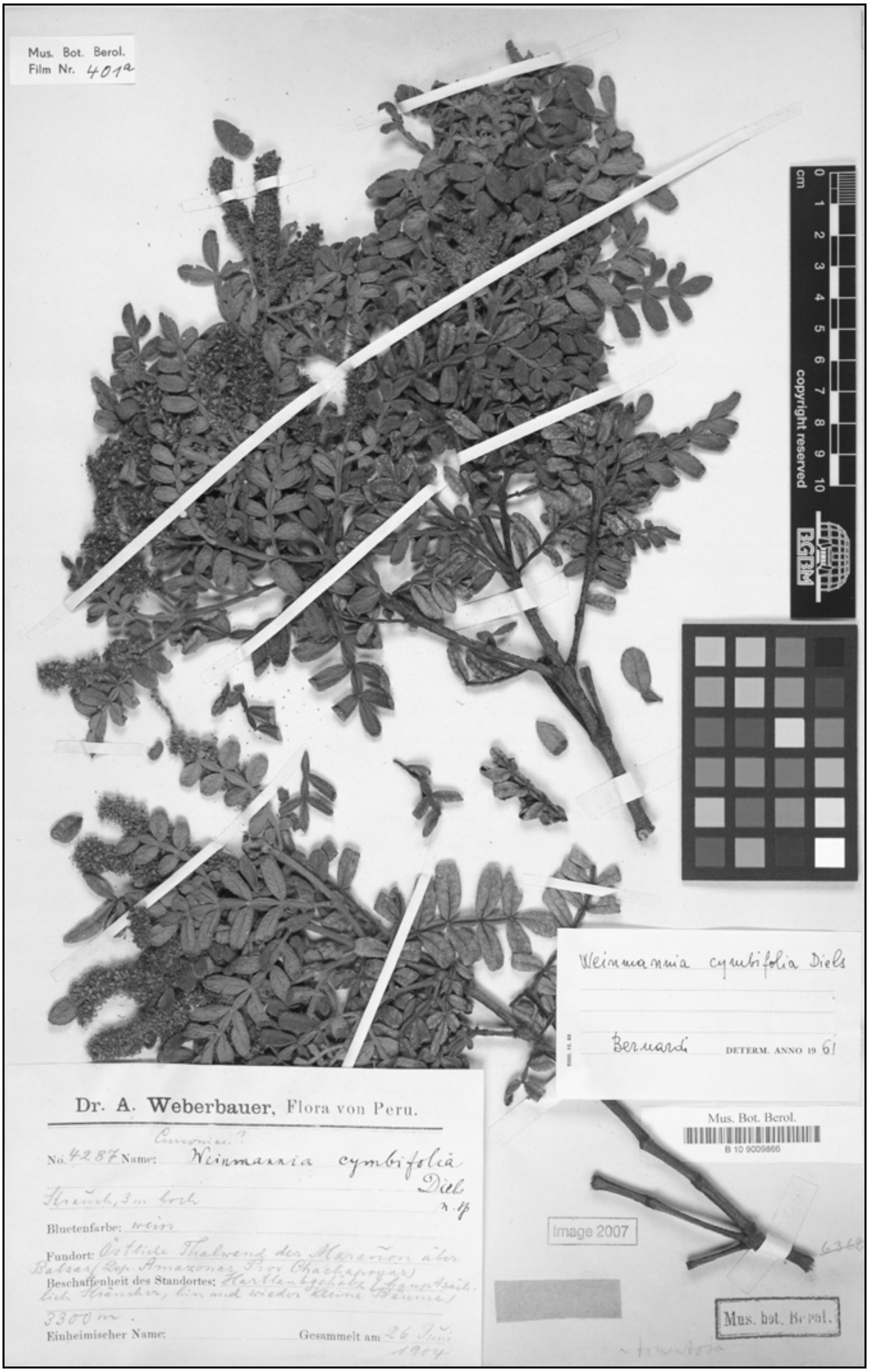
Keywords: local connections, token, handwriting recognition, writer recognition, DTW.

1 Introduction

Many of the approximately 3.5 million herbarium sheets in the Botanical Museum of Berlin have handwritten annotations mixed with printed annotations or labels. The printed text can be processed with OCR software, which provides also the bounding box of recognized text. The final aim is to recognize patterns in the OCR output like family, genus, species, author, variety, location, collection date and annotations. With the local information about e.g. headlines on labels, also the position of handwritten words is provided (see figure 2) when it is located in a predefined region. By localizing the words "*DETERM. ANNO*" the handwritten word "*Bernardi*" can be found in the same writing line.

Because of low quality of our historical documents not all texts have been recognized correctly by the OCR software. Against the promise of the software producer to recognize 99% of the text only about 70% have been localized and recognized correctly. Therefore it is necessary to find a comparison method for similar words. Among the employed algorithms (also used in [16]) there were two which afford very good results. These are on the one hand the Levenshtein Distance and on the other hand the Triplet method.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 174–185, 2010.



Mus. Bot. Berol.
Film Nr. 401ª

Weinmannia cymbifolia Diels

Bernardi DETERM. ANNO 19 61

Mus. Bot. Berol.
B 10 0009866

Image 2007

Mus. bot. Berol.

6368

~ tomentosa

Dr. A. Weberbauer, Flora von Peru.

Cunoniac.?

No. 4287 Name: Weinmannia cymbifolia Diel n. sp.

Strauch, 3 m hoch

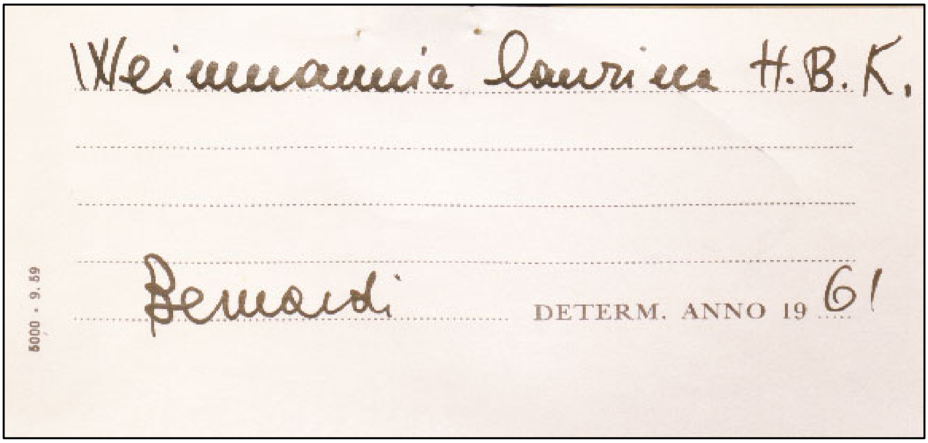
Bluetenfarbe: weiss

Fundort: Östliche Thalwand des Marañon über Balsas (Dep. Amazonas Prov. Chachapoyas)

Beschaffenheit des Standortes: Hartlaubgehölz (hauptsächlich Sträucher, hin und wieder kleine Bäume) 3300 m.

Einheimischer Name: Gesammelt am 26 Juni 1904

Fig. 1. Herbarium document



Weinmannia laurina H.B.K.

Bernardi DETERM. ANNO 1961

5000 · 9.59

Fig. 2. Handwritten word "Bernardi"

All herbar documents contain several printed tokens which indicate the finder of the plant for example. When the herbar documents were checked by the OCR software the coordinates of the text have been saved. By using these coordinates it is possible to search in a defined area around the printed tokens and consider all the texts in this area. With the help of this method it is possible to localize words in block letters or handwritten words in the defined area. It is the aim to get more information from the text fragments like the place, the country and the region in the country where the plant was found, the finder of the plant and so on [17].

In some of the sheets the collector is unknown, because an appropriate note is missing. These sheets should be allocated by the analysis of the handwriting contained in it. Moreover it would be fine to detect some of the handwritten keywords. In handwritten word recognition one of the main difficulties is the high intra-writer and inter-writer variability. Historical documents add another level of complexity resulting from lower quality sources due to various aging and degradation factors, such as faded ink, stained paper, dirt and yellowing. In literature sequential approaches promise an accurate description of word images by 1-D sequences of features vectors. There exist suitable algorithms for efficient treatment of such sequences, like dynamic time warping (DTW) [14] or hidden Markov models (HMM) [11]. Due to the fact that the training sets in our case are of small size it is not suitable to use HMM. DTW promises also success with small training sets. In literature there are also approaches of writer recognition which are mainly based on coordinate sequence data of a digitizer tablet. But in our case we are dealing with old static handwritings of writers who are not present. This so-called off-line writer recognition represents a more complicated problem since no coordinate sequences are available.

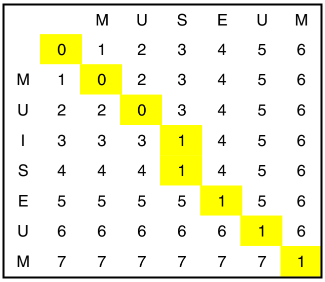
Research in automatic identification of writers focused mainly on the statistical approach. This led to the extraction of characteristics such as run lengths [10] and inclination distributions as well as entropy characteristics. Newer approaches, e.g. that of Siddiqi [12] try to combine global and local features but still with modest success. Niels [5] uses character prototypes and differentiates writers on the basis of how often the prototypes occur in a long text. For this, a time-consuming analysis of the characters has to be made by a handwriting expert. Srihari [6] developed individuality-characteristics for static pictures by extraction of macro and micro features. It was shown that individual characters posses different capabilities of discriminating between writers. Said [11] presents a global approach and regards the handwriting as different textures, which he received by application of the Gabor filters and the co-occurrence matrix. Marti [4] analyzes the difference in handwritings by structural characteristics of each text line. Schomaker [8] uses the contour of connected components. Bensefia [1] uses local characteristics which originate from the analysis of the upper contour's minima.

2 Recognition Methods for Printed Words

At first it was necessary to develop methods to compare two words, which do not exactly match. In the beginning we developed several comparing methods. By testing these methods it became apparent that only two of these could be used: Levenshtein Distance and Triplet Method.

2.1 Levenshtein Distance

This method is used in spell checking and duplicate recognition. It is also possible to use Levenshtein Distance to find similar words. This method compares two character bands. The result of this method is an integer value which shows how many characters have to be changed, removed or added in one character band to get the other.



		M	U	S	E	U	M
	0	1	2	3	4	5	6
M	1	0	2	3	4	5	6
U	2	2	0	3	4	5	6
I	3	3	3	1	4	5	6
S	4	4	4	1	4	5	6
E	5	5	5	5	1	5	6
U	6	6	6	6	6	1	6
M	7	7	7	7	7	7	1

Fig. 3. Example Levenshtein distance

As shown in figure 3, the Levenshtein distance is one, because only the "*i*" from "*Muiseum*" has to be removed to get "*Museum*". A small result is better, because words with a small Levenshtein distance are more similar than word with a big one.

2.2 Triplet Method

This method separates short character bands of the two words which should be checked. To clarify this, an example is shown in figure 4.

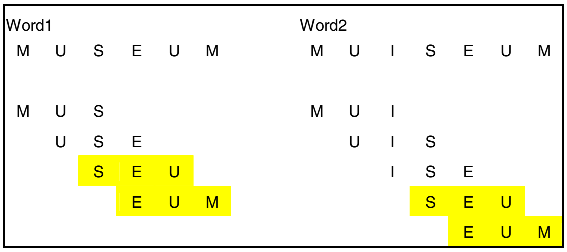
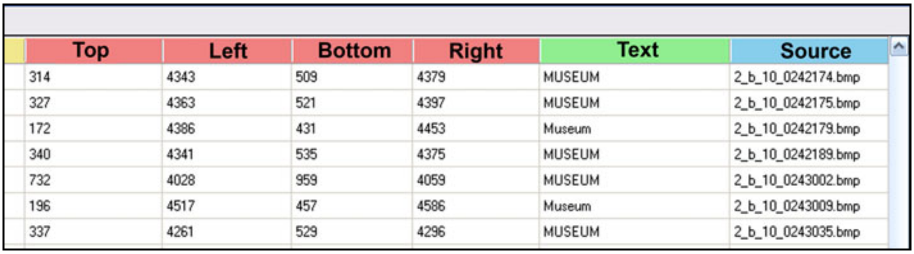


Fig. 4. Example triplet method

Character bands with a length of three characters are created from each word, like shown in figure 4. After that each character band of the first word is compared with each of the second word. The result of this method is an integer value which shows how many character bands occur in both words. To find similar words with this method, the result must be high. This method can be used with different lengths of character bands. For short words it is better to use doublets (length two), because short words do not contain many character bands with length three.

3 Searching Methods

By using the above described methods it is possible to search for words in the Herbar documents. These words can be looked for directly or by using one of these methods, to find similar words. It is also possible to find a text in a defined area around a token by using the local connections of the text.



Top	Left	Bottom	Right	Text	Source
314	4343	509	4379	MUSEUM	2_b_10_0242174.bmp
327	4363	521	4397	MUSEUM	2_b_10_0242175.bmp
172	4386	431	4453	Museum	2_b_10_0242179.bmp
340	4341	535	4375	MUSEUM	2_b_10_0242189.bmp
732	4028	959	4059	MUSEUM	2_b_10_0243002.bmp
196	4517	457	4586	Museum	2_b_10_0243009.bmp
337	4261	529	4296	MUSEUM	2_b_10_0243035.bmp

Fig. 5. Result window exact search for “Museum”

3.1 Exact Search

This search method uses the Levenshtein distance with a result of null. Therefore only the entered word can be found. In figure 5 an extraction of a result window is shown. If a word was not recognized correctly by OCR it will not be found by using this method.

As shown in figure 5, it is only possible to find the word “*Museum”* by using this method.

3.2 Not Exact Search

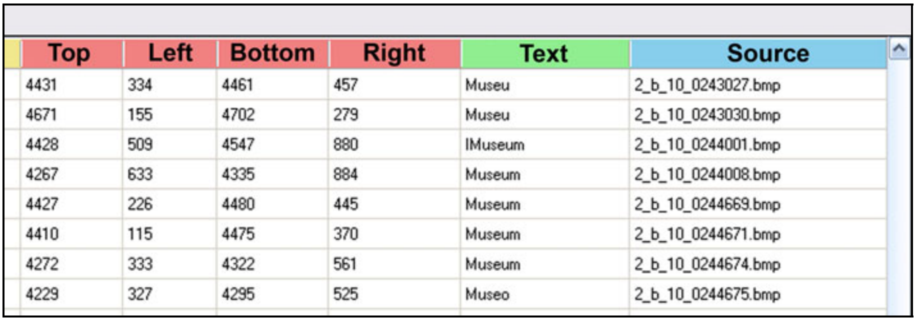
To find similar words to the entered it is necessary to allow e.g. a Levenshtein distance different from null. After entering a word and choosing a recognition method in a graphical user interface it is possible to find similar words to the entered one. By using this method it is possible to find words which were not recognized correctly by OCR. In figure 6 an extraction is shown to demonstrate this method.

For the result in figure 6, the Levenshtein distance was set to a maximum of two. As you can see, not only the word "*Museum*" was found but also "*Museu*", "*Museo*" and "*Imuseum*" were found.

3.3 Searching in Defined Areas

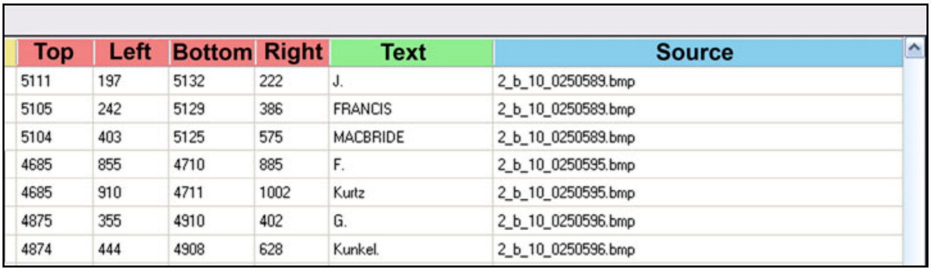
All Herbar documents contain several printed tokens which indicate the finder of the plant for example. So it is possible to search for a token in a Herbar document. If any token was found a previously defined area can be checked if there are more recognized texts. This is possible because for every word its position within the image was saved.

As it can be seen in figure 7, in source "*2_b_10_0250596.bmp*" the words "*G.*" and "*Kunkel*" have been found. In figure 8 an extraction from the corresponding Herbar document is shown.



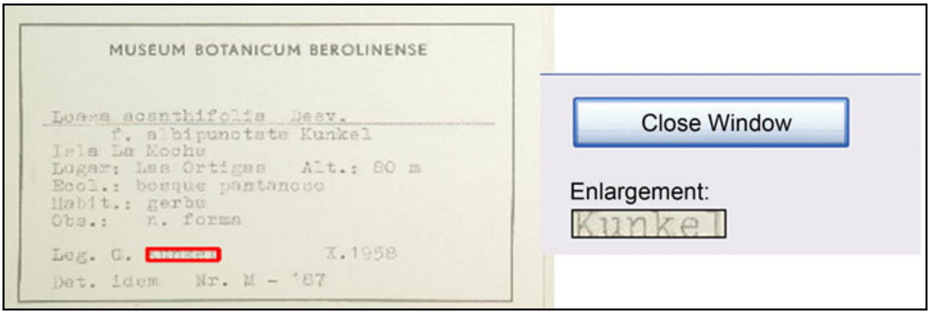
Top	Left	Bottom	Right	Text	Source
4431	334	4461	457	Museu	2_b_10_0243027.bmp
4671	155	4702	279	Museu	2_b_10_0243030.bmp
4428	509	4547	880	IMuseum	2_b_10_0244001.bmp
4267	633	4335	884	Museum	2_b_10_0244008.bmp
4427	226	4480	445	Museum	2_b_10_0244669.bmp
4410	115	4475	370	Museum	2_b_10_0244671.bmp
4272	333	4322	561	Museum	2_b_10_0244674.bmp
4229	327	4295	525	Museo	2_b_10_0244675.bmp

Fig. 6. Result window not exact search



Top	Left	Bottom	Right	Text	Source
5111	197	5132	222	J.	2_b_10_0250589.bmp
5105	242	5129	386	FRANCIS	2_b_10_0250589.bmp
5104	403	5125	575	MACBRIDE	2_b_10_0250589.bmp
4685	855	4710	885	F.	2_b_10_0250595.bmp
4685	910	4711	1002	Kurtz	2_b_10_0250595.bmp
4875	355	4910	402	G.	2_b_10_0250596.bmp
4874	444	4908	628	Kunkel.	2_b_10_0250596.bmp

Fig. 7. Result window category finder



MUSEUM BOTANICUM BEROLINENSE

Loasa acanthifolia Desv.
f. albipunctata Kunkel
Isla La Mocha
Lugar: Las Ortigas Alt.: 80 m
Ecol.: bosque pantanoso
Habit.: gerbu
Obs.: n. forma
Leg. G. Kunkel X.1958
Det. idem Nr. M - '87

Fig. 8. Searched token "Leg." and located text

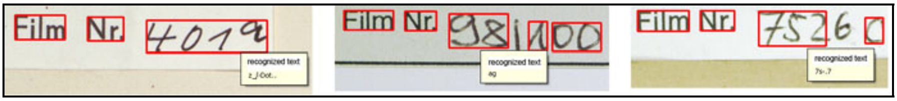
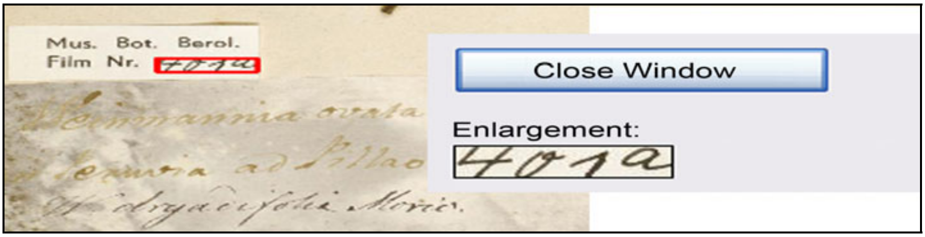


Fig. 9. Found film numbers and recognized text

As you can see in figure 8 the searched token "*Leg.*" was found and the located text in the defined area is shown.

This search method can be used to find block letters or handwriting. Some categories deliver block letters searching for the film number for example. In figure 9 you can see some film numbers and the recognized texts by commercial OCR software.

By using the coordinates it is possible to copy the areas which contain the film number. The part of the picture can be given to a software, which is able to recognize these handwritten numerals. In figure 10 an extraction from a Herbar document is shown in which a film number was found.

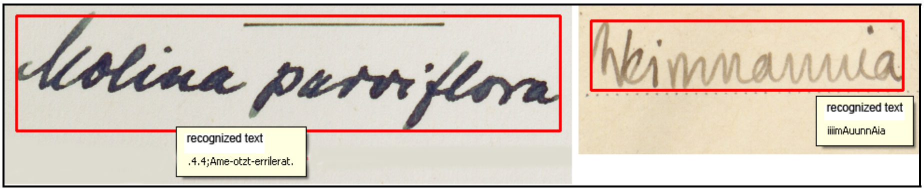


Mus. Bot. Berol.
Film Nr. 4019

Fig. 10. Searched token "Film Nr." and located text

Another possibility of this search method is to find handwriting. With the used commercial OCR software it is not possible to recognize handwritten words. The Herbar documents contain several labels with handwritten text. One of them is the "*Botanical museum*" label. The headline is written by a machine and the remaining text of the label is mostly handwritten. Figure 11 shows some located text with the OCR-recognized words and figure 12 the searched token with the located text.

As you can see in figure 11 no text was recognized correctly by the OCR.



recognized text
.4.4;Ame-otzt-errilerat.

recognized text
iiiimAuunnAia

Fig. 11. Original text and recognized text by OCR

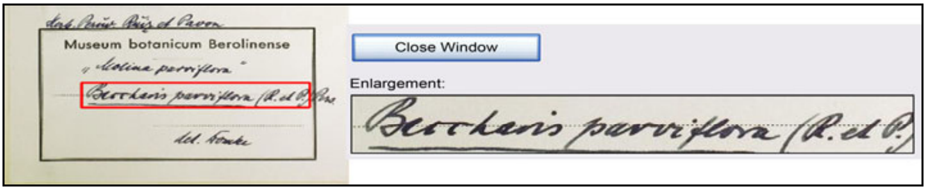


Fig. 12. Searched token “Museum botanicum Berolinense” and located text

4 Handwritten Word Recognition and Writer Recognition

For both, word recognition and writer recognition a normalized word image is the input to the recognizer. For feature extraction a sliding window of one pixel width is moved over the image from left to right. At each position of the window a vector of 6 features is extracted. So each word image is converted into a sequence of 6-dimensional feature vectors (see figure 15) which are afterwards separately discretely approximated. The 6 time series are: upper writing line (yellow), lower writing line

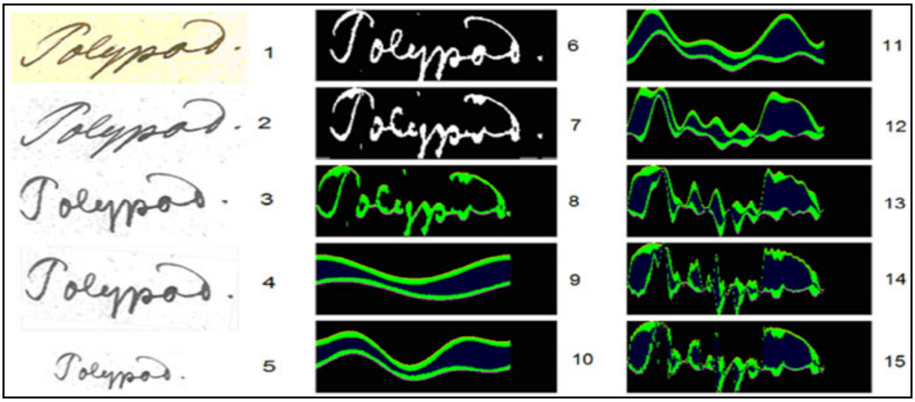


Fig. 13. 1 Original word, 2 Gray level image, 3 Slant correction, 4 Slope correction, 5 Normalization, 6 Binary image, 7 Reconstructed word by time series, 8 Holes removed, 9 Reconstructed by 2 Coefficients, 10 Reconstructed by 4 Coefficients, 11 Reconstructed by 8 Coefficients, 12 Reconstructed by 16 Coefficients, 13 Reconstructed by 32 Coefficients, 14 Reconstructed by 64 Coefficients, 15 Reconstructed by 128 Coefficients

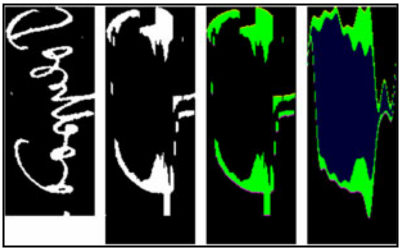


Fig. 14. Mirroring the word along the main axis

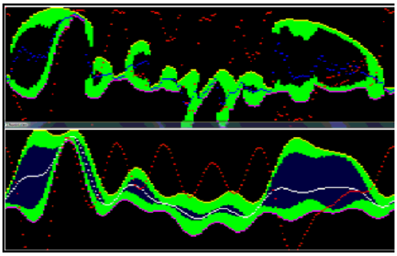


Fig. 15. 6 time series and low pass filtering

(turquoise), mass (green), hollow space (blue), gravity (white) and variance (red). The possible steps are shown in figure 13. Slant and slope correction proved as valuable in word recognition. In writer recognition step 3 and 4 are skipped. After mirroring the word along the main axis (see figure 14) the procedure is repeated.

4.1 Normalization and Feature Compression by Fourier Series

In order to reduce the amount of data and to compare writers by words of varying lengths the approximation by Fourier series was explored.

By a Fourier expansion a repetitive function can be represented as a set of sine and cosine functions, whose frequencies are integral multiples of the basic frequency ω=2π/T.

$$f(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cdot \cos(n\omega t) + b_n \cdot \sin(n\omega t))$$

The Fourier coefficients a_n und b_n can be computed by the Euler formulas:

$$a_n = \frac{2}{T} \int_c^{c+T} f(t)\cos(n\omega t)dt$$

$$b_n = \frac{2}{T} \int_c^{c+T} f(t)\sin(n\omega t)dt$$

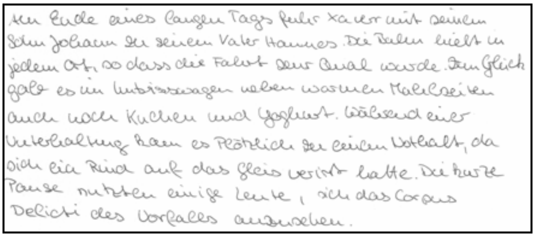
f(t) is approximated by finite trigonometric polynomial $f_n(t)$.

$$f_n(t) = \frac{a_0}{2} + \sum_{k=1}^{n} (a_k \cdot \cos(k\omega t) + b_k \cdot \sin(k\omega t))$$

By the coefficients the word can be back-transformed (see figure 13(9-15)).

4.2 Results Writer Recognition

For testing word recognition and writer recognition by words a multiple occurrence of a word is needed. So a database with a unique text (see figure 16) had to be established.



Am Ende eines langen Tags fuhr Xaver mit seinem
Sohn Johann zu seinem Vater Hannes. Die Bahn hielt in
jedem Ort, so dass die Fahrt zur Qual wurde. Zum Glück
gab es im Imbisswagen neben warmen Mahlzeiten
auch noch Kuchen und Joghurt. Während einer
Unterhaltung kam es Plötzlich zu einem Aufenthalt, da
sich ein Rind auf das Gleis verirrt hatte. Die kurze
Pause nutzten einige Leute, sich das Corpus
Delicti des Vorfalles anzusehen.

Fig. 16. Text written 5 times by 104 writers

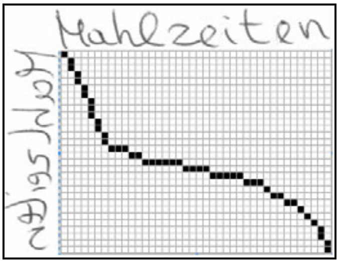
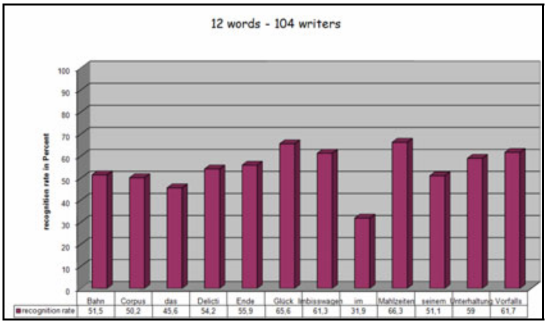
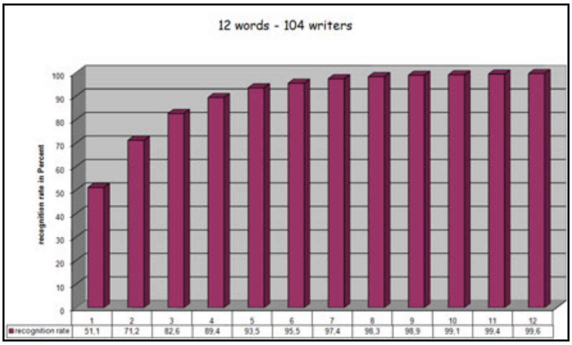


Fig. 17. Dynamic time warping



	Bahn	Corpus	das	Delicti	Ende	Glück	Imbisswagen	im	Mahlzeiten	seinem	Unterhaltung	Vorfalls
■ recognition rate	51,5	50,2	45,6	54,2	55,9	65,6	61,3	31,9	66,3	51,1	59	61,7

Fig. 18. Recognition rates with one word



	1	2	3	4	5	6	7	8	9	10	11	12
■ recognition rate	51,1	71,2	82,6	89,4	93,5	95,5	97,4	98,3	98,9	99,1	99,4	99,6

Fig. 19. Recognition rates with multiple words

For the comparison of a large number of writers 6240 words from 104 writers were extracted from the images. The words posses different discriminatory abilities. In figure 18 the writer recognition rates using only one word is shown when using 16 Fourier coefficients for each of the 6 feature vectors. The 1-D sequences of the word images were also classified directly with dynamic time warping method. DTW is an algorithm for measuring similarity between two sequences which may vary in length. It suites matching sequences with nonlinear warping (see figure 17). The results with DTW are better but the method is very time consuming.

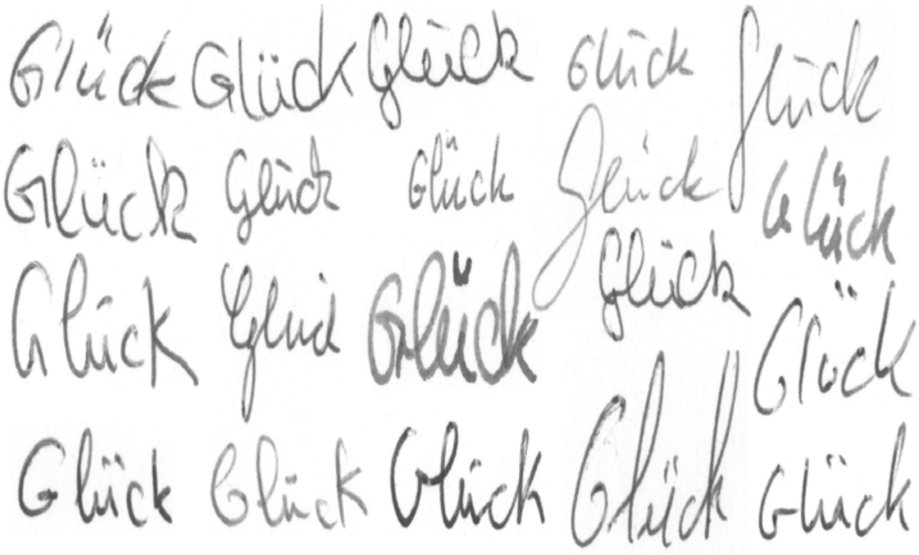


Fig. 20. The word "Glück" of 20 writers

The combination of several words promises a higher recognition rate by more information. It arises in figure 19 that the recognition rate with a higher number of used words increases considerably.

4.3 Results Word Recognition

Because dynamic time warping method is very time consuming only 20 writers (see figure 20) (5 samples of each) and 12 different words (altogether 1200 words) were tested. With 16 Fourier coefficients for each of the 6 feature vectors a word recognition rate of 96% is achieved. With DTW the processing costs are higher but a word recognition rate of 98.6% is achieved.

5 Conclusion

Before processing handwritten words they have to be localized. This is done by use of OCR-programs which provide also the coordinates of recognized text. Especially with printed labels that are filled out by handwriting a priori knowledge helps to find the approximate location of a handwritten word. By the described method the meaning of special key words can be determined with high accuracy. To recognize the writer of a text one word is insufficient. A combination of multiple words leads to high recognition rates.

Acknowledgment

This project is financed by the state of Lower Saxonia and the Volkswagen Foundation.

References

[1] Bensefia, A., Paquet, T., Heutte, L.: A writer identification and verification system. Pattern Recognition Letters 26(13), 2080–2092 (2005)

[2] Rath, T.M., Manmatha, M.: Word Image Matching Using Dynamic Time Warping. In: CVPR 2003, pp. 521–527 (2003)

[3] Marti, U., Bunke, H.: Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. Int. Journal of Pattern Recognition and Artificial Intelligence 15, 65–90 (2001)

[4] Marti, U.V., Messerli, R., Bunke, H.: Writer Identification Using Text Line Based Features. In: Proc. of the 6th International Conference on Document Analysis and Recognition, Seattle, USA, pp. 101–105 (2001)

[5] Niels, R., Grootjen, F., Vuurpijl, L.: Writer identification through information retrieval: the allograph weight vector. In: Proceedings of the 11. Int. Conference on Frontiers in Handwriting Recognition, Montreal (2008)

[6] Srihari, S., Arora, S.H., Lee, S.: Individuality of handwriting. J. of Forensic Sciences 47(4), 1–17 (2002)

[7] Schlapbach, A., Bunke, H.: Off-line Handwriting Identification Using HMM Based Recognizers. Publications Uni Bern (2004)

[8] Schomaker, L., Bulacu, M.: Automatic Writer Identification Using Connected-Component Contours and Edge-Based Features of Uppercase Western Script. IEEE Transactions of Pattern Analysis and Machine Intelligence 26(6), 787–798 (2004)

[9] Steinke, K.-H., Dzido, R., Gehrke, M., Prätel, K.: Feature recognition for herbarium specimens (Herbar-Digital). In: Proceedings of TDWG, Perth (2008)

[10] Steinke, K.-H.: Recognition of Writers by Handwriting Images. In: Duff, M. (ed.) Conference on Pattern Recognition 1981, Oxford (1980)

[11] Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proc. of the IEEE 77, 257–286 (1989)

[12] Siddiqi, I., Vincent, N.: Combining global and local features for writer identification. In: Proceedings of the 11. Int. Conference on Frontiers in Handwriting Recognition, Montreal (2008)

[13] Steinke, K.-H.: Lokalisierung von Schrift in komplexer Umgebung, Tagungsband der Jahrestagung der deutschen Gesellschaft für Photogrammetrie, Jena März (2009)

[14] Sakoe, H., Chiba, S.: Dynamic Programming algorithm optimasation for spoken word recognition. IEEE Transactions on Acoustics, Speech and Signal Processing, 159–165 (1978)

[15] Steinke, K.-H., Gehrke, M., Dzido, R.: Writer Recognition by Combining Local and Global Methods. In: International Congress on Image and Signal Processing, Tianjin China (October 2009)

[16] Heidorn, P.B., Qin, W.Y., Beaman, R., Cellinese, N.: Learning by Example: Machine Learning and Herbarium Label Digitization. In: Joint Plant Science and Conference Botany 2007, Chicago Illinois, July 7-11 (2007)

[17] Mund, B.: Diploma thesis: Datamining in OCR Datenbanken, University of Applied Sciences and Arts, Hanover, Hannover (January 2010)

Saliency-Based Candidate Inspection Region Extraction in Tape Automated Bonding

Martina Dümcke[1] and Hiroki Takahashi[2]

[1] The University of Bremen, Germany
[2] The Department of Human Communication,
The University of Electro-Communications, Japan

Abstract. Electronic circuits are composed of components connected by traces which conduct the current. While the interconnections between the components can be created by assembling individual pieces of wire, it is nowadays common to use printed circuit boards. Tape automated bonding (TAB) is a technique to assemble chips and printed circuit boards. Because TAB become smaller, their inspection methods are required to adapt to the decreasing size of the electric circuits' pattern. An image of a TAB is taken during the manufacturing process and analysed using image processing algorithms to inspect it for flaws in its pattern. This paper proposes an algorithm to find candidate inspection regions in a TAB pattern based on visual saliency. Orientation information contained in the image is processed to detect probable error regions and exclude correct regions from further inspection. The algorithm finds all the flaws in an image and in the case of regular patterns, marks only 5% of the image pixels as belonging to a candidate inspection region. The results show that a saliency-based approach is applicable on the task of finding flaws in the pattern of an electric circuit.

Keywords: defect detection, visual saliency, Tape Automated Bonding.

1 Introduction

Tape Automated Bonding (TAB) is a method to assemble chips and printed circuit boards which are found in electronic devices. Aiming to make devices smaller and thereby more portable, the size of TAB decrease and inspection methods need to be improved to adapt to the smaller pattern size of the electronic circuits.

An image of a TAB is taken to verify its correctness and the pattern is checked via image processing techniques. "Tape automated bonding" means in this case an image of an electric circuit which has been taken during the manufacuring process. Our algorithm detects errors in the pattern of an electric circuit. These errors are short/open, near short/near open and crack/wane illustrated in Fig.1. These flaws cause a suspension in the current flow and must be avoided.

Research on defect detection has been carried out previously, among others on ceramic [1] and fabric [2], [3]. These methods extract frequency information

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 186–196, 2010.

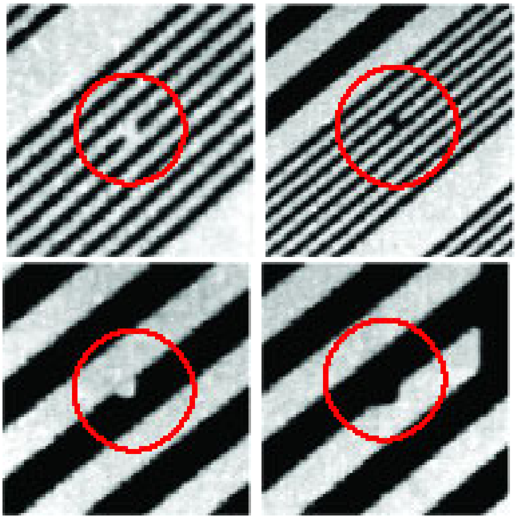


Fig. 1. Possible flaws in the pattern of an electric circuit: open/short (first row from left to right) and crack/wane (second row)

to find mistakes but have the advantage to deal with uniform textures, which is not the case in an image of an electric circuit's pattern. P. Perner [4] combines knowledge-based processing with image processing to detect misprints in offset printing and the system can be applied to other defect detection tasks.

Research on human cognition and its computational simulation started some years ago. Especially the discovery of orientation selective cells in the visual cortex a few decades ago (Hubert & Wiesel, 1959) launched a wave of research aiming to apply this phenomenon to computer systems. Computational neuroscience, a field which simulates the behavior of neurons with computer models, and the field of neuroinformatics, which applies neuronal systems' information processing to technical systems, are two examples. These research fields have a variety of applications, for example in pattern recognition, texture segmentation [5] and also visual saliency.

The simulation of visual saliency of Itti, Koch and Niebur [6] triggered much research in the last years. Saliency maps are based on the visual preprocessing step and show the attention flow. Image parts standing out due to strong changes in intensity or orientation receive the visual attention first and are found by the saliency method.

Our work adapts the saliency method of Itti, Koch and Niebur [6] to find irregularities in a TAB pattern. Using the fact that a TAB is mainly formed of sinusoidal patterns and irregularities hence pop out, visual saliency simulation is suitable for this task.

Our previous work on extracting candidate inspection regions in TAB combines several orientation responses to exclude correct regions. An orientation response using a short wavelength is subtracted from a response obtained using a long wavelength and the remaining regions are labelled as regions that need to be further inspected [7]. The algorithm finds all the flaws but has the drawback to have false positives in patterns with several pattern sizes. This paper explains a new approach using the response with the least changes in orientation and also excludes correct regions in the case of several pattern sizes.

In this paper we will describe the computation of saliency maps on which our research is based in section II, section III explains our approach to solve the task of finding flaws in an electric circuit and sections IV and V contain the experimental data and a conclusion of our research.

2 Saliency Maps

Our research bases on a computational model for visual saliency proposed by L. Itti, C. Koch and E. Niebur [6]. The main features of the saliency-based algorithm will be presented in this section.

Because natural scenes are often overloaded with information, human cognition preprocesses the information in order to extract prominent regions. Hence the attention is shifted to image parts which contrast to the neighboring regions due to a strong change in intensity or orientation. Saliency maps model this preprocessing step and aim to find the regions in an image where the attention is focussed on.

Koch and Ulmann proposed a model of saliency in 1985 [8] and expanded it in 1998 [6]. The results of this research were saliency maps which highlight the regions where the visual stimulus strongly differs from its surrounding. The model stays close to the biological explanation of the visual attention's preprocessing steps and uses among others Gabor filters which respond in a similar way to the behavior of orientation-selective cells in the visual cortex [5]. This model takes into account the blurred border regions of the visual field by using a "center-surround" feature. This "center-surround" mechanism is reached by low-passing the image and thereby reducing its size at each step creating a "Gaussian pyramid". Nine levels of the image are obtained in this way with level 0 being the original image and level 8 being the coarsest scale of the pyramid. Fig. 2 illustrates four of the nine scales of a Gaussian pyramid from the pattern shown in Fig. 6. Three feature maps based on intensity, color and orientation are computed using point-by-point subtraction between a center scale (level 2, 3 or 4 of the pyramid) and a surround scale (level 5, 6, 7 or 8 of the pyramid). An example of an orientation map can be seen on Fig. 4. Intensity, color and orientation are therefore seen as the primary visual features contributing to visual attention's pre-selection. Hence six maps for intensity, six maps for color and 24 maps for orientation are combined into one saliency map where regions with high values pinpoint outstanding image regions.

The model of Itti, Koch and Niebur then implements the attention shift using neural networks but this paper refrains from explaining this procedure as our work omits this step because there is no need to shift between salient regions.

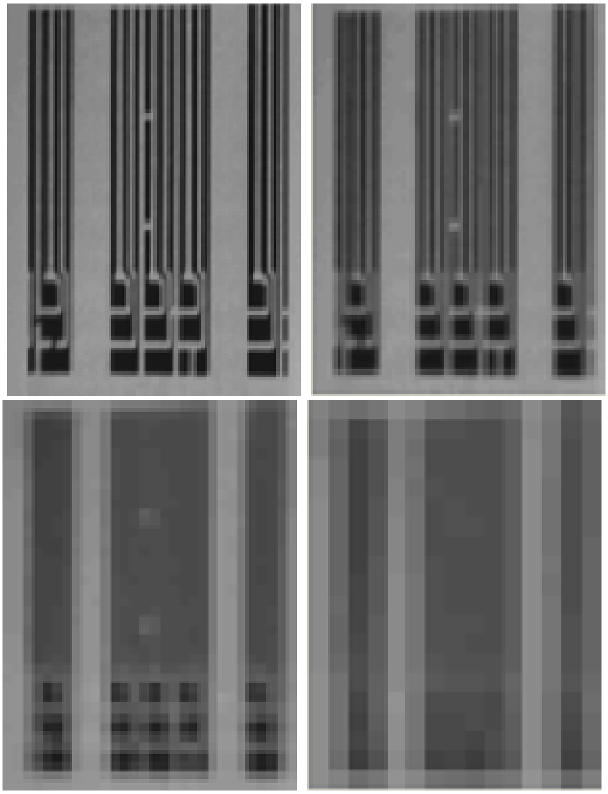


Fig. 2. Layer 3 to 6 of a Gaussian pyramid created from an input image of a TAB (the input image is depicted in Figure 6, the sub images are resized after filtering)

3 Saliency-Based Candidate Inspection Region Extraction

The concept of the saliency-based algorithm presented in this paper bases on two characteristics of defects in TAB: a) flaws are characterized by a change in orientation within the pattern and b) flaws represent only few, irregular changes. In the case of regularly occuring changes, there is a high chance that these variations are not errors but part of the electric cicuit's shape. The method therefore extracts differences in orientation within the image and marks regions where irregular changes occur.

The algorithm first creates a "Gaussian pyramid" from the input image. Gabor filters are then applied to several layers of this pyramid to compute feature maps based on orientation information contained in the image. A threshold is applied to the orientation maps after normalization and the map with the least changes in orientation is selected. Finally a median filter is applied to this map to remove noise.

3.1 Gaussian Pyramid

The first steps of the method are similar to the steps used in Itti, Koch and Niebur [6] to compute saliency maps and will be repeated here for the sake of reference. In the case of our research, the input image is an image of an electric circuit and is filtered several times using a Gaussian filter. This blurs the image and reduces it's size in several steps, creating a "Gaussian pyramid". The input images have a height of 640 pixels, a width of 480 pixels and every sub image of the pyramid is halfed by the filtering process. A pyramid of nine layers is created in this way, where level 0 is the original image and higher levels are more and more blurred. Fig. 2 shows such an image with it's corresponding levels of scale. The information contained in the pyramid represents the human vision: lower levels represent the sharp vision in the center of the visual field whereas upper levels of the pyramid represent the blurred vision in the border region of the visual field. The "center-surround" feature of the human vision, stating that the vision is sharp in the center region of the visual field and blurred in the border regions, is simulated by combining a center level of the pyramid with an upper level. The pyramid is therefore further processed by applying Gabor filter to several layers of the pyramid and combining them using point-by-point subtraction.

3.2 Orientation Maps

Gabor filters simulate the behavior of simple cells in the visual cortex. A Gabor filter is a sinusoidal function combined with a Gaussian function and responds to frequency changes of a wavelength λ along a direction θ.

$$g_{\lambda,\theta,\psi,\sigma,\gamma}(x,y) = \exp\left(-\frac{x_\sigma^2 + \gamma^2 y_\sigma^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x_\sigma}{\lambda} + \psi\right) \tag{1}$$

where x_σ and y_σ are the rotated coordinates defined by

$$\begin{cases} x_\sigma = x\cos\theta + y\sin\theta \\ y_\sigma = -x\sin\theta + y\cos\theta \end{cases} \tag{2}$$

λ is the wavelength, θ marks the direction along which the frequency changes are processed, ψ defines the phase offset of the Gaussian function, σ is the width of the Gaussian envelope and γ tunes the ellipticity of the Gaussian function. x and y are the (x,y)-coordinates in the image.

Orientation maps, here denoted with O, are hence created by filtering a center scale $c \in \{2, 3, 4\}$ of the Gaussian pyramid and a surround scale $s = c + d$, $d \in \{3, 4\}$ with a constant wavelength value in different directions and combining the resulting images using point-by-point subtraction.

$$O(c, s, \theta, \lambda) = |O(c, \theta, \lambda) \ominus O(s, \theta, \lambda)| \tag{3}$$

where $\ominus$ denotes point-by-point subtraction.

The coarser scale image is rescaled to the size of the finer scale image for the subtraction.

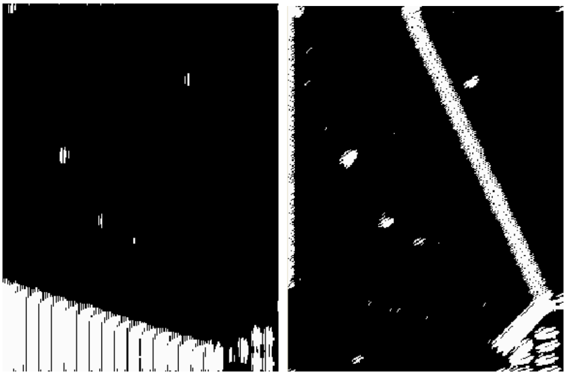


Fig. 3. Orientation maps of the input image in Fig.7 in the directions 0 ° (left) and 45 ° (right)

ψ and γ are set to fixed values: We use asymmetric Gabor filters with ψ equal to 90 and γ is set to 0.5 as suggested in [5]. σ has a value of half the wavelength λ as suggested in [9]. Each Gabor filter is tuned so that it reacts to a regular pattern of a variable wavelength λ in a direction $\theta \in \{0\,°, 45\,°, 90\,°, 135\,°\}$. A center scale $c \in \{2, 3, 4\}$ of the Gaussian pyramid and a surround scale $s = c + d$, $d \in \{3, 4\}$ are filtered with one wavelength in one direction and combined using point-by-point subtraction. Two examples of orientation maps in different directions are illustrated in Fig.3.

3.3 Normalization

The orientation maps are normalized to emphasize responses which are globally weak but strong compared to their neighborhood. The normalization function scales the values of a map to a range [0, 1]. The map from which its minimum has been subtracted is divided by its maximum. The global maximum M and local maxima are then found and an average $\overline{m}$ of the local maxima excluding the global maximum is calculated. The map is normalized by globally multiplying it by $(M - \overline{m})$. Focussing on local maxima enables to promote local changes in orientation instead of a single global peak. Other linear and non-linear normalization functions have been proposed that may be more biologically plausible [10], but this method proved to be a fast and efficient solution for our task.

3.4 Thresholding

A binary threshold is applied to the maps after normalization and sets the map's pixel values that are higher than the threshold to 1 and the other values to 0. The threshold value is computed for each map individually as $\frac{sum\ of\ pixel\ value}{width \cdot height} + offset$, where the $offset$ is a small value enabling to get only high responses of the map. For our simulation, we use an offset of 0.05. An orientation map before and after a threshold was applied is depicted in Fig.4. This step is performed in order to have similar values when the maps are compared.

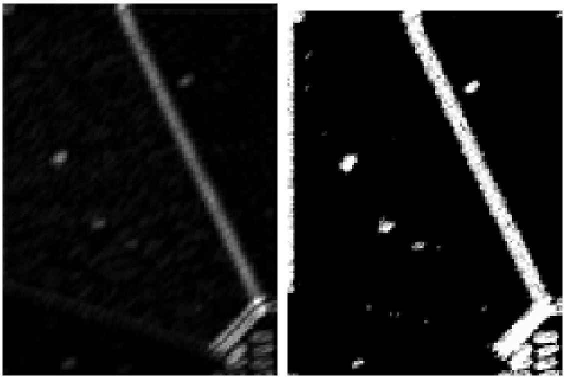


Fig. 4. Orientation map before (left) and after (right) thresholding: weak responses to the filter are suppressed

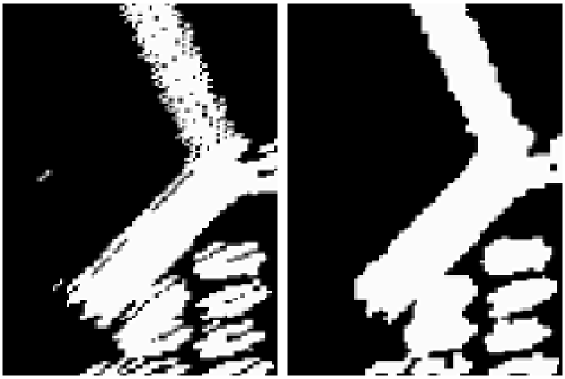


Fig. 5. Error regions before (left) and after (right) applying a median filter: one-pixel-regions are eliminated

3.5 Map Selection and Elimination of One-Pixel Regions

Because the result of the normalization step are maps with pixel value 1 when the Gabor filter strongly responded to the pattern and 0 elsewhere, the sum of the pixel values reveals the amount of pixels that strongly responded to the filter. Furthermore, the map with the smallest sum is the map with the direction that responded the least to the filter. The map with the smallest sum of all the orientation maps is thus chosen and pinpoints the regions that need to be further inspected.

To eliminate one-pixel regions and "close" the found probable error regions, a median filter is applied to the resulting orientation map (illustrated in Fig. 5). Median filters are often used for noise-removal because they sort the values of a pixel-neighborhood in an array and set the center pixel's value to the median value of this array. For our algorithm a 3×3 median filter is sufficient. Filtering an image with a filter of this kind eliminates one-pixel wide regions. In our case, this process sets pixels that are not recognized as part of error regions but are surrounded by recognized pixels to error region pixels. On the same time, single pixels that have been mistaken as probable error regions are eliminated.

4 Results

The algorithm has been applied on test images with patterns containing the flaws open/short, near open/near short and crack/wane pictured in Fig.1. The test images can be categorized into images taken from electric circuits with big pattern size shown in Fig.6, small pattern size illustrated in Fig.7 and mixed

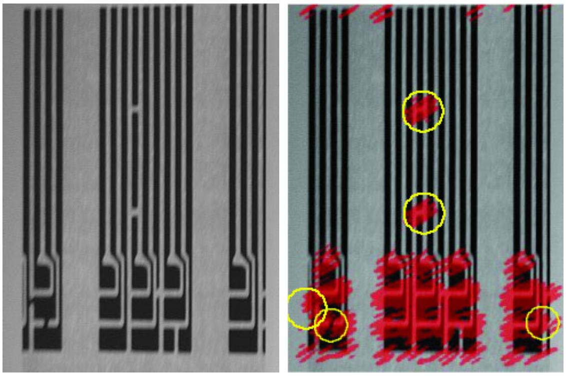


Fig. 6. Original image and result of the algorithm applied on a big sized pattern. The resulting map is overlayed over the original image and candidate inspection regions are colored. 35% of the image's pixels are marked as candidate inspection regions. The encircled regions are true flaw regions.

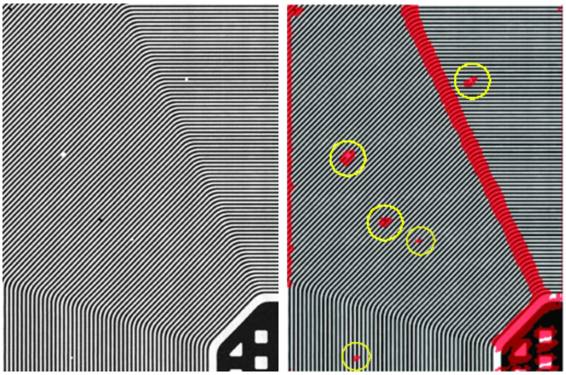


Fig. 7. Original image and result of the algorithm applied on a small sized pattern. 9% of the image's pixels are marked as probable error regions.

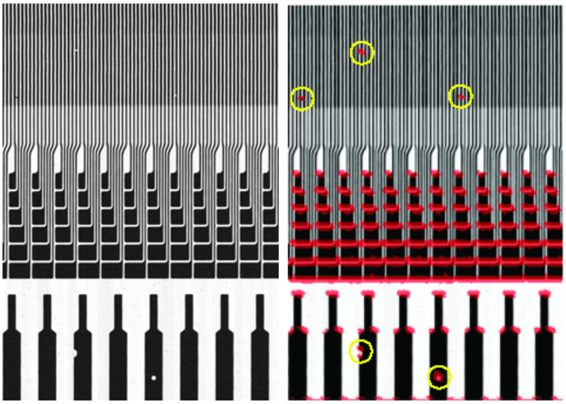


Fig. 8. Original image and result of the algorithm applied on a TAB with several pattern sizes. 12.68% of the image's pixels are marked as belonging to a defect region.

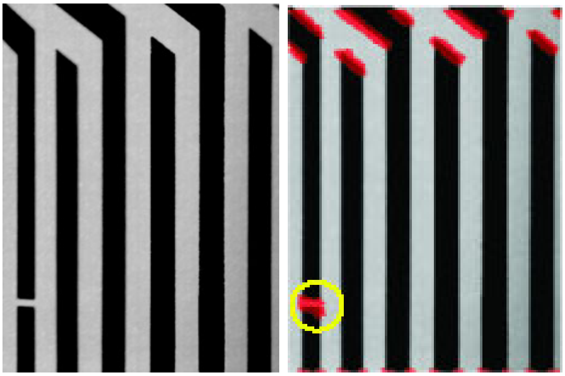


Fig. 9. Original image and result of the algorithm applied on a pattern presenting few changes. 4.67% of pixels from the image are labeled as region that need to be further inspected.

pattern size. The electric circuit presents both big and small pattern sizes, depicted in Fig.8.

Flaws are found and few correct regions are marked as probable error regions. We measured the algorithm's performance by verifying whether all the defects are found and by the number of pixels marked for further processing. The amount of image pixels labeled as candidate inspection regions varies between 6.76% as in Fig.9 and 35% as shown in Fig.6 of the image's pixels for different test images. An example of such a pattern is shown in Fig.9. The average percentages of pixels detected as belonging to candidate inspection regions with a wavelength value λ = 5 are listed in Table 1. Especially electric circuits with limited changes in the pattern's shape make it possible to get few false-positives.

Table 1. Percentage of pixels belonging to candidate inspection regions with $\lambda = 5$

Pattern type	percentage of pixels marked as candidate inspection region
Big pattern	20.84%
Small pattern	7.83%
Mixed pattern	14.72%

Table 2. Percentage of pixels belonging to candidate inspection regions for different wavelengths

Wavelength	$\lambda = 1$	$\lambda = 5$	$\lambda = 10$	$\lambda = 15$
Big pattern	**32.18**	**20.84**	**14.66**	**15.5**
Small pattern	**10.64**	**7.83**	5.56	7.16
Mixed pattern	**22.49**	**14.72**	18.99	19.32

A wavelength value of 5 proved to be a good value for the Gabor filter. As the algorithm focuses on orientation changes, the wavelength value may vary within a small range without having an impact on the result, nonetheless too short or too long a wavelenght will not find all the flaws in a TAB pattern. Table 2 shows the results for different wavelengths. The values are formated in bold type face in the case of no false negative.

Because the orientation maps are computed independently, the maps can be processed in parallel. A parallel computation decreases the computation time of the algorithm and the method performs in real time so that it can be used during the manufacturing process.

The algorithm uses orientation maps in the direction 0 °, 45 °, 90 ° and 135 °. Using more directions augments the complexitiy but does not notably improve the results. Filtering the image in four directions is sufficient for the task of finding flaws in a TAB pattern.

Because the algorithm extracts regions where changes in the pattern's shape occur, variations belonging to the TAB shape may also be labeled as potential error region and a top-down supervision would be needed to eliminate these regions from further inspection as well.

5 Conclusion and Future Work

We have devised an algorithm to extract candidate inspection regions in the pattern of a TAB using image processing techniques based on the saliency method of Itti, Koch and Niebur. Our algorithm processes the orientation information contained in an image of an electric circuit in order to find potential flaws in the pattern and exclude correct regions, keeping the amount and size of candidate inspection regions low. The algorithm builds on the characteristics that flaws are a few changes in orientation within the image.

Test results based on TAB images with different pattern sizes have been presented and the results have been discussed. The algorithm suceeds in finding flaws in a TAB and depending on the pattern's shape, the amount of false positives is low. In the case of regular patterns, only 5% of the image's pixels is marked whereas in the case of more complex patterns the amount of image's pixels labeled as belonging to a potential error region remains less than 40%. These results reveal that the concept of visual saliency is indeed applicable to the task of finding flaws in TAB.

Using parallel computation, the algorithm can be performed at a speed that makes it possible to be used during the manufacturing process. The method helps to detect defects early in the creation process and thus helps to reduce costs.

Similar to many algorithms in the image processing field, our method depends strongly on the values chosen for the parameters. In our case the results depend mainly on carefully chosen wavelengths used for the Gabor filters. Future research will be made to find a suitable value automatically.

References

1. Boukouvalas, C., Kittler, J., Marik, R., Mirmehdi, M., Petrou, M.: Ceramic Tile Inspection for Colour and Structural Defects. In: Proceedings of AMPT 1995, pp. 390–399 (1995)
2. Chan, C., Pang, H.: Fabric Defect Detection by Fourier Analysis. IEEE Transactions on industry applications 36(5) (2002)
3. Kumar, A., Pang, K.: Defect Detection in Textured Materials Using Gabor Filters. IEEE Transactions on industry applications 38(2) (2002)
4. Perner, P.: Knowledge-Based Image Inspection System for Automatic Defect Recognition, Classification and Process Diagnosis. Machine Vision and Applications 7, 135–147 (1994)
5. Petkov, N., Kruizinga, P.: Computational Model of Visual Neurons Specialized in Detection of Periodic and Aperiodic Oriented Visual Stimuli: Bar and Grating Cells. Biological Cybernetics 76 (1997)
6. Itti, L., Koch, C., Niebur, E.: A Model of Saliency-based Visual Attention for Rapid Scene Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(11) (1998)
7. Dumcke, M., Takakura, A., Ali Akbari, M., Takahashi, H.: Saliency-based Algorithm for Extracting Candidate Inspection Regions in Tape Automated Bonding. In: NICOGRAPH International proceedings (2009)
8. Koch, L., Ulmann, S.: Shifts in selective visual attention: towards the underlying neural circuitry. In: Human Neurobiology. Springer, Heidelberg (1985)
9. Ma, H., Doermann, D.: Font Identification Using the Grating Cell Texture Operator. SPIE proceedings series, vol. 5676, pp. 148–156 (2004)
10. Itti, L., Koch, C.: Feature combination strategies for saliency-based visual attention systems. Journal of Electronic Imaging 10(1), 161–169 (2001)
11. Itti, L., Koch, C.: Computational Modelling of Visual Attention. Nature Reviews Neuroscience 2 (2001)

Image Classification Using Histograms and Time Series Analysis: A Study of Age-Related Macular Degeneration Screening in Retinal Image Data

Mohd Hanafi Ahmad Hijazi[1], Frans Coenen[1], and Yalin Zheng[2]

[1] Department of Computer Science, University of Liverpool, Liverpool, L69 3BX UK
[2] Ophthalmology Research Unit, School of Clinical Sciences,
University of Liverpool, Liverpool, L69 3GA UK
{m.ahmad-hijazi,coenen,yalin.zheng}@liverpool.ac.uk
http://www.csc.liv.ac.uk

Abstract. An approach to image mining is described that combines a histogram based representation with a time series analysis technique. More specifically a Dynamic Time Warping (DTW) approach is applied to histogram represented image sets that have been enhanced using CLAHE and noise removal. The focus of the work is the screening (classification) of retinal image sets to identify age-related macular degeneration (AMD). Results are reported from experiments conducted to compare different image enhancement techniques, combination of two different histograms for image classification, and different histogram based approaches. The experiments demonstrated that: the image enhancement techniques produce improved results, the usage of two histograms improved the classifier performance, and that the proposed DTW procedure out-performs other histogram based techniques in terms of classification accuracy.

Keywords: Image mining, Medical image mining, Dynamic time warping, Image classification, Histogram based classification.

1 Introduction

There is much current interest within the data mining community in image mining [1,2], especially medical image mining [3,4,5]. This paper describes a histogram based approach to medical image mining, whereby the histograms are conceptualised as time series, and consequently time series analysis techniques may be applied to classify the images. The focus of the paper is the screening of colour fundus images to identify the possibility of a condition known as age-related macular degeneration (AMD) [6]. However, the described approach has much more general applicability.

Image mining can be undertaken in a number of ways, each requiring a different style of data input. The simplest approach is to express the image set as a set of tabular records, where each column (attribute) represents some image attribute, and then apply established tabular data mining techniques [2,5].

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 197–209, 2010.

The issue here is the identification of the most appropriate image attributes [3,5,7,4,14]. Alternative representations encode the images as graphs where the nodes represent blocks of pixels of similar colour and the edges some relationship between the blocks [3]. In this case graph mining techniques can be applied. Another representation, and that advocated in this paper, is the histogram based approach whereby RGB (red, green and blue) and HSI (hue, saturation, intensity) values are represented as histograms.

In this paper histograms are conceptualised in terms of time series. By considering images in this manner time series analysis techniques may be applied. This paper advocates the use of Dynamic Time Warping (DTW) [13,8], although other time series techniques may be used.

For histogram based medical image mining techniques (and many other mining techniques) to perform well it is desirable to first remove "noise" from the image set. In the case of our retina image set, the focus of this paper, we are interested in removing blood vessels from the image. The successful removal of noise is often helped by first applying some image enhancement process to the input data.

This paper is organised as follows: further details of the AMD background is presented in Section 2. Section 3 reports on some previous works on histogram based approaches, gives a brief review of popular image enhancement methods, and explanations of some general background concerning time series analysis, as applied to image mining, and DTW in particular. In Section 4 image enhancement is considered in the context of AMD; the results from a series of enhancement experiments are reported. The proposed AMD retina image screening process is outlined in Section 5, which includes details of our approach to: (i) noise reduction in the input image set and (ii) DTW. The experimental setup of the screening process, using real data, is reported in Section 6. The results are evaluated and discussed in Section 7. A summary, main findings and some conclusions are presented in Section 8.

2 Age-Related Macular Degeneration

The macula is a small area located at the very centre of the retina, as shown in Figure 1(a) (indicated by a dashed circle). Dominated by cone photoreceptors, the macula allow people to see fine detail as well as colours. Sometimes the delicate cells of the macula become damaged and stop functioning. This leads to a number of eye disorders including age-related macular degeneration (AMD), where the macula degenerates with age [6]. AMD is often diagnosed, at its early stage, by the identification of drusen (yellowish-white subretinal deposits), through screening of patient retinal images. Drusen is usually the first clinical indicator of AMD. The severity of AMD is categorised as being either: *early*, *intermediate*, *advanced non-neovascular*, or *advanced neovascular* [6]. Each category is characterised by the existence of various sizes and shapes of drusen, pigment abnormality and/or other lesions. An example of a retina image that features drusen is given in Figure 1(b) (indicated by the white arrow). Drusen

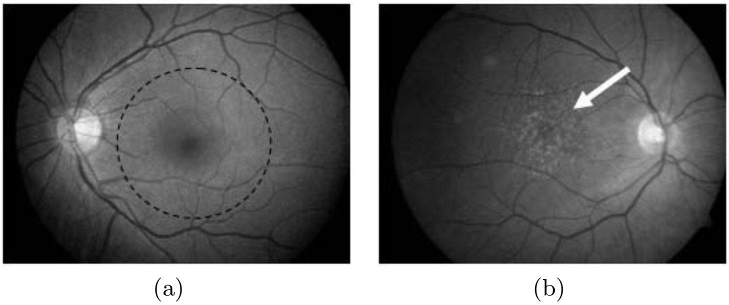


Fig. 1. (a) Normal and (b) AMD retinal images in greyscale

itself are categorised as *hard* and *soft* drusen. Hard drusen have a well defined border, while soft drusen have boundaries that often blend into the background. The identification of features of AMD is thus not a straightforward process [9], and consequently hampers automated diagnosis of AMD at an early stage.

3 Previous Work

In this section an overview of relevant previous work is presented. The section is divided into three sub-sections. Sub-section 3.1 deals with histogram based approaches to image mining and especially medical image mining. Sub-section 3.2 deals with time series analysis, and especially DTW, in the context of medical image analysis. Sub-section 3.3 gives a brief overview of current image enhancement techniques.

3.1 Histogram Based Image Mining

Mining images according to content, in particular the image's colour distribution, is common in image mining [10,11,12,14]. The basic idea is to extract relevant information from the images (colour, texture, etc.) as feature vectors; which can then be represented in other forms, such as histograms, tables or graphs. Histograms tend to be used to signify colour and saturation information. Similarity measures may be used to measure "distance" between images [11,12] for image retrieval, categorisation and classification.

Various similarity measures have been applied to histogram based image classification and retrieval techniques, and each has been empirically measured [15,11]. Earth Mover's Distance (EMD) [16] computes the distance between two distributions, which are represented by user defined signatures. The minimum amount of work needed to transform one distribution into the other is used to measure how similar those distributions are. EMD has been reported to perform well for small sample sizes. Manhattan distance (L_1) has yielded the most effective common dissimilarity measures for histogram retrieval [15]. Euclidean Distance (L_2) is the most common metric used for calculating distance and is used in this paper as the base metric for performance evaluation.

3.2 Time Series Analysis for Image Mining

There is very little reported work on time series analysis for image mining. In [10] colour distribution was represented as a time series for image classification and clustering. Using a time series data representation, called Symbolic Aggregate approXimation (SAX) [17], and the K-nearest neighbour technique for classification. The results in [10] demonstrated a promising approach to time series analysis for image mining. The distinctions between this approach and that presented in this paper are in the time series data representation (SAX represents time series as a sequence of symbols) and the similarity measures adopted.

DTW is a technique for measuring the similarity between two time series. It has been most commonly used in time series analysis [13,8], but can also be applied to other domains [18]. DTW uses a dynamic programming approach to align two time series and then generates a *warping path* that maps (aligns) the two sequences onto each other. To map two time series T and S, of length n and m respectively, where $T = t_1, t_2, ..., t_n$ and $S = s_1, s_2, ...s_m$ a n-by-m matrix will be formed, where the (i^{th}, j^{th}) grid point corresponds to the alignment or distance between two points t_i and s_j. The warping path, W, is then the set of matrix elements that defines a mapping between T and S, defined as $W = w_1, w_2, ...w_K$, where $max(m, n) \leq K < m + n - 1$. The distance $d(t_i, s_j)$ between two points t_i and s_j is used to identify potential warping paths. There are many distance measure that may be used, the most common is the Euclidean distance, and this is the measure used in this paper. Thus:

$$d(t_i, s_j) = w_k = (t_i - s_j)^2 \tag{1}$$

The minimal warping path is selected by calculating the minimum cumulated distance between T and S as:

$$DTW(T, S) = min \left[\sqrt{\sum_{k=1}^{K} w_k} \right] \tag{2}$$

Figure 2(a) shows an example of the comparison of two time series, the generated warping path is given in Figure 2(b). A perfect match between the two time series will produce a direct diagonal line between the two corners of the grid.

3.3 Image Enhancement

Image enhancement is an important pre-processing step for most image mining applications. Many such techniques are reported in the literature [19,2,20,21]. The most common enhancement in the context of histograms is Histogram Equalisation (HE), commonly used to enhance the contrast of an image. HE "spreads out" the most frequent intensity values to produced a better distributed histogram. Through this transformation, the contrast of an image is improved globally; unfortunately the enhancement may result in some bright parts of the image being enhanced to the extent that they are "over exposed" and

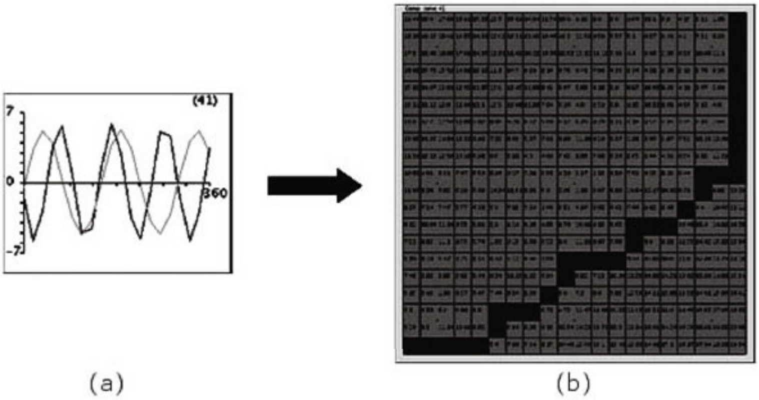


Fig. 2. (a) Two time series for comparison, (b) generated warping path

edges become less distinct. To overcome this problem Contrast Limited Adaptive HE (CLAHE) was introduced [21]. CLAHE computes several histograms that correspond to different sections of an image and equalise each histograms independently. Other enhancement technique exploit the correlation of low frequency coefficients with the illumination variation called Adaptive Histogram Equalisation with Rescaled Low Frequency (AHERLF) [19]. Using AHERLF the image contrast was first stretched using HE, and then transformed into a frequency based representation (coefficients) by means of Discreet Cosine Transform (DCT). The low coefficients, which were directly related to the illumination variations, were then rescaled by dividing by a constant. The approach [19] results in a more uniform illumination image with a better visualisation. A comparison of the above techniques is given in section 4.

An approach to combining the green and red channel histograms for retinal image enhancement has been reported in [20]. The Green channel has the highest contrast between retinal objects (blood vessels, fovea and etc.) and the background, while the red channel is much brighter and thus may improve the visualisation of dark area in the green channel image. In [20] the histogram matching was used to modify the Green channel histogram and consequently produced a histogram that displays the advantages of both channels.

4 Eliminating Noise

The quality of data mining results are typically detrimentally affected by the existence of noise in data [22]. In the case of image mining we typically wish to remove features in the input image set that are not considered relevant. With respect to the AMD screening application it was desirable to “remove” blood vessels from the retina image data. The segmentation of retinal blood vessels was conducted using the 2D matched filters proposed in [23].

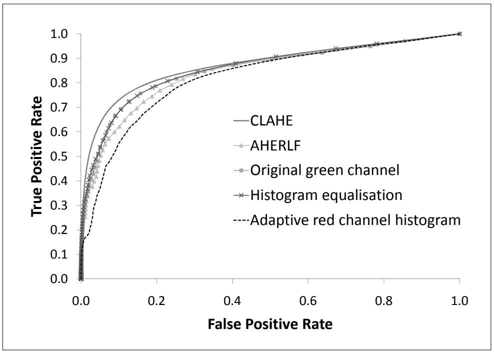


Fig. 3. ROC curves illustrating the performance of the vessel segmentation by using different image enhancement techniques

Table 1. Area under curve (AUC) on vessel segmentation performance calculated for each enhancement technique on green channel

Image enhancement method	**AUC**
CLAHE	0.9180
AHERLF	0.8990
Green channel	0.9117
Histogram equalisation	0.9054
Adaptive red channel histogram	0.8941

The identification and removal of noise can best be facilitated by first enhancing the images. The authors conducted a set of experiments to identify the most appropriate of the enhancement techniques identified above, compared with no enhancement, to support the identification of blood vessels in retina images. For evaluation purpose, the Receiver Operating Characteristic (ROC) curve [24] was used (Figure 3). The area under curve that corresponds to the overall performance on retinal vessel segmentation is shown in Table 1. From the table it can be seen that the best performance was achieved using CLAHE, this technique was therefore adopted to enhance the retina images.

5 AMD Screening

As noted above DTW provides a technique for comparing two curves. This can be fruitfully adapted for the purpose of data classification following the Case Based Reasoning (CBR) [25,26] paradigm. Using this paradigm a new case is classified according to its similarity with a set of known pre-classified cases stored in a *Case Base*. With respect to the AMD screening application described here a set of pre-labelled retina images was used to form the “case base”. New “unseen” images could then be classified according to the “nearest match” within the case base, the class of the most similar case found in the case base being the class of the new case.

The proposed AMD image classification process is outlined in Figure 4. To represent each image, the green channel (RGB colour model) and saturation component (HSI colour model) histograms were extracted. The green channel was selected because it displays maximum contrast [9], and has the best discriminatory power between retinal anatomy and the retinal background [23]. In the context of AMD the green channel has been shown to produce good and consistent classification performance compared to other RGB channels and HSI components [18], particularly with respect to "normal" images. The saturation component was chose due to its good performance (in particular to identify AMD images) in AMD classification as shown in [18].

Prior to the generation of the histograms the images were pre-processed. First, CLAHE image enhancement was applied to each colour image to emphasise the contrast of the image and the "visibility" of retinal blood vessel edges. Next, the segmentation of retinal vessels was conducted using 2D matched filters [23] on the green channel. This was followed by the extraction of green channel and saturation component information, which was then represented in the form of histograms. The retinal blood vessel pixels were subsequently removed by subtracting the vessels intensity value from the generated histograms. The 'cleaned' green channel and saturation component histograms, referred to as curve hereafter, formed the "case base" (C) comprising: green ($G = g_0, g_1, ..., g_I$) and saturation ($S = s_0, s_1, ..., s_I$) curves (where I is the number of images and $G, S \in C$). New images to be classified formed a second "case base", $\bar{C}$, comprising: green ($\bar{G} = \bar{g}_0, \bar{g}_1, ..., \bar{g}_J$ and saturation, $\bar{S} = \bar{s}_0, \bar{s}_1, ..., \bar{s}_J$ curves (where J is the number of images and $\bar{G}, \bar{S} \in \bar{C}$). In the classification stage, each curve in $\bar{G}$ is compared with the content of G using DTW. A list of the n most similar g, $sim(\bar{g})$ is produced for each $\bar{g}$:

$$sim(\bar{g}_j) = \left\{ \left(g_0, \delta_0^j\right), ..., \left(g_n, \delta_n^j\right) \right\} \tag{3}$$

$$\delta_n^j = DTW\left(\bar{g}_j, g_i\right) \tag{4}$$

$DTW(\bar{g}, g)$ is the minimal warping path or distance of the green channel "new case" curve, $\bar{g} \in \bar{G}$ and its most similar green channel "case base" curve, $g \in G$ (equation 2), $0 \leq i < I$ and $0 \leq j < J$. Each $\bar{g} \in \bar{G}$ is then classified according to the class associated with its most similar $g \in G$. The classifier will sort the curves in $sim(\bar{g}_j)$ in ascending order of similarity. Where several $g \in G$ have similar δ values a similar distance measuring process is applied to $\bar{S}$ and S. This is undertaken where the distance between two δ values is less than a user specified threshold, $diff$:

$$diff_j = \delta_0^j \times (1 + \alpha) \tag{5}$$

where α is a predefined constant (set to 0.3 in this paper). This step will produce a list of saturation curves distance, $sim(\bar{s})$, for each $\bar{s} \in \bar{S}$ of length m (m is determined by the number of δ that is less than $diff$):

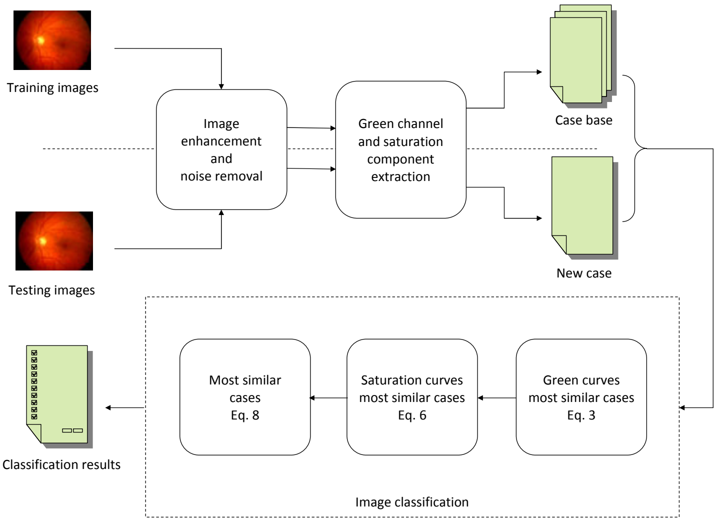

Training images
Image enhancement and noise removal
Green channel and saturation component extraction
Case base
Testing images
New case
Classification results
Most similar cases Eq. 8
Saturation curves most similar cases Eq. 6
Green curves most similar cases Eq. 3
Image classification

Fig. 4. Proposed AMD screening process

$$sim\left(\bar{s}_j\right) = \left\{\left(s_0, \bar{\delta}_0^j\right), ..., \left(s_m, \bar{\delta}_m^j\right)\right\} \tag{6}$$

$$\bar{\delta}_m^j = DTW\left(\bar{s}_j, s_i\right) \tag{7}$$

Finally, we can work out the list of m most similar cases, $c \in C$ for each $\bar{c} \in \bar{C}$ as follows

$$sim\left(\bar{c}_j\right) = \left\{\left(c_0, \mu_0\right), ..., \left(c_m, \mu_m\right)\right\} \tag{8}$$

$$\mu_m = \frac{1}{2}\left(\delta_m^j + \bar{\delta}_m^j\right) \tag{9}$$

Each "new case", $\bar{c} \in \bar{C}$ will then be classified as belong to the same class of its most similar curve in the "case base", $c \in C$.

6 Experimental Setup

In this study 144 hand labelled images were used, of which 86 featured AMD that were collected as part of the ARIA project[1]. Ten-fold Cross Validation (TCV) was applied in all experiments (where one tenth of the images were taken as a testing set and the rest for training in each fold). The aims of the experiments were: (i) to investigate the classification performance, using images enhanced with CLAHE and noise removed, against the raw images; (ii) to evaluate the performance of combining two different histograms, green and saturation histograms, on image classification; and (iii) to investigate how well the DTW approach operated with respect to other histogram based approaches (L_1, L_2, and EMD).

[1] http://www.eyecharity.com/aria_online/

The constant α (equations 5) was set to 0.3 (determined through a series of tests), while the number of most similar green curves n (equations 3) was set to 5. Three evaluation metrics were utilised to measure the classification performance: sensitivity, specificity and accuracy. Overall accuracy was used to measure the overall performance of the classifier in terms of classifying retinal images correctly according to their class.

7 Results and Discussions

In this section the results of the three sets of experiments, introduced above, are presented and discussed.

7.1 Performances of Enhanced Images for AMD Classification

Table 2 shows the effect of image contrast enhancement and noise removal, on the classification result. The proposed AMD screening approach, for each green (equations 3) and saturation (equations 6) curve, was applied separately. From the table it can be seen that the application of noise reduction produced superior results, an average improvement of 4% per evaluation metric compared to the raw images for green channel curves, while saturation curves recorded slight improvements only on both the sensitivity and accuracy.

The pre-processing (image enhancement and noise removal) of the retinal images drastically changed most of the histogram curves. Overall, the curves became smoother and more consistent according to their classes, this was particularly so in the case of the normal control images. A more distinctive pattern could also be observed between classes. Enhancement significantly altered the time series analysis output in all cases. It is suggested that the better performance, resulting from the "smoothed" curves, has contributed to a better DTW.

Figure 5 shows examples of the green histogram curves, and the best matching curve, for three selected images (each column represents an image). Two of the images featured AMD, one did not. Figure 5(a-c) presents the "raw" curves without enhancement. Figure 5(g-i) shows the same curves after the application of the proposed enhancement and noise reduction techniques. From the figure it can be seen that there are significant differences between the two sets of curves.

Table 2. Ten-fold Cross Validation (TCV) results of applying image pre-processing on retinal images for AMD classification

	Original images		**Pre-processed images**	
	Green	Saturation	Green	Saturation
Specificity (%)	57	55	62	55
Sensitivity (%)	69	81	72	83
Accuracy (%)	64	71	68	72

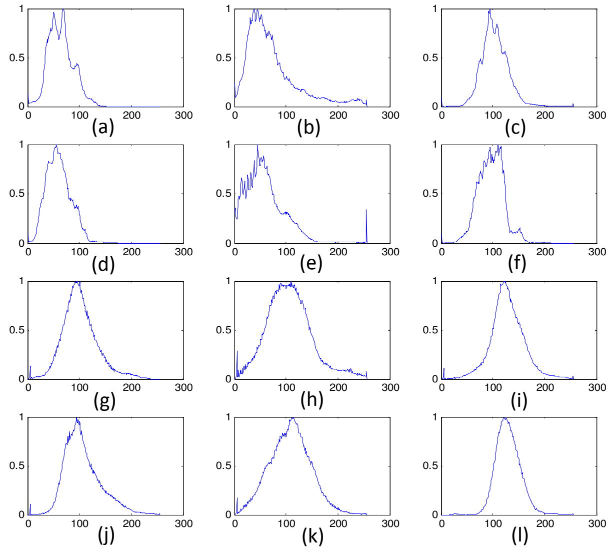


Fig. 5. Sample green channel curves: (a-c) raw image histograms, (d-f) most similar case for curves a to c, (g-i) enhanced and noise removed histograms, (j-l) most similar case for curves g to i

Applying the proposed DTW technique to the raw images given in Figure 5(a-c) resulted in misclassifications in all cases. The closest matches in each case are presented in Figure 5(d-f). After pre-processing, curves (g) and (i) were classified correctly, although curve (h) remained misclassified. Careful analysis of the results suggests that the application of enhancement and noise reduction techniques to the images has a significant effect on the nature of the histogram curves and consequently on the classification accuracy.

7.2 Performances Using Both Green and Saturation Histograms

Both the green and saturation curves produced good performances. However closer inspection of the results revealed that the difference between the best fit and the second best fit was sometimes marginal, thus calling into question the robustness of the approach. It was also noted that in the case of the green channel histograms the distinction between the best fit and the second best fit tended to be larger than in the case of the saturation histograms (but not in all cases). It was therefore deemed appropriate to combine the evidence of both types of histogram (the process described in Section 5) so that a more robust classification would result. Recall from Section 5 that the saturation histogram is only considered where the difference between the top two classes using green channel histograms disagree and the difference in similarity values (δ) is less than $diff$.

Table 3. Ten-fold Cross Validation (TCV) results of using green and saturation curves for AMD classification

	Green	**Saturation**	**Green and saturation**
Specificity (%)	62	55	62
Sensitivity (%)	72	83	79
Accuracy (%)	68	72	72

From Table 3 it can be seen that the use of both sets of histograms (where appropriate) produced better results than when using the green histograms alone; and comparable with the saturation histograms results, but with a much greater degree of "confidence" in the end results.

7.3 Comparison of Performances of Various Histograms Based Image Classification Techniques for AMD Classification

Table 4 presents a comparison of classification accuracy using the proposed DTW technique and L_1, L_2, and EMD to classify retinal images using the pre-processed and original images. TCV was again used throughout. Using DTW' and $L_1^{'}$ with raw images produced a superior specificity compared to others. DTW however performed exceptionally well with respect to sensitivity and overall accuracy (79% and 72% each). The other techniques (L_1, L_2, and EMD), applied to raw and enhanced images, produce mixed results as shown in the Table.

The results presented in Table 4 indicate that the ability of DTW to find dissimilarities between two time series by calculating the shortest path between points in the time series data may have contributed to the results produced. Other techniques that calculate point to point distances between two curves are not good enough to classify complex and non-uniform curves, like does used to represent retinal images in this paper. DTW measures the distance between a point in a time series curve to all points in the other time series curve and selects the shortest distance. From Table 4 the best performance recorded was 79% sensitivity.

Table 4. Ten-fold Cross Validation (TCV) Results using DTW, L1, L2 and EMD for retinal images classification

	DTW	L_1	L_2	**EMD**	**DTW'**	$L_1^{'}$	$L_2^{'}$	**EMD'**
Specificity (%)	62	62	61	55	*64*	*64*	56	52
Sensitivity (%)	*79*	76	74	67	75	76	73	69
Accuracy (%)	*72*	70	69	62	71	71	66	62

8 Conclusions

In this paper, an image classification technique, founded on a histogram based representation and DTW was described. The focus of the work was the classification of retinal image data to provide an AMD screening service. In the case of the AMD application the images were represented in terms of their green channel and saturation component histograms. Experiments were conducted to compare the legitimacy of the technique with respect to: (i) image enhancement and noise reduction, (ii) combination of two different histograms, and (iii) other histogram based techniques. From the experimentation it was found that noise reduction (removal of blood vessels in the case of the AMD data) and image enhancement produced better classification results. The results were improved further by using both the green and saturation histograms. The experiments also indicated that the proposed DTW technique, combined with the proposed enhancement strategy, produced the best classification results compared to other histogram based techniques. The best result achieved by the proposed approach was a sensitivity of 79%. It is worth noting that in other work [27], the mean sensitivity achieved through manual graders observation on different sets of retinal images was 86%. For future works, we intend to investigate further noise removal, as well as focusing the screening process on only the central area of the retinal image (the Macula). We are also aware of the issues of inadequate dataset size, therefore, effort is being made to collect more examples of both AMD and normal images.

References

1. Hsu, W., Lee, M.L., Zhang, J.: Image mining: Trends and developments. Intelligent Information Systems 19, 7–23 (2002)
2. Ordonez, C., Omiecinski, E.R.: Discovering association rules based on image content. In: IEEE Forum on Research and Technology Advances in Digital Libraries, pp. 38–49 (1999)
3. Elsayed, A., Coenen, F., Jiang, C., Garcia-Finana, M., Sluming, V.: Corpus callosum MR image classification. In: Proceedings of AI 2009, pp. 333–346. Springer, London (2009)
4. Perner, P.: Image mining: issues, framework, a generic tool and its application to medical-image diagnosis. Engineering and Applications of Artificial Intelligence 15, 205–216 (2002)
5. Yu, X., Hsu, W., Lee, W.S., Lozano-Peres, T.: Abnormality detection in retinal images (2004)
6. Jager, R.D., Mieler, W.F., Mieler, J.W.: Age-related macular degeneration. The New England Journal of Medicine 358, 2606–2617 (2008)
7. Hsu, W., Lee, M.L., Goh, K.G.: Image mining in IRIS: Integrated retinal information system. ACM SIGMOD Record 29, 593 (2000)
8. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: First SIAM International Conference on Data Mining (2001)
9. Rapantzikos, K., Zervakis, M., Balas, K.: Detection and segmentation of drusen deposits on human retina: Potential in the diagnosis of age-related macular degeneration. Medical Image Analysis 7, 95–108 (2003)

10. Al-Aghbari, Z.: Effective image mining by representing color histograms as time series. Journal of Advanced Computational Intelligence and Intelligent Informatics 13, 109–114 (2009)
11. Conci, A., Castro, E.M.M.: Image mining by content. Expert System with Applications 23, 377–383 (2002)
12. Foschi, P.G., Kolippakkam, D., Liu, H., Mandvikar, A.: Feature extraction for image mining. In: International Workshop on Multimedia Information Systems, pp. 103–109 (2002)
13. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: AAAI Workshop on Knowledge Discovery in Databases, pp. 229–248 (1994)
14. Huiskes, M.J., Pauwels, J.: Indexing, learning and content-based retrieval for special purpose image databases. Advances in Computers 65, 203–258 (2005)
15. Brunelli, R., Mich, O.: Histograms analysis for image retrieval. Pattern Recognition Letters 34, 1625–1637 (2001)
16. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International Journal of Computer Vision 40, 99–121 (2000)
17. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, pp. 2–11. ACM, New York (2003)
18. Hijazi, M.H.A., Coenen, F., Zheng, Y.: A histogram approach for the screening of age-related macular degeneration. In: Medical Image Understanding and Analysis 2009, BMVA, pp. 154–158 (2009)
19. Hossain, M.F., Alsharif, M.R.: Image enhancement based on logarithmic transform coefficient and adaptive histogram equalization. In: International Conference on Convergence Information Technology, pp. 1439–1444. IEEE, Los Alamitos (2007)
20. Salem, N.M., Nandi, A.K.: Novel and adaptive contribution of the red channel in pre-processing of colour fundus images. Journal of the Franklin Institute 344, 243–256 (2007)
21. Zuiderveld, K.: Academic Press Graphics Gems Series. In: Contrast limited adaptive histogram equalization, pp. 474–485. Academic Press Professional, Inc., London (1994)
22. Cios, K., Swiniarski, R., Pedrycz, W., Kurgan, L.: Data, ch. 2, pp. 27–47. Springer, US (2007)
23. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE Transactions on Medical Imaging 8, 263–269 (1989)
24. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874 (2006)
25. Kolodner, J.: Case-based reasoning. Morgan Kaufmann, San Francisco (1993)
26. Perner, P.: Introduction to case-based reasoning for signals and images. In: Case-based reasoning on images and signals, pp. 1–24. Springer, Heidelberg (2008)
27. Jain, S., Hamada, S., Membrey, W.L., Chong, V.: Screening for age-related macular degeneration using nonstereo digital fundus photographs. Eye 20, 471–475 (2006)

Entropic Quadtrees and Mining Mars Craters

Rosanne Vetro and Dan A. Simovici

Univ. of Massachusetts Boston, Dept. of Comp. Science, 100 Morrissey Blvd. Boston,
Massachusetts 02125 USA
{rvetro,dsim}@cs.umb.edu

Abstract. This paper introduces entropic quadtrees, which are structures derived from quadtrees by allowing nodes to split only when nodes point to sufficiently diverse sets of objects. Diversity is evaluated using entropy attached to the histograms of the values of features for sets designated by the nodes.

As an application, we used entropic quadtrees to locate craters on the surface of Mars, represented by circles in digital images.

1 Introduction

In this paper we introduce a variant of quadtrees, a well-known data strucure used in spatial databases. A *quadtree* $\mathcal{T}$ is a tree structure defined on a finite set of nodes that either contains no nodes or is comprised of a root node and 4 quad-subtrees. In a full quadtree, each node is either a leaf or has degree exactly 4. Our variant of quadtrees requires that each node that has descendents is pointing to an area that has a sufficient level of diversity as assessed by the value of an information-theoretical measure.

We provide an algorithm that captures high complexity areas of an image. This algorithm is used for the detection of circular shapes that can possibly correspond to craters.

The algorithm is composed by two methods. The first method uses an information-theoretical approach to create an edge filter that generates a binary image from complex areas which may contain edges. The second method applies a Circle Hough Transform (CHT) with modified threshold to detect the presence of circular shapes in complex areas. The new threshold is imposed to increase the quality of the results given the lack of prior knowledge about the number of craters in an image and the difficulty to estimate a good threshold for the minimum number of votes required in the parameter space to indicate true center points. Efficient methods for crater detection such as [1], [2] and many others referenced by the authors of [3] have been proposed. We provide a distinct approach where no external pre-processing of the original image other than conversion to the JPEG format and resizing is needed. Likewise, no external image filters are used.

In Section 2 we introduce the framework for the rest of the paper. The notion of entropy associated to a partition is presented as well as its usefulness in measuring diversity.

In Section 3 we introduce the proposed algorithm and explain the searching process. In subsection 3.1 we describe the information theoretic method used for mining complex subareas that may contain edges. The CHT method with modified threshold is described in subsection 3.2. Section 4 contains a description of the experiments and major challenges we faced. Finally, Section 5 contains our conclusions and ideas for future work.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 210–221, 2010.

2 Partitions, Entropy, and Trees

Information theory involves the quantification of information and was created with the purpose of finding fundamental limits on compressing, reliably storing, and communicating data. Entropy is a important measure of information in the theory that quantifies the uncertainty associated with probability distributions.

Let S be a finite set. A *partition* on S is a non-empty collection of non-empty subsets of S, $\pi = \{B_1, \ldots, B_n\}$ such that

(i) $B_i \cap B_j = \emptyset$ for $1 \leq i, j \leq n$ and $i \neq j$;
(ii) $\bigcup\{B_i \mid 1 \leq i \leq n\} = S$.

The sets $B_1, \ldots, B_n$ are referred to as the *blocks* of π.

We denote by $\mathsf{Part}(S)$ the set of partitions of S. For $\pi, \sigma \in \mathsf{Part}(S)$ define $\pi \geq \sigma$ if each block B of π is a union of blocks of σ. It is well-known that the relation "$\geq$" is a partial order on $\mathsf{Part}(S)$. The largest partition on S is the single-block partition $\omega_S = \{S\}$, while the smallest partition on S is $\iota_S = \{\{x\} \mid x \in S\}$.

We define now a partial order relation $\geq_k$ on $\mathsf{Part}(S)$ as follows. If $\pi = \{B_1, \ldots, B_n\}$ and $\sigma = \{C_1, \ldots, C_m\}$, then $\pi \geq_k \sigma$ if the following conditions are satisfied:

1. there exists a subcollection of σ that consists of k blocks $\{C_{j_1}, \ldots, C_{j_k}\}$ such that $\bigcup\{C_{j_\ell} \mid 1 \leq \ell \leq k\}$ is a block B_h of π;
2. for $1 \leq i \leq n$ and $i \neq h$, B_i is a block of σ.

For $k = 2$ the relation $\geq_2$ is the direct coverage relation, where the larger partition π is obtained by fusing two blocks of σ.

If $\pi \in \mathsf{Part}(S)$ and $\pi = \{B_1, \ldots, B_n\}$, its entropy is the number

$$\mathcal{H}(\pi) = -\sum_{i=1}^{n} \frac{|B_i|}{|S|} \log_2 \frac{|B_i|}{|S|},$$

which is actually the entropy of the discrete probability distribution

$$\mathbf{p} = \left(\frac{|B_1|}{|S|}, \ldots, \frac{|B_n|}{|S|}\right).$$

Defining the entropy for partitions rather than for probability distributions has the advantage of linking the entropic properties to the partially ordered set of partitions. An important fact is that the entropy is anti-monotonic relative to the partial order defined on partitions. In other words, for $\pi, \sigma \in \mathsf{Part}(S)$, $\pi \leq \sigma$ implies $\mathcal{H}(\pi) \geq \mathcal{H}(\sigma)$. It is easy to verify that $\mathcal{H}(\omega_S) = 0$ and that $\mathcal{H}(\iota_S) = \log_2 |S|$. This shows that the entropy can be used to evaluate the uniformity of the elements of S in the blocks of π since the entropy value increases with the uniformity of the distribution of the elements of S. Note that as the uniformity increases, so does the associated uncertainty.

If C is a non-empty subset of S, and $\pi \in \mathsf{Part}(S)$, the *trace* of π on C is the partition

$$\pi_C = \{B \cap C \mid B \in \pi \text{ and } B \cap C \neq \emptyset\}.$$

The trace of a partition allows us to define the conditional entropy of two partitions. Namely, if $\pi, \sigma \in \mathsf{Part}(S)$ and $\sigma = \{C_1, \ldots, C_m\}$, then the *entropy of* π *conditioned by* σ is the number

$$\mathcal{H}(\pi|\sigma) = \sum_{j=1}^{m} \frac{|C_j|}{|S|} \mathcal{H}(\pi_{C_j}).$$

It can be shown [4,5] that the conditional entropy is an anti-monotonic function of the first argument and a monotonic function of the second. In other words, $\pi_1 \leq \pi_2$ implies $\mathcal{H}(\pi_1|\sigma) \geq \mathcal{H}(\pi_2|\sigma)$ and $\sigma_1 \leq \sigma_2$ implies $\mathcal{H}(\pi|\sigma_1) \leq \mathcal{H}(\pi|\sigma_2)$.

A *measure* on S is a function $m : \mathcal{P}(S) \longrightarrow \mathbb{R}_{\geq 0}$ such that $m(U \cup V) = m(U) + m(V)$ for every disjoint subsets U and V of S. For example, if S is the set of pixels of a gray image S, $m(U)$ can be defined as the number of pixels having a certain degree of grayness contained by the subset U.

Let D be a finite set. A D-feature function on S is a function $f : S \longrightarrow D$. Each feature function $f : S \longrightarrow D$ defines a partition $\ker f$ on S defined by

$$\ker f = \{f^{-1}(d) \mid d \in D, f^{-1}(d) \neq \emptyset\}.$$

We refer to $\ker f$ as the *kernel partition* of f.

For example, if S is the set of pixels of an image, we could define $f(p)$ as the degree of grayness of the pixel $p \in S$. Another example that is relevant in the study of biodiversity is to consider a set S of observation points in a territory, and define $f(p)$ as the number of species of birds sighted in a certain day in p.

If $C \subseteq S$, then the characteristics of the trace partition $(\ker f)_C$ define the concentration of the values that f takes on the set C. If $D = \{d_1, \ldots, d_k\}$, the blocks of the partition $(\ker f)_C$ have the relative sizes

$$\frac{|f^{-1}(d_1) \cap C|}{|C|}, \ldots, \frac{|f^{-1}(d_k) \cap C|}{|C|}$$

and the distribution of these sizes can be conveniently represented using a histogram.

Definition 1. *Let $\varPi = (\pi_1, \pi_2, \ldots, \pi_n)$ be a descending chain of partitions on S such that $\pi_1 = \omega_S$, $f : S \longrightarrow D$ be a feature function, $m : \mathcal{P}(S) \longrightarrow \mathbb{R}_{\geq 0}$ be a measure defined on S and let $\theta, \mu > 0$ be two positive number referred to as the* entropic threshold *and the* measure threshold*, respectively.*

The entropic tree *defined by $\varPi, f, m, \theta$ and μ is a tree $\mathcal{T}(\varPi, f, m, \theta, \mu)$ whose set of nodes consists of blocks of the partitions π_i such that the following conditions are satisfied:*

(i) *the root of the tree is the set S, the unique block of ω_S;*
(ii) *an edge (B, C) exists in the tree only if $B \in \pi_i$, $C \in \pi_{i+1}$, and $C \subseteq B$;*
(iii) *if B is a block of the partition π_i, then $\mathcal{T}(\varPi, f, m, \theta, \mu)$ contains the set of edges $\{(B, C) \mid B \in \pi_i \text{ and } C \in \pi_{i+1}, C \subseteq B\}$ if and only if $\mathcal{H}((\ker f)_B) \geq \theta$ and $m(B) \geq \mu$.*

If $\mathcal{T}(\varPi, f, m, \theta, \mu)$ contains the set of edges $\{(B, C) \mid B \in \pi_i \text{ and } C \in \pi_{i+1}\}$ we say that the node B is *split* in the tree $\mathcal{T}(\varPi, f, m, \theta, \mu)$. Since splitting involves a sufficiently large value of the entropy and a node of sufficiently large measure, longer paths in the tree point towards subsets of S that contain a large diversity of values of the feature function f.

An entropic quadtree is an entropic tree $\mathcal{T}(\Pi, f, m, \theta, \mu)$ such that $\Pi = (\pi_1, \ldots, \pi_n)$ is a descending chain of partitions on S, $\pi_1 \geq_4 \pi_2 \geq_4 \cdots \geq_4 \pi_n$. The entire image area S corresponds to the root of the quadtree.

The expansion of a node B is based on its entropy value and the predetermined threshold used for the splitting condition, as well as the size of the corresponding sub-area. Only nodes with area greater or equal to the defined minimum window size are expanded. The complex areas corresponding to leaves at the highest level on the quadtree are classified according to the possibility of presence of an edge.

3 Algorithm Description

The algorithm proposed constructs a full entropic quadtree related to the image entropy concentration to find high complexity areas that can also contain edges. Later, a slightly modified CHT is used to detect the presence of circles in the complex areas found during the entropy analysis. The algorithm receives as input the 8 bits gray scale version of an image, a minimum window size for analysis, a threshold relevant to the node splitting condition, the minimum and maximum radius values for the searched craters and a threshold for the CHT. Its output lists the detected craters as well as their estimated center points highlighted and superimposed over the original image. A text file with data indicating the center points, radius and Hough Space bin points of each detected crater is also generated.

The construction of the entropic quadtree is based on the measurements of the entropy in image sub-areas, which can also be regarded as tree nodes.

The entire image area corresponds to the root of the quadtree. The expansion of each node is based on its entropy value and the predetermined threshold used for the splitting condition, as well as the size of the corresponding sub-area. Only nodes with area greater or equal to the defined minimum window size are expanded. The complex areas corresponding to leaves at the highest level on the quadtree are classified according to the possibility of presence of an edge.

First, the algorithm determines the average gray intensity of the original image, as well as the low intensity average(average of gray shades below average intensity) and high intensity average(average of gray shades above average intensity). Then, the pixels in each area with minimum size for analysis are mapped to two different sets according to the thresholds corresponding to the average of low intensity shades or the average of high intensity shades of the original image. Crater edges can be found in areas that contain only dark shades of gray or areas containing light shades of gray.

The classification considers the number of pixels in a minimum size window that are above the high intensity average threshold if at least one pixel in the area has gray shade above the average intensity. Otherwise, if all the pixels in the area have low intensity, the classification considers the number of pixels in the minimum size window that have shade below the low intensity average threshold.

Let n be the number of pixels satisfying one of those conditions and h the height of our minimum window. Also, suppose we have a square window. When $h - 2 < n$

$< h^2 - 1$, the entropy value remains considerably high and the area is classified as a leaf that possibly contains an edge. Only those leaves are relevant to our algorithm.

After the high complexity regions that may contain edges are found, the algorithm determines another threshold corresponding to a high intensity shade which is higher than the high intensity average shade, lower than the maximum intensity found in the image and has the highest histogram value among the shades satisfying the couple previous conditions. This last threshold which we will call ”near maximum intensity” threshold is used to highlight high intensity pixels corresponding to edges.

Pixels with light shades of gray(higher than average intensity) that form edges usually have intensity greater than the ”near maximum intensity” threshold. Finally, the entropy analysis generates a binary image where pixels with shades of gray below the low intensity threshold and pixels with shades of gray above the near maximum intensity threshold are mapped to white. All the other pixels are mapped to black. The resultant binary image corresponds to the output of the entropy analysis and input of the Circle Hough Transform method. As previously mentioned, the original image does not need any pre-processing. The entropy analysis works as an information theoretic edge filter that generates a binary image from complex areas which may contain edges.

Our next step is to apply the CHT to detect circles in the binary image. As described in subsection 3.2, the CHT method maintains an accumulator array to find triplets (a, b, r) that describe circles where (a, b) is the center of a circle with radius r. Each point (a, b) in the image receives a score value referred to as the *number of votes* equal to the number of points (x, y) fall on the perimeter of the circle (a, b, r). This score is stored in a accumulator array. The detected center points have de highest numbers of votes.

Two stopping conditions are commonly used by the CHT algorithm: the maximum number of circles to be found and a threshold for the minimum number of votes related to a point in the parameter space.

In our application, there is no systematic way to reasonably predict both values. Furthermore, it was observed that for any set of radii where the difference between the minimum and maximum radius is relatively small, the chances of a point to represent a real circle center decreases as the number of votes related to the point gets further from the peak value found in the accumulator array. Points with a number of votes relatively far from the peak value usually correspond to near true center points, near center points of poorly delimited circles or points that received votes in the parameter space simply due to noisy pixels that are not part of any circle edge. To alleviate this problem, we created a new threshold for the number of votes corresponding to the maximum distance from the peak value in the accumulator array as our stopping condition for the CHT method. We also restricted each search to small sets of contiguous radii. Details are described in subsection 3.2.

The algorithm is presented in Fig. 1. The function COMPUTE_ENTROPY evaluates the entropy associated with the histogram of the pixels in the node's area.

The recursive method SPLIT introduced in Fig. 2 expands a node if its feature satisfies the splitting condition and if its area is greater or equal to the predefined minimum area size. Thus, each leaf on the quadtree is classified according to the possibility of

```
Input: 8 bits gray scale version of an image, a minimum area size, entropy threshold, the
       minimum and maximum radius, CHT threshold
Result: Detected craters as well as their estimated center points highlighted and
        superimposed over the original image; a text file with data indicating the center
        points, radius and Hough Space bin points of each detected crater
begin
    $nId \longleftarrow ROOT$;
    $nLevel \longleftarrow 0$;
    $root \longleftarrow newNode(nId, nLevel, image.width, image.height)$;
    $COMPUTE\_ENTROPY(root)$;
    $SPLIT(root)$;
    $entropyImg \longleftarrow PROCESS\_ENTROPY\_IMAGE()$;
    $COMPUTE\_CHT(entropyImg, minRadius, maxRadius, thrCHT)$;
end
```

Fig. 1. FIND_CRATERS(*image,minArea,thrEntropy,minRadius, maxRadius,thrCHT*)

presence of an edge and only those which may contain an edge are considered at the next method PROCESS_ENTROPY_IMAGE.

This method generates a binary image representing the entropy analysis to find complex areas that may contain edges. Pixels with shades of gray below the low intensity threshold and pixels with shades of gray above the near maximum intensity threshold are highlighting in white. All remaining pixels are mapped to black.

COMPUTE_CHT detects circles in the binary image with radii between the minimum and maximum values given as arguments. It also highlights the detected craters as well as their estimated center points over the original image and generates a text file with data related to the craters found such as radius, center points and number of points in the bins associated with each center point.

3.1 Information-Theoretical Method

Our method evaluates the entropy of the local histograms of image sub-areas to find high complexity regions. The partition blocks of a node, used for the entropy analysis, consist of pixels with the same shade of gray.

Fig. 3 presents the information-theoretic method proposed. It computes the entropy associated with the histogram of the pixels in a node's area. This histogram is created by the method INSERT_GRAYSHADE. The result generated by COMPUTE_ENTROPY is successively used by the recursive method SPLIT shown in Fig. 2. Only the nodes corresponding to sub-areas of the image where the entropy is above the predefined entropy threshold and have area greater or equal to the pre-defined minimum area size are expanded. We observed that leaves at the highest level in the resultant quadtree may naturally have different associated entropy values. CLASSIFY_LEAF classifies a minimum area node as containing or not an edge. As previously mentioned, only leaves that may contain edges are relevant for our algorithm.

```
Input: A node n from a quadtree
Result: Expands the node creating four children, if node satisfies the necessary
        requirements
begin
    if $(n.feature > method\_lower\_bound)$ and $(n.area > minArea)$ then
        $nLevel \longleftarrow n.level + 1$;
        $nId \longleftarrow n.id + A$;
        $topLeft \longleftarrow newNode(nId, nLevel, n.rect.x, n.rect.y, ;$
        $n.rect.width/2, n.rect.height/2)$;
        $COMPUTE\_ENTROPY(topLeft)$;
        $nId \longleftarrow n.id + B$;
        $topRight \longleftarrow newNode(nId, nLevel, n.rect.x +$
        $n.rect.width/2,\ n.rect.y, n.rect.width/2, n.rect.height/2)$;
        $COMPUTE\_ENTROPY(topRight)$;
        $nId \longleftarrow n.id + C$;
        $bottonLeft \longleftarrow newNode(nId, nLevel, n.rect.x, n.rect.y+ ;$
        $n.rect.height/2, n.rect.width/2, n.rect.height/2)$;
        $COMPUTE\_ENTROPY(bottonLeft)$;
        $nId \longleftarrow n.id + D$;
        $bottonRight \longleftarrow newNode(nId, nLevel, n.rect.x + n.rect.width/2, ;$
        $n.rect.y + n.rect.height/2, n.rect.width/2, n.rect.height/2)$;
        $COMPUTE\_ENTROPY(bottonRight)$;
        $RELEASE(n)$;
        $SPLIT(topLeft)$;
        $SPLIT(topRight)$;
        $SPLIT(bottonLeft)$;
        $SPLIT(bottonRight)$;
    else
        $SAVE\_INFO\_NODE(n)$;
        $DRAW(n)$;
        $RELEASE(n)$;
end
```

Fig. 2. SPLIT(n)

3.2 Circular Hough Transform Method

The Hough Transform is a standard method for shape recognition in digital images. It was first applied to the recognition of straight lines [6,7] and later extended to circles [8,9], ellipses [10], and arbitrary shaped objects [11]. The Circular Hough Transform (CHT) can be used to determine the parameters of a circle when a number of points that fall on the perimeter are known. A circle with radius r and center (a, b) can be described with the parametric equations:

$$x = a + r \cos \varphi \text{ and } y = b + r \sin \varphi.$$

The locus of (x, y) points in the Hough or parameter space falls on a circle of radius r centered at (a, b). The true center point will be common to all parameter circles, and

```
Input: A node $n$ from a quadtree
Result: The node entropy related to the histogram of the pixels in the area.
begin
    $entropy \longleftarrow 0$;
    foreach $pixel \in n.area$ do
        INSERT_GRAYSHADE($HISTOGRAM$,$pixel.shade$);
    foreach $shade \in HISTOGRAM$ do
        $p \longleftarrow number\_of\_pixels\_with\_shade$;
        $s \longleftarrow total\_number\_of\_pixels\_in\_the\_node$;
        $g \longleftarrow (p \div s)$;
        $entropy- = (g) \times (\lg_2(g))$;
    if $(n.area == minArea)$ then
        $relevantLeaf \longleftarrow CLASSIFY\_LEAF(HISTOGRAM)$;
        if $(!relevantLeaf)$ then
            return 0;
    return $entropy$;
end
```

Fig. 3. COMPUTE_ENTROPY(n)

can be found with an accumulator array that stores the number of votes for each point in the parameter space. Multiple circles with the same radius can be found with the same technique.

The main disadvantage of the transform is the fact that the parameter space corresponds to a 3-dimensional space, which makes the computational complexity and storage requirements $O(n^3)$. If the circles in an image are of known radius r, the search can be reduced to a 2-dimensional space.

The method used in our algorithm searches for all circles with radius between two values given as arguments. It differs from other version of CHT methods because of its stopping condition. For reasons previously mentioned, our method does not use the maximum number of circles or the minimum threshold for the number of votes in order to end the search. Instead, it uses a threshold corresponding to the maximum allowed difference between the peak value in the accumulator array of votes and any other number of votes related to a point in the parameter space.

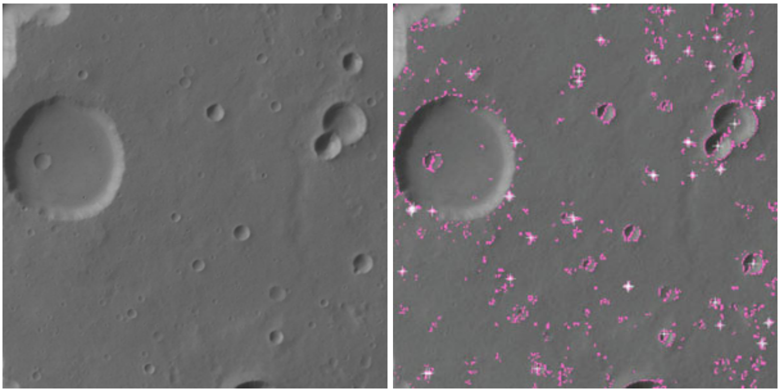
Let $A_{[W][H][R]}$ denote the accumulator array of votes where W is the image width, H is the image height and R depends on the size of the radius set with minimum element $rmin$ and maximum element $rmax$, and on the value for the chosen radius increment i. Let t denote the introduced threshold and v be the greatest value stored in the accumulator array A corresponding to a point (w, h, r) where $0 \leq w \leq W - 1$, $0 \leq h \leq H - 1$ and $0 \leq r <= (rmax - rmin) \div i$. Then an arbitrary point (w', h', r') where $0 \leq w' \leq W - 1$, $0 \leq h' \leq H - 1$ and $0 \leq r' <= (rmax - rmin) \div i$ having v' votes in A is detected as a circle center iff $v - v' \leq t$. We observed that our threshold works well for small groups of contiguous radii. Since the size of the group is small, all the radii are close in value and points corresponding to the center of a circle with one of those radii also have a relatively close number of votes in the parameter space.

```
Input: image generated by the entropy analysis,minimum and maximum radius, new
       threshold for stop condition
Result: Detected circles, their estimated center points and radius
begin
    HOUGH_TRANSFORM(entropyImg,minRadius,maxRadius);
    PROCESS_HS();
    GET_CENTER_POINTS(thrCHT);
    DRAW_CIRCLES();
    PRINT_CIRCLES_DATA();
end
```

Fig. 4. COMPUTE_CHT$(entropyImg, minRadius, maxRadius, thrCHT)$

Therefore, the difference between the number of votes corresponding to centers of true circles that are reasonably well delimited cannot be large when the search is performed for a small group of contiguous radii.

Fig. 4 presents the CHT proposed. HOUGH_TRANSFORM computes the Hough Transform of the binary image generated during the entropy analysis and PROCESS_HS generates an image corresponding to the Hough Space. GET_CENTER_POINTS finds circles and their center points by checking the accumulator array containing votes for each pixel in the image. DRAW_CIRCLES() highlights the detected craters as well as their estimated center points over the original image. PRINT_CIRCLES_DATA generates a text file with data related to the craters found such as radius, center points and number of points in the bins associated with each center point.



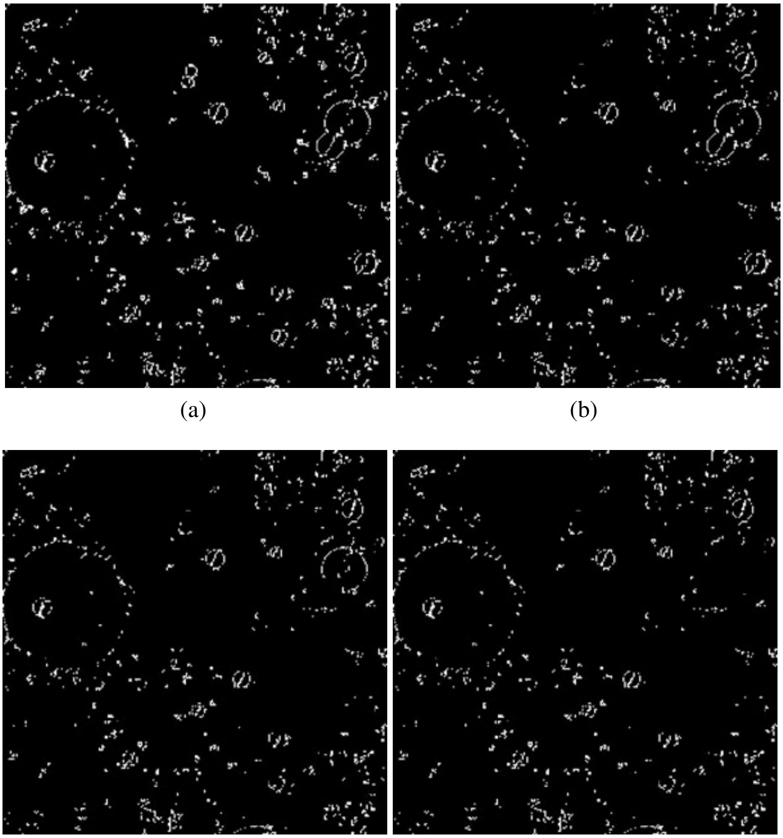
(a) (b)

Fig. 5. Sample image from Mars surface (768x768, 24 bits per pixel) (a) Original. (b) Final image generated by the algorithm. Detected craters and their estimated centers are highlighted in pink and white on the original image.

4 Experimental Results

Experiments were performed over the decompressed 768x768, 8 bits gray scale version of the JPEG digital image corresponding to a picture of Mars surface presented in Fig. 5(a). This image was obtained from the original 24 bits/pixel PGM digital image labeled 3_24 used as training site by the authors of [2]. 3_24 corresponds to one section of a footprint image(h0905_0000) from the High Resolution Stereo Camera (HRSC) instrument of the MarsExpress orbiter. This footprint is about 8248 x 65448 pixels in size and was split into 264(6 x 44) sections of 1700 x 1700 pixels each. Image 3_24 corresponds to one of those sections.

The use of gray scale images allowed the methods to be applied over a reduced color space. We used a 3 × 3 minimum area for the entropy analysis and an high entropy



(a)

(b)

(c)

(d)

Fig. 6. Intermediate results of crater detection. (a) Binary image generated by the entropy analysis. (b) Binary image generated after detection of craters with radius between 5 and 20. (c) Binary image generated after detection of craters with radius between 21 and 36. (d) Binary image generated after detection of craters with radius between 37 and 52. For (b), (c) and (d), the pixels corresponding to the circles detected by the CHT are mapped to black.

threshold equal to 3 due to the heavy presence of texture in the original image. Images corresponding to natural scenes, objects and faces with a textured background or images with a high level of noise contain a large amount of information. It is natural that those images contain more areas with high entropy than images with less textured background. Fig. 6(a) shows the binary image generated during the entropy analysis.

We chose a threshold equal to 30 for the CHT method and divided the search into runs containing 15 contiguous values of the radius. We focused on searching for craters with radii varying from 5 to 52 pixels. It was also observed that the choice regarding the search for only 15 radii at a time, combined with a threshold equals to 30 provided reasonably good results. Since the values of the radius in each run are close, the difference between the number of votes for the points corresponding to centers of well delimited circles is usually not greater than 30. Our algorithm was able to detect 50 craters with radii varying between 5 and 20 pixels, 2 craters with radii varying from 20 to 36 pixels and one crater with radius equal to 45 pixels.

Fig. 5(b)shows the final image generated by the algorithm. The detected craters and estimated center points are highlighted over the original image. Notice that for some craters, the center points are slightly shifted to the left or right of the true center point because the characteristic shadow inside the crater is also detected as an edge by the entropy analysis. As presented in Figs. 6(b), 6(c) and 6(d), the algorithm cleans the areas of the entropy analysis image corresponding to the craters found(by mapping their pixels to black) after each CHT run. This cleaning process helps to decrease the amount of noise and therefore undesirable circles overlapping for subsequent runs.

The heavy presence of texture in the image can highly impact the quality of the intermediate image generated by the entropy analysis, which works also as a edge detector tool based on entropy. As the level of texture or noise increases, so does the entropy of regions in the picture. As consequence, the distinction between hight entropy nodes that may contain edges becomes harder. On the other hand, the quality of the results generated by the CHT method highly depend on the quality of the entropy analysis image taken as input. Specially, as the detected edges get more and more similar to the real crater edges, it becomes easier for the CHT method to accurately recognize those circles corresponding to craters. We noticed that the image generated by the entropy analysis does not show all the possible true edges corresponding to crater borders. In order to avoid the capturing of heavy noise, we use a high entropy threshold. By using such high threshold, the algorithm cannot capture true crater borders in areas where the variance among the pixels is not high. As a consequence, those craters cannot be detected by the CHT method. Therefore, improving the detection of edges for heavily noisy or textured images during the entropy analysis can directly impact the quality of the final results. Results also show that the algorithm may detect a larger number of false positives craters as the radius increases. Remains of smaller circles that were not completely cleaned from the binary image due to the imperfection of circle edges, may contribute for undesirable circle overlapping in the Hough Space.

5 Conclusion

An algorithm to detect circles that can possibly correspond to craters in images was introduced. The algorithm performs an information-theoretic analysis of the histogram of sub regions of the image in order to find complex areas that may contain edges. A modified CHT detects circles in those complex areas and provides information about center points and radius of the circles found. A threshold corresponding to the maximum distance from the peak value in the accumulator array is used as stopping condition for the method.

There is no external pre-processing of the original image 3_24 other than conversion to JPEG format and resizing. No external edge filter is used to process the original image prior to the CHT method.The entropy analysis works as an edge filter that generates the binary image given as input to the CHT method. The heavy presence of noise and texture may compromise the quality of the complex areas found during the entropy analysis and impact the quality of the final results. Therefore, by improving the robustness of the entropy analysis against heavy noise and texture, more craters will accurately be detected.

We intend to extend the application of information-theoretical techniques to other structures associated with spatial data sets such as grid-files, (k, d)-trees, and R-trees. Another area of great potential is the application of entropic quadtrees to the identification of terrain areas that contain a high level of biodiversity.

Acknowledgements

The authors thank Dr. Wei Ding for providing the original image 3_24 used in this work as well as the ground truth data corresponding to the manually marked craters in 3_24.

References

1. Bue, B.D., Stepinski, T.F.: Machine detection of martian impact craters from digital topography data. IEEE Transactions on Geoscience and Remote Sensing 45(1), 265–274 (2007)
2. Urbach, E.R., Stepinski, T.F.: Automatic detection of sub-kilometer craters in high resolution planetary images. Planetary and Space Science 57, 880–887 (2009)
3. Salamuniccar, G., Loncaric, S.: Gt-57633 catalougue of martian impact craters developed for evaluation of crater detection algorithms. Planetary and Space Science 56, 1992–2008 (2008)
4. Simovici, D.A., Jaroszewicz, S.: An axiomatization of partition entropy. IEEE Transactions on Information Theory 48, 2138–2142 (2002)
5. Simovici, D., Jaroszewicz, S.: Generalized conditional entropy and decision trees. In: Extraction et Gestion des connaissances – EGC 2003, Lavoisier, Paris, pp. 363–380 (2003)
6. Leavers, V.F.: Survey: which hough transform? Computer Vision Graphics and Image Processing: Image Understanding 58(2), 250–264 (1993)
7. Illingworth, J., Kittler, J.: Survey: a survey of the hough transform? Computer Vision Graphics and Image Processing 44(1), 87–116 (1988)
8. Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Communications of the Association of Computin Machinery 15(1), 11–15 (1972)
9. Davis, E.R.: A modified hough scheme for general circle location. Pattern Recognition Letters 7, 37–43 (1987)
10. Yip, R.K.K., Tam, P.K.S., Leung, D.N.K.: Modification of hough transform for circles and ellipses detection using a 2-dimensional array. Pattern Recognition 25, 1007–1022 (1992)
11. Pao, D.C.W., Li, H.F., Jayakumar, R.: Shape recognition using the straight line hough transform: theory and generalization. IEEE Transactions on Pattern Analysis and Machine Intelligence 14(11), 1076–1089 (1992)

Hybrid DIAAF/RS: Statistical Textual Feature Selection for Language-Independent Text Classification

Yanbo J. Wang[1], Fan Li[1], Frans Coenen[2], Robert Sanderson[3], and Qin Xin[4]

[1] Information Management Center, China Minsheng Banking Corp., Ltd., Beijing, China
{wangyanbo,lifan}@cmbc.com.cn
[2] Department of Computer Science, University of Liverpool, Liverpool, UK
coenen@liverpool.ac.uk
[3] Los Alamos National Laboratory, Los Alamos, New Mexico, USA
rsanderson@lanl.gov
[4] Simula Research Laboratory, Oslo, Norway
xin@simula.no

Abstract. Textual Feature Selection (TFS) is an important phase in the process of text classification. It aims to identify the most significant textual features (i.e. key words and/or phrases), in a textual dataset, that serve to distinguish between text categories. In TFS, basic techniques can be divided into two groups: linguistic vs. statistical. For the purpose of building a language-independent text classifier, the study reported here is concerned with statistical TFS only. In this paper, we propose a novel statistical TFS approach that hybridizes the ideas of two existing techniques, DIAAF (Darmstadt Indexing Approach Association Factor) and RS (Relevancy Score). With respect to associative (text) classification, the experimental results demonstrate that the proposed approach can produce greater classification accuracy than other alternative approaches.

Keywords: Associative Classification, (Language-independent) Text Classification, Text Mining, Textual Feature Selection.

1 Introduction

1.1 General Background

The increasing number of electronic documents that are available to be explored online has led to *text mining* becoming a promising school of current research in *Knowledge Discovery in Data* (*KDD*), and is attracting increasing attention from a wide range of different groups of people. Text mining aims to extract various models of hidden, interesting, previously unknown and potentially useful knowledge (i.e. rules, patterns, regularities, customs, trends, etc.) from sets of collected textual data (i.e. web news, e-mails, research papers, meeting minutes, etc.), where a collected textual dataset can be sized in *Giga-bytes*. In a natural language context, a given textual dataset is commonly refined to produce a *documentbase* — a set of electronic documents that typically consists of thousands of documents, where each document may contain hundreds of words.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 222–236, 2010.

One major application of text mining is *Text Classification/Categorization* (*TC*) — the automated assignation of "unseen" documents into predefined text groups. TC, as a well established research filed, has been studied for almost half a century; early work on TC can be dated back to the 1960s (see for instance [21]). During the past decade, TC has been extensively investigated at the intersection of research into KDD and machine learning. Machine learning based TC focuses on *directly* assigning "unseen" documents into text categories without being concerned with presenting to end users reasons why and how the classification predictions have been made. KDD based TC typically mines and generates human readable classification rules from textual data that are further used to build a text classifier for assigning "unseen" documents into text classes; such generated textual rules can be presented to the end user. In our study, we concentrate on KDD based TC.

In general, TC can be divided into two groups: (i) *single-label* TC, which assigns each "unseen" document into exactly one (predefined) text class; and (ii) *multi-label* TC, which assigns each "unseen" document into one or more text class. With respect to single-label TC, three different approaches can be identified: (i) *one-class* TC, which learns from positive document samples only, and either assigns an "unseen" document into the predefined (text) class or ignores the assignation of this document; (ii) *two-class* (or *binary*) TC, which learns from both positive and negative document samples, and assigns each "unseen" document into the predefined class or the complement of this class; and (iii) *multi-class* TC, which simultaneously deals with all given classes comprising all document samples, and assigns each "unseen" document into the most appropriate class. This paper is concerned with the single-label multi-class TC study.

Usually text mining requires the given documentbase to be first preprocessed so that it is in an appropriate format. Hence the process of TC, in a general context, can be identified as *documentbase preprocessing* plus *data classification*. The nature of such preprocessing comprises: (i) *documentbase representation*, the process of creating a data model to precisely interpret a given documentbase in an explicit and structured manner; and (ii) *Textual Feature Selection* (*TFS*), the process of extracting the most significant textual information from the given documentbase.

In documentbase representations, the "*bag of* *" or *Vector Space Model* (*VSM*) [25] is considered to be appropriate for many text mining applications. The VSM can be described as follows: given a documentbase $\mathit{Đ}$, each document $D_j \in \mathit{Đ}$ is represented by a single numeric vector, and each vector is a subset of some vocabulary V. The vocabulary V is a representation of the set of textual features (documentbase attributes) that are used to characterize the documents. The VSM is usually presented in a *binary form*, where "*each coordinate of a document vector is zero (when the corresponding attribute is absent) or unity (when the corresponding attribute is present)*" [16]. In TC, there are two major approaches used to define the "bag of *" (vector space) model: the "*bag of words*" and the "*bag of phrases*". The experimental work, in this paper, is designed with respect to both approaches.

Theoretically speaking, the textual features of a document can include every word or phrase that might be expected to occur in a given documentbase. However, this is computationally unrealistic, so it requires some method of preprocessing documents to identify the *key* textual features that will be useful for a particular text mining application, such as TC. TFS aims to select a limited number of textual features from the entire set representing the documentbase. With respect to TFS (sometimes referred to as "*textual feature reduction*"), techniques can be generally divided into two groups: *linguistic* and *statistical.*

Linguistic TFS methods identify significant textual features depending on the rules and/or regularities in semantics, syntax and/or lexicology. Typical methods in this group include: *stop*-word lists, stemming, lemmatization, *part-of-speech* tagging, etc. Such techniques are designed with particular languages and styles of language as the target, and involve deep linguistic analysis. For the purpose of building a *language-independent* text classifier (e.g. [8, 29]) that is generally applicable to *cross-lingual*, *multi-lingual* and/or *unknown-lingual* textual data collections, the statistical approach is most appropriate. This is the focus of this paper. A number of statistical TS mechanisms have been proposed, including: Darmstadt Indexing Approach Association Factor (DIAAF), Relevancy Score (RS), Mutual Information (MI), etc.

Classification (or "*data categorization*") deals with structured data, especially *tabular* data, and aims to assign "unseen" data instances into predefined data groups, based on a classifier constructed from a training set of data instances associating with (predefined) class-labels. Mechanisms on which classification algorithms have been based can be separated into two "families": (i) *classification direct learning*, classification without rule generation; and (ii) *classification rule mining* (e.g. [23]), classification with rule generation (and presentation).

Classification direct learning algorithms focus on directly categorizing "unseen" data records into predefined data groups without concern for presenting, to the end users, why and how the categorization predictions have been made. Typical mechanisms include: naïve Bayes, support vector machine and neural networks. Classification rule mining algorithms mine and generate human readable Classification Rules (CRs), again with the objective of building a classifier to classify "unseen" data instances. Typical approaches include: decision trees (C4.5) [23] and RIPPER [9] (Repeated Incremental Pruning to Produce Error Reduction).

One approach to classification rule mining other than C4.5 and RIPPER is to employ Association Rule Mining (ARM) [1] methods to identify the desired CRs, i.e. *associative classification* [2]. Associative classification mines a set of Classification Association Rules (CARs) from a *class-transactional database*. The authors of [6] and the authors of [28] together suggested that results presented in the studies of [19, 20, 32] show that in many cases associative classification offers greater classification accuracy than other classification rule mining methods, such as C4.5 and RIPPER.

During the past decade, associative classification has been applied to TC (e.g. [3, 8, 29, 33]). Note that the binary format of the VSM representation translates easily into the class-transactional format. The advantages offered by associative classification, with respect to other classification rule mining approaches, can be summarized by quoting Antonie and Zaïane [3]:

- Associative text classifier "*is fast during both training and categorization phases*", especially when handling very large databases [3].

- An associative text classifier "*can be read, understood and modified by humans*".

Given the above advantages offered by associative classification with respect to TC, this approach has been adopted in this paper to support the study of statistical TFS for language-independent TC.

1.2 Contribution

A hybrid statistical TFS approach is proposed, which integrates the ideas of two existing (statistical TFS) techniques: DIAAF (Darmstadt Indexing Approach Association Factor) and RS (Relevancy Score), namely Hybrid DIAAF/RS. The evaluation of Hybrid DIAAF/RS, under both the language-independent "bag of words" and "bag of phrases" documentbase representation settings, was conducted using the TFPC (Total From Partial Classification) associative classifier [5, 6, 7]; although any other associative classification algorithm could equally well have been employed. With respect to associative TC, the experimental results demonstrate that Hybrid DIAAF/RS can produce better classification accuracy than other statistical TFS approaches (e.g. DIAAF, RS, MI), thus improving the performance of language-independent TC.

1.3 Paper Organization

The rest of this paper is organized as follows. Section 2 describes some related work relevant to our study, where both the language-independent "bag of words" and "bag of phrases" approaches are reviewed. The DIAAF and RS as well as MI statistical TFS mechanisms are outlined in section 3. In section 4, we propose the Hybrid DIAAF/RS (statistical TFS) approach. The experimental results are presented in section 5. Finally our conclusions and open issues for further research are given in section 6.

2 Documentbase Representation

2.1 Language-Independent "Bag of Words"

The "bag of words" approach has been used in TC investigation for a long time. In this approach, each document is represented by the set of words that are used in the document. Information on the ordering of words within documents as well as the structure of the documents is lost. The problem with this approach is how to effectively and efficiently select a limited, computationally manageable, subset of words from the entire set represented in the documentbase. Usually the "bag of words" approach first removes all punctuation marks (sometimes, all non-alphabetic characters, i.e. numbers, symbols, etc.) from the original documentbase. Then significant words that contribute to the TC task are selected using TFS.

In [8] the authors introduce a three-phase framework for language-independent "bag of words" construction (as follows):

1. Words are first defined in a documentbase "*as continuous sequences of alphabetic characters delimited by non-alphabetic characters, e.g. punctuation marks, white space and numbers*"; all non-alphabetic characters are then removed from the documentbase.
2. Common and rare words are collectively considered to be the *noise* words in a documentbase. They can be identified by their *support* value, i.e. the percentage of documents in the training dataset in which the word appears. Common words are words with a support value above a user-defined Upper Noise Threshold (UNT), and are referred to as upper noise words. Rare words are those with a support value below a user-defined Lower Noise Threshold (LNT), and are referred to as lower noise words. Both upper and lower noise words are then removed from the documentbase.
3. The desired set of significant words is drawn from an ordered list of potential significant words. A potential significant word also referred to as a key word is a non-noise word whose *contribution* value exceeds some user-specified threshold *G*. The contribution value of a word is a measure of the extent to which the word serves to differentiate between classes and can be calculated in a number of ways. Finally the first *K* words are selected from the ordered list of potential significant words, which are further concerned in the CRM stage of TC.

In the third phase, those words whose contribution value exceeds the threshold *G* are placed into a potential significant word list, in descending ordered according to the contribution value. This list may include words that are significant for more than one class (noted as "*all words*"), or it may be decided to include only those words that are significant with respect to one class only (i.e. "*uniques*"). From the potential significant word list the final list of significant words are chosen. Two strategies can be proposed for achieving this. The first is to simply choose the first *K* words from the ordered list (the "*top K*"). This may, however, result in an unequal/unbalanced distribution of significant words between classes. The second approach chooses the top "*K* / |*C*|" words for each class (referred to as "*dist*"), so as to include an equal number of significant words for each class, where *C* is the set of predefined classes within a documentbase.

2.2 Language-Independent "Bag of Phrases"

Instead of representing a documentbase using words, many TC studies consider the usage of phrases. In the "bag of phrases" approach, each element in a document vector represents a phrase describing an ordered combination of words appearing contiguously in sequence (sometimes with some *maximum word gap*). The motivation for this approach is that phrases carry more contextual and/or syntactic information than single words. For example Scheffer and Wrobel [26] argue that the "bag of words" representation does not distinguish between "*I have no objections, thanks*" and "*No thanks, I have objections*".

One "bag of phrases" approach is to use *n*-grams (see for instance [22]), where each sequence of *n* ordered and adjacent words in a document is identified as a phrase ($n \leq$ the size of the document). However, the main question with respect to *n*-grams is what should the value of *n* be? This remains a current research issue.

In [8] the authors propose a language-independent "bag of phrases" approach based on the language-independent "bag of words" construction (see section 2.1). In section 2.1, three categories of word were defined:

- **Upper Noise Words:** Words whose support is above a user-defined UNT (Upper Noise Threshold);
- **Lower Noise Words:** Words whose support is below a user-defined LNT (Lower Noise Threshold); and
- **Significant Words (G):** Selected key words that are expected to serve to distinguish between classes.

In this section, another two categories of word are further defined (also as introduced in [8]):

- **Ordinary Words (O):** Other non-noise words that have not been selected as significant words; and
- **Stop Marks (S):** The "key" punctuation marks: '*,*' '*.*' '*:*' '*;*' '*!*' and '*?*', referred to as *delimiters*, and used in phrase identification. All other non-alphabetic characters are ignored.

It also identifies (in [8]) two groups of categories of words:

- **Noise Words (N):** The union of upper and lower noise words; and
- **Non-noise Words:** The union of significant and ordinary words.

Significant phrases are defined as sequences of words that include at least one significant word. Four different schemes for determining phrases (and constructing a "bag of phrases") were distinguished in [8], depending on: (i) what are used as *delimiters* and (ii) what the *contents* of the phrase should be made up of:

- **DelSNcontGO:** Phrases are delimited by stop marks (S) and/or noise words (N), and made up of sequences of one or more significant words (G) and ordinary words (O). Sequences of ordinary words delimited by stop marks and/or noise words that do not include at least one significant word are ignored.
- **DelSNcontGW:** As DelSNcontGO but replacing ordinary words in phrases by *wild card* symbols (W) that can be matched to any single word. The idea here is that much more generic phrases are generated.
- **DelSOcontGN:** Phrases are delimited by stop marks (S) and/or ordinary words (O), and made up of sequences of one or more significant words (G) and noise words (N). Sequences of noise words delimited by stop marks and/or ordinary words that do not include at least one significant word are ignored.
- **DelSOcontGW:** As DelSOcontGN but replacing noise words in phrases by *wild card* characters (W). Again the idea of this scheme is to produce generic phrases.

The experimental results presented in [8] show that, with respect to the accuracy of classification, DelSNcontGO outperforms other alternative schemes. In this paper, the DelSNcontGO language-independent "bag of phrases" approach will be returned to in Section 5 (experimental results).

3 Statistical Textual Feature Selection

Statistical TFS mechanisms are desired to automatically calculate a weighting score for each textual feature in a document. A significant textual feature is one whose weighting score exceeds a user-supplied weighting threshold. These techniques do not involve linguistic analysis. With regard to TC, the common intuitions are as follows:

- The more times a textual feature appears across the documentbase in documents of all classes the worse it is at discriminating between the classes.
- The more times a textual feature uniquely appears in a class the more relevant it is to this particular class.

In the past, a number of statistical models have been proposed in statistical TFS; three major ones are introduced as follows: Darmstadt Indexing Approach Association Factor (DIAAF), Relevancy Score (RS), and Mutual Information (MI).

- **DIAAF:** Originally, the Darmstadt Indexing Approach (DIA) [13] was "*developed for automatic indexing with a prescribed indexing vocabulary*" [14]. In machine learning, the author of [27] indicates that DIA "*considers properties (of terms, documents, categories, or pairwise relationships among these) as basic dimensions of the learning space*". Examples of such properties include document length, occurrence frequency between textual features and predefined classes, training data generality of each predefined class, etc. One pair-wise relationship in consideration herein is the term-category relationship, noted as the DIA Association Factor (DIAAF) [27], which can be employed to select significant textual features for TC problems. The computation of DIAAF score, also reported in [12], is achieved by using a probabilistic (**Pr**) form:

  $$diaaf\_score(u_h, C_i) = \mathbf{Pr}(C_i \mid u_h) = count(u_h \in C_i) \mathbin{/} count(u_h \in Đ)\,,$$

  where $Đ$ represents a given documentbase, u_h represents a textual feature in $Đ$, C_i represents a set of documents (in $Đ$) labeling with a particular text class, $count(u_h \in C_i)$ is the number of documents containing u_h in C_i, and $count(u_h \in Đ)$ is the number of documents containing u_h in $Đ$. The DIAAF score expresses the proportion of the feature's occurrence in the given class divided by the feature's documentbase occurrence.
- **RS:** The initial concept of RS was given by Salton and Buckley [24], as relevancy weight. It aims to measure how "unbalanced" a textual feature (term) u_h is across documents in a documentbase $Đ$ with and without a particular text class C_i. They define a term's relevancy weight as: "*the proportion of relevant documents in which a term occurs divided by the proportion of nonrelevant items in which the term occurs*" [24]. In [31] the idea of RS was based on relevancy weight with the objective of selecting significant textual features in $Đ$ for the TC application. A term's relevancy score can be defined (in logarithm) as: the number of relevant (the target text class associated) documents in which a term occurs divided by the number of non-relevant documents in which a term occurs.

Sebastiani [27] and Fragoudis *et al.* [12] calculate the RS score in probabilistic (**Pr**) form using:

$$relevancy\_score(u_h, C_i) = log((\mathbf{Pr}(u_h \mid C_i) + d) / (\mathbf{Pr}(u_h \mid \neg C_i) + d)) \,,$$

where $\neg C_i$ (equals to $Đ - C_i$) represents the set of documents labeling with the complement of the predefined class C_i, and d is a constant damping factor. In [31] the value of d was initialized as 1/6. This formula can also be written in the following form:

$$relevancy\_score(u_h, C_i) = log((count(u_h \in C_i) / |C_i| + d) \\ / (count(u_h \in (Đ - C_i)) / |Đ - C_i| + d)) \,,$$

where $|C_i|$ is the size function of set C_i, $|Đ - C_i|$ is the size function of set $Đ - C_i$, and $count(u_h \in (Đ - C_i))$ is the number of documents containing u_h in $Đ - C_i$.

- **MI:** Another important existing statistical TFS mechanism other than DIAAF and RS is Mutual Information (MI). Early study of MI can be seen in [4] and [11]. This statistical model is applied to determine whether a genuine association exists between two textual features or not. In TC, MI has been broadly utilized in a variety of approaches to select the most significant textual features that serve to classify documents. The computation of the MI score between a textual feature u_h and a predefined text class C_i, also reported in [12], is achieved using:

  $$mi\_score(u_h, C_i) = log(\mathbf{Pr}(u_h \mid C_i) / \mathbf{Pr}(u_h)) \,.$$

  This score expresses the proportion (in a logarithmic term) of the frequency with which the feature occurs in documents of the given class divided by the feature's documentbase frequency.

4 Proposed Textual Feature Selection

With respect to language-independent TC, we propose a novel statistical TFS technique in this section. In the previous section, two statistical TFS mechanisms DIAAF and RS were described. The proposed technique is a variant of the original RS approach that makes use of the DIAAF approach, namely Hybrid DIAAF/RS.

Recall that the formula for calculating the RS score is given by:

$$relevancy\_score(u_h, C_i) = log((\mathbf{Pr}(u_h \mid C_i) + d) / (\mathbf{Pr}(u_h \mid \neg C_i) + d)) \,.$$

The core computations here can be recognized as $\mathbf{Pr}(u_h \mid C_i)$ and $\mathbf{Pr}(u_h \mid \neg C_i)$. The DIAAF score is calculated using:

$$diaaf\_score(u_h, C_i) = \mathbf{Pr}(C_i \mid u_h) \,.$$

Substituting for the core computations into the RS score formula using the DIAAF (related) formula, a new RS fashion formula (Hybrid DIAAF/RS) is defined:

$$diaaf\text{-}relevancy\_score(u_h, C_i) = log((\mathbf{Pr}(C_i \mid u_h) + d) / (\mathbf{Pr}(C_i \mid \neg u_h) + d)) \,,$$

where $\neg u_h$ represents a document that does not involve the feature u_h, and d is a constant damping factor (as mentioned in the original RS). The formula can be further expanded as:

$$diaaf\text{-}relevancy\_score(u_h, C_i) = log((count(u_h \in C_i) / count(u_h \in \text{Đ}) + d) / ((count(\neg u_h \in C_i) / count(\neg u_h \in \text{Đ}) + d)) ,$$

where $count(\neg u_h \in C_i)$ is the number of documents containing no u_h in C_i, and $count(\neg u_h \in \text{Đ})$ is the number of documents containing no u_h in Đ.

The algorithm for identifying significant textual features (i.e. key words in our situation, with regard to sections 2.1 and 2.2) in Đ, based on Hybrid DIAAF/RS, is given in Algorithm 1 (as follows):

```
**Algorithm 1: Key Word Identification - Hybrid DIAAF/RS**
**Input:** (a) A documentbase Đ (the training part, where the
               noise words have been removed);
           (b) A user-defined significance threshold $G$;
           (c) A constant damping factor $d$;
**Output:** A set of identified key words $S_{KW}$;
**Begin Algorithm:**
    $S_{KW}$ ← an empty set for holding the identified key
               words in Đ;
    $C$ ← **catch** the set of predefined text classes
           within Đ;
    $W_{GLO}$ ← **read** Đ to create a global word set, where the
               word documentbase support $supp_{GLO}$ is
               associated with each word $u_h$ in $W_{GLO}$;
    **for each** $C_i \in C$ **do**
        $W_{LOC}$ ← **read** documents that reference $C_i$ to
                   create a local word set, where the local
                   support $supp_{LOC}$ is associated with each
                   word $u_h$ in $W_{LOC}$;
        **for each** word $u_h \in W_{LOC}$ **do**
            contribution ← $log(((u_h.supp_{LOC} / u_h.supp_{GLO})$
                               $+ d) / ((|C_i| - u_h.supp_{LOC})$
                               $/ (|Đ| - u_h.supp_{GLO}) + d))$;
            **if** (contribution ≥ $G$) **then**
                **add** $u_h$ into $S_{KW}$;
        **end for**
    **end for**
    **return** ($S_{KW}$);
**End Algorithm**
```

An example of Hybrid DIAAF/RS score calculation is provided in Table 1. Given a documentbase Đ containing 100 documents equally divided into 4 classes (i.e. 25 per class), and assuming that word u_h appears in 30 of the documents and that the value of d (constant damping factor) is 0, then the Hybrid DIAAF/RS score per class can be calculated as shown in the table.

The rationale of this approach is that a significant textual feature (term) with respect to a particular text class should have:

1. A high ratio of the class based term support (document frequency) to the documentbase term support; and/or
2. A low ratio of the class based term support of non-appearance to the documentbase term support of non-appearance.

Table 1. An example of the Hybrid DIAAF/RS score calculation

Class	**# docs per class**	**# docs with u_h per class**	**# docs without u_h per class**	**# docs with u_h in Đ**	**# docs without u_h in Đ**	**Pr(C_i | u_h) + d**	**Pr(C_i | $\neg u_h$) + d**	**Hybrid DIAAF/ RS Score**
1	25	15	10	30	70	0.500	0.143	0.544
2	25	10	15	30	70	0.333	0.214	0.192
3	25	5	20	30	70	0.167	0.286	-0.234
4	25	0	25	30	70	0	0.357	$-\infty$

5 Experimental Results

In this section, we present an evaluation of our proposed statistical TFS approach, using three popular text collections: Usenet Articles, Reuters-21578 and MedLine-OHSUMED. The aim of this evaluation is to assess the approach with respect to the accuracy of classification in both language-independent "bag of words" (section 2.1) and "bag of phrases" (section 2.2) settings. All evaluations given in this section were conducted using the TFPC[1] associative classification algorithm; although any other associative classifier could equally well have been employed. All algorithms involved in the evaluation were implemented using the standard Java programming language. The experiments were run on a 1.87 GHz Intel(R) Core(TM)2 CPU with 2.00 GB of RAM running under Windows Command Processor.

5.1 Experimental Data Description

For the experiments outlined in the following subsections, five individual documentbases were used. Each was extracted (with regard to the documentbase extraction idea in [30]) from one of the three above mentioned text collections.

The Usenet Articles collection is a popular text collection compiled by Lang [17] from 20 different newsgroups, and is sometimes referred to as the "20 Newsgroups" collection. Each newsgroup represents a predefined class. There are exactly 1,000 documents per class with one exception, the class "soc.religion.christian" that contains 997 documents only. In comparison with other common text collections, the structure of "20 Newsgroups" is relatively "*neat*", every document is labeled with one class only, and almost all documents have a "proper" text-content. In the context of this paper, a proper text-content document is one that contains at least q *recognized* words. The value of q is usually small (q is set to be 20 in our study). Previous TC

[1] TFPC software may be obtained from
http://www.csc.liv.ac.uk/~frans/KDD/Software/Apriori-TFPC/aprioriTFPC.html

studies have used this text collection in various ways. For example, in [10] the entire "20 Newsgroups" was randomly divided into two non-overlapping and (almost) equally sized documentbases covering 10 classes each. In this paper we adopted the approach of [10]. The entire collection was randomly split into two documentbases covering 10 classes each: 20NG.D10000.C10 and 20NG.D9997.C10.

Reuters-21578 is another well known text collection widely applied in text mining. It comprises 21,578 documents collected from the Reuters newswire service with 135 predefined classes. However, many TC studies (see for example [18, 34]) have used only the 10 most populous classes. There are 68 classes that consist of fewer than 10 documents, and many others consist of fewer than 100 documents. The extracted documentbase, suggested in [18] and [34], is referred to as Reuters.D10247.C10 and comprises 10,247 documents with 10 classes. However this documentbase includes multi-labeled documents that are inappropriate for a single-label TC investigation (the approach adopted in our study). In this paper, the processing of the Reuters-21578 based documentbase comprised two stages: (1) identification of the top-10 populous classes, as in [18] and [34]; and (2) removal of multi-labeled and/or non-text documents from each class. As a consequence the class "wheat" had only one "*qualified*" document, and no document was found for class "corn". Hence, the final documentbase, namely Reuters.D6643.C8, omitted the "wheat" and "corn", classes leaving a total of 6,643 documents in 8 classes.

The MedLine-OHSUMED text collection, collected by Hersh *et al.* [15], consists of 348,566 records relating to 14,631 predefined MeSH (Medical Subject Headings) categories. The OHSUMED collection accounts for a subset of the MedLine text collection for 1987 to 1991. The process of extracting a documentbase from MedLine-OHSUMED in our study can be detailed as follows. First, the top-100 most populous classes were identified in the collection. These included many super-and-sub class-relationships. Due to the difficulty of obtaining a precise description of all the possible taxonomy-like class-relationships, we simply selected two sets (groups) of 10 target-classes from these classes by hand, so as to exclude obvious super and sub class-relationships in each group. Documents that are either multi-labeled or without a proper text-content (containing $< q$ recognized words) were then removed from each class. Finally two documentbases, namely OHSUMED.D6855.C10 and OHSUMED.D7427.C10, were created.

5.2 Results Using the "Bag of Words" Representation

This section, reports on a set of experiments to evaluate the proposed Hybrid DIAAF/RS TFS approach, in comparison of alternative mechanisms (i.e. DIAAF, RS, and MI), with respect to the "bag of words" representation. Accuracy figures, describing the proportion of correctly classified "unseen" documents, were obtained using Ten-fold Cross Validation (TCV). A *support* threshold value of 0.1%, a *confidence* threshold value of 35% and a Lower Noise Threshold (LNT) value of 0.2% were used as suggested in [8] and [29]. The Upper Noise Threshold (UNT) value was set to be 20%. Following the main findings of [8] the evaluations were conducted using: (i) the "all words" rather than "uniques" strategy in the construction of a potential significant word list, and (ii) the "dist" rather than "top *K*" strategy for choosing the final significant words. The parameter *K* (maximum number of selected final significant words) was set to 1,000. To ensure that sufficient potential significant words were generated for each category, the *G* parameter was given a zero minimal value so that the parameter could be ignored. In both RS and Hybrid DIAAF/RS, 0 was used as the constant damping factor value.

Table 2. Classification accuracy — comparison of the four statistical TFS approaches in the language-independent "bag of words" setting

	DIAAF	**RS**	**MI**	**DIAAF/RS**
20NG.D10000.C10	76.72	76.72	76.72	**77.01**
20NG.D9997.C10	80.61	80.61	80.61	**80.75**
Reuters.D6643.C8	85.40	86.34	86.56	**86.81**
OHSUMED.D6855.C10	77.54	**79.28**	79.27	79.17
OHSUMED.D7427.C10	**78.97**	77.21	77.45	78.12
Average Accuracy	79.85	80.03	80.12	**80.37**
# of Best Accuracies	1	1	0	**3**

The results presented in Table 2 compare 20 classification accuracy values (using the "bag of words" representation) using the test documentbases. From Table 2 it can be seen that the proposed Hybrid DIAAF/RS technique worked better than the other alternative approaches:

1. The overall average classification accuracy throughout can be ranked in order as: Hybrid DIAAF/RS (80.37%), MI (80.12%), RS (80.03%) and DIAAF (79.85%).
2. The number of cases of best classification accuracies obtained throughout the five documentbases can be ranked in order as: Hybrid DIAAF/RS (3 out of 5 cases), DIAAF (1 case), RS (1 case), and MI (none of any case).

5.3 Results Using the "Bag of Phrases" Representation

In this section, we present the experimental results comparing the proposed Hybrid DIAAF/RS TFS approach with previously developed TFS methods (i.e. DIAAF, RS, and MI) using the language-independent "bag of phrases" representation. According to the results presented in [8], the DelSNcontGO phrase generation scheme outperforms other alternative schemes, thus DelSNcontGO was selected to be used in our experiments. All parameters in this section were kept consistent to the parameter setting described in section 5.2 except that *K* was set to 900 for the OHSUMED documentbases. The reason to decrease the value of *K* was that using $K = 1{,}000$ generated more than 2^{15} while the TFPC associative classifier limited the total number of identified attributes[2] (significant words/phrases) to 2^{15}.

Table 3 gives the 20 classification accuracy values obtained using the given documentbases. From Table 3 it can be seen that the proposed Hybrid DIAAF/RS approach outperforms the other alternative approaches:

[2] The TFPC algorithm stores attributes as a signed short integer.

1. The overall average classification accuracy can be ranked ordered as follows: Hybrid DIAAF/RS (81.01%), DIAAF (80.75%), RS (80.52%) and MI (80.49%).
2. The number of cases of best classification accuracies obtained throughout the five documentbases can be ranked in order as: Hybrid DIAAF/RS (3 out of 5 cases), DIAAF (1 case), RS (1 case), and MI (none of any case).

Table 3. Classification accuracy — comparison of the four statistical TFS approaches in the language-independent "bag of phrases" setting

	DIAAF	RS	MI	DIAAF/RS
20NG.D10000.C10	76.96	76.96	76.96	**77.32**
20NG.D9997.C10	81.72	81.72	81.72	**82.09**
Reuters.D6643.C8	87.63	87.94	87.99	**88.53**
OHSUMED.D6855.C10	79.20	**80.16**	80.04	80.03
OHSUMED.D7427.C10	**78.24**	75.80	75.75	77.07
Average Accuracy	80.75	80.52	80.49	**81.01**
# of Best Accuracies	1	1	0	**3**

6 Conclusions

This paper is concerned with an investigation of the statistical textual feature selection for (single-label multi-class) language-independent text classification. An overview of the language-independent documentbase preprocessing, in terms of the "bag of words" and the "bag of phrases" documentbase representations, was provided in section 2. Both the DIAAF and RS statistical TFS techniques were reviewed in section 3. A Hybrid DIAAF/RS (statistical) TFS approach was consequently introduced in section 4, which integrates the ideas of DIAAF and RS. From the experimental results, it can be seen that the proposed Hybrid DIAAF/RS approach outperforms other alternative (statistical TFS) mechanisms in both the language-independent "bag of words" and "bag of phrases" settings regarding the approach of associative classification, Hybrid DIAAF/RS produced the greatest average classification accuracy and the highest number of cases of best classification accuracies throughout the five chosen textual datasets (documentbases). This in turn improves the performance of language-independent text classification.

The results presented in this paper corroborate that the traditional text classification problem can be solved, with good classification accuracy, in a language-independent manner. Further research is suggested to identify the improved statistical textual feature selection mechanism and further improve the performance of language-independent text classification.

Acknowledgments. The authors would like to thank Dr. Jiongyu Li from the China Minsheng Banking Corp., Ltd., Professor Paul Leng and Chuntao Jiang from the University of Liverpool, Songpo Wang from IBM China, Fan Wu from the Beijing eMay Softcom Technology Ltd., and Zhijie Jia from the Beijing Friendship Hotel for their support with respect to the work described here.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Database. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, May 1993, pp. 207–216. ACM Press, New York (1993)
2. Ali, K., Manganaris, S., Srikant, R.: Partial Classification using Association Rules. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Newport Beach, CA, USA, August 1997, pp. 115–118. AAAI Press, Menlo Park (1997)
3. Antonie, M.-L., Zaïane, O.R.: Text Document Categorization by Term Association. In: Proceedings of the 2002 IEEE International Conference on Data Mining, Maebashi City, Japan, December 2002, pp. 19–26. IEEE Computer Society, Los Alamitos (2002)
4. Church, K.W., Hanks, P.: Word Association Norms, Mutual Information, and Lexicography. In: Proceedings of the 27th Annual Meeting on Association for Computational Linguistics, Vancouver, BC, Canada, pp. 76–83. Association for Computational Linguistics (1989)
5. Coenen, F., Leng, P.: An Evaluation of Approaches to Classification Rule Selection. In: Proceedings of the 4th IEEE International Conference on Data Mining, Brighton, UK, November 2004, pp. 359–362. IEEE Computer Society, Los Alamitos (2004)
6. Coenen, F., Leng, P., Zhang, L.: Threshold Tuning for Improved Classification Association Rule Mining. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 216–225. Springer, Heidelberg (2005)
7. Coenen, F., Leng, P.: The Effect of Threshold Values on Association Rule based Classification Accuracy. Journal of Data and Knowledge Engineering 60(2), 345–360 (2007)
8. Coenen, F., Leng, P., Sanderson, R., Wang, Y.J.: Statistical Identification of Key Phrases for Text Classification. In: Proceedings of the 5th International Conference on Machine Learning and Data Mining, Leipzig, Germany, July 2007, pp. 838–853. Springer, Heidelberg (2007)
9. Cohen, W.W.: Fast Effective Rule Induction. In: Proceedings of the 12th International Conference on Machine Learning, Tahoe City, CA, USA, July 1995, pp. 115–123. Morgan Kaufmann Publishers, San Francisco (1995)
10. Deng, Z.-H., Tang, S.-W., Yang, D.-Q., Zhang, M., Wu, X.-B., Yang, M.: Two odds-radio-based Text Classification Algorithms. In: Proceedings of the Third International Conference on Web Information Systems Engineering workshop, Singapore, December 2002, pp. 223–231. IEEE Computer Society, Los Alamitos (2002)
11. Fano, R.M.: Transmission of Information (A Statistical Theory of Communication. The MIT Press, Cambridge (1961)
12. Fragoudis, D., Meretaskis, D., Likothanassis, S.: Best Terms: An Efficient Feature-selection Algorithm for Text Categorization. Knowledge and Information Systems 8(1), 16–33 (2005)
13. Fuhr, N.: Models for Retrieval with Probabilistic Indexing. Information Processing and Management 25(1), 55–72 (1989)
14. Fuhr, N., Buckley, C.: A Probabilistic Learning Approach for Document Indexing. ACM Transactions on Information System 9(3), 223–248 (1991)
15. Hersh, W.R., Buckley, C., Leone, T.J., Hickman, D.H.: OHSUMED: An Interactive Retrieval Evaluation and New Large Test Collection for Research. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, July 1994, pp. 192–201. ACM/Springer (1994)
16. Kobayashi, M., Aono, M.: Vector Space Models for Search and Cluster Mining. In: Berry, M.W. (ed.) Survey of Text Mining – Clustering, Classification, and Retrieval, pp. 103–122. Springer, New York (2004)
17. Lang, K.: News Weeder: Learning to Filter Netnews. In: Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, CA, USA, July 1995, pp. 331–339. Morgan Kaufmann Publishers, San Francisco (1995)

18. Li, X., Liu, B.: Learning to Classify Texts using Positive and Unlabeled Data. In: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, Acapulco, Mexico, August 2003, pp. 587–594. Morgan Kaufmann Publishers, San Francisco (2003)
19. Li, W., Han, J., Pei, J.: CMAR: Accurate and Efficient Classification based on Multiple Class-association Rules. In: Proceedings of the 2001 IEEE International Conference on Data Mining, San Jose, CA, USA, November-December 2001, pp. 369–376. IEEE Computer Society, Los Alamitos (2001)
20. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, August 1998, pp. 80–86. AAAI Press, Menlo Park (1998)
21. Maron, M.E.: Automatic Indexing: An Experimental Inquiry. Journal of the ACM 8(3), 404–417 (1961)
22. Moschitti, A., Basili, R.: Complex Linguistic Features for Text Classification: A Comprehensive Study. In: McDonald, S., Tait, J.I. (eds.) ECIR 2004. LNCS, vol. 2997, pp. 181–196. Springer, Heidelberg (2004)
23. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Francisco (1993)
24. Salton, G., Buckley, C.: Term-weighting Approaches in Automatic Text Retrieval. Information Processing & Management 24(5), 513–523 (1988)
25. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Information Retrieval and Language Processing 18(11), 613–620 (1975)
26. Scheffer, T., Wrobel, S.: Text Classification Beyond the Bag-of-words Representation. In: Proceedings of the Workshop on Text Learning, held at the Nineteenth International Conference on Machine Learning, Sydney, Australia (2002)
27. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)
28. Shidara, Y., Nakamura, A., Kudo, M.: CCIC: Consistent Common Itemsets Classifier. In: Proceedings of the 5th International Conference on Machine Learning and Data Mining, Leipzig, Germany, July 2007, pp. 490–498. Springer, Heidelberg (2007)
29. Wang, Y.J., Coenen, F., Leng, P., Sanderson, R.: Text Classification using Language-independent Pre-processing. In: Proceedings of the Twenty-sixth SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Peterhouse College, Cambridge, UK, December 2006, pp. 413–417. Springer, Heidelberg (2006)
30. Wang, Y.J., Sanderson, R., Coenen, F., Leng, P.: Document-base Extraction for Single-label Text Classification. In: Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery, Turin, Italy, September 2008, pp. 357–367. Springer, Heidelberg (2008)
31. Wiener, E., Pedersen, J.O., Weigend, A.S.: A Neural Network Approach to Topic Spotting. In: Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, NV, USA, April 1995, pp. 317–332 (1995)
32. Yin, X., Han, J.: CPAR: Classification based on Predictive Association Rules. In: Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, May 2003, pp. 331–335. SIAM, Philadelphia (2003)
33. Yoon, Y., Lee, G.G.: Practical Application of Associative Classifier for Document Classification. In: Proceedings of the Second Asia Information Retrieval Symposium, Jeju Island, Korea, October 2005, pp. 467–478. Springer, Heidelberg (2005)
34. Zaïane, O.R., Antonie, M.-L.: Classifying Text Documents by Associating Terms with Text Categories. In: Proceedings of the 13th Australasian Database Conference, Melbourne, Victoria, Australia, January-February 2002, pp. 215–222. CRPIT 5 Australian Computer Society (2002)

Multimedia Summarization in Law Courts: A Clustering-Based Environment for Browsing and Consulting Judicial Folders

E. Fersini[1], E. Messina[1], and F. Archetti[1,2]

[1] DISCo, Università degli Studi di Milano-Bicocca,
Viale Sarca, 336 - 20126 Milano, Italy
{fersini,messina,archetti}@disco.unimib.it

[2] Consorzio Milano Ricerche,
Via Cicognara 7 - 20129 Milano, Italy
archetti@milanoricerche.it

Abstract. Digital videos represent a fundamental informative source of those events that occur during a penal proceedings, which thanks to the technologies available nowadays, can be stored, organized and retrieved in short time and with low cost. However, considering the dimension that a video source can assume during a trial recording, several requirements have been pointed out by judicial actors: fast navigation of the stream, efficient access to data inside and effective representation of relevant contents. One of the possible solutions to these requirements is represented by multimedia summarization aimed at deriving a synthetic representation of audio/video contents, characterized by a limited loss of meaningful information. In this paper a multimedia summarization environment is proposed for defining a storyboard for proceedings celebrated into courtrooms.

1 Introduction and Motivation

Multimedia summarization techniques analyze several informative sources comprises into a multimedia document, with the aim of extracting a semantic abstract. Multimedia summarization techniques available in literature can be divided in three main categories: (1) internal techniques, which exploit low level features of audio, video and text; (2) external techniques, which refer to the information typically associated with a viewing activity and interaction with the user; (3) hybrid techniques, which combine internal and external information.

These techniques are focused on different types of features: (a) domain specific, i.e. typical characteristics of a given domain known a priori and (b) non-domain specific, i.e. non-generic features associated with a particular context.

With respect to internal techniques the main goal is to analyze low-level features derived from text, images and audio contents within a multimedia document. Interesting example can be found in [1], [2] and [3]. In [1] the semantics of objects and events occurring within news video are extracted from subtitles and used to specialize / improve the systems of automatic speech recognition. In

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 237–247, 2010.

[2] the performance related to the identification of special events are increased by combining scene recognition techniques with OCR-based approaches for subtitles recognition in baseball video documents. In [2] the scenes containing text in football videos are recognized using OCR techniques, for then a subsequent identification of key events through audio and video features.

In order to reduce the semantic gap between low level features and semantic concepts, research is moving towards the inclusion of external information that usually comprise knowledge about user-based information and the context in which a multimedia document evolves. The techniques able to generate a video summary on the basis of external information are limited to three case studies [4] [5] [6] focused on using domain specific features. In [4] a summarization technique is proposed in order to gather context information from the acquisition/registration phase, in particular by monitoring the movement of citizens around their houses. Cameras at a specific position and pressure sensors are used to track users. Since users are not required to provide any kind of information, the summary is produced by analyzing data concerning the movement (such as the distance between steps and direction changes). In [5] and [6] semantic annotations, collected during the production phase of the video and described by the standard MPEG-7, are analyzed. In particular in [6] a sequence of audio-video segments is produced on the basis of annotations from video sports (baseball matches), such as players' names or specific events occurring during the match. In [5] a video, characterized by a set of MPEG-7 macro-semantic annotations collected during the acquisition phase, is further annotated by users in order to indicate their level of interest in each video segment. The associations between preferences and the macro-annotations are then modelled by using supervised learning approaches to enable the generation of automatic summary of new multimedia documents.

An attempt that tries to combine the peculiarities of the previous techniques is represented by Hybrid Techniques. Hybrid summarisation techniques combine the advantages provided by internal and external approaches by analyzing a combination of internal and external information. As overviewed for the previous techniques, the hybrid ones can be distinguished in domain specific and non-domain specific. Examples of domain-specific hybrid techniques are related to music videos [7], broadcast news [8] and movies [9]. In non-domain specific approaches we can find two main investigations:

- in [10] the summarization approach could be described by two stages: (1) frames are grouped by a clustering approach, using colour image features; (2) during the editing phase, manual annotations of the representative frame of each cluster, with a subsequent spread to frames of the same cluster, are required. The summaryof is then generated by choosing those representative elements of each cluster matching the user query.
- in [11] an annotation tool is used during the editing phase in order to propagate semantic descriptors to non-labelled contents. During the summary generation phase, the user profile is considered in order to create a customized synthetic representation.

According to the output that a multimedia summarization technique should generate, we can distinguish between static and dynamic summaries. A static summary, also known also as storyboard, can be viewed as a series of key frames or video segments. Approaches for static summaries are focused on identifying relevant contents, do not considering the sequential aspect. They are described by a sub-sampling activity tuned according to the number of desired key-frames. Their use is related to hypermedia documents in order to access the internal parts of a multimedia source. This kind of summary should respect the following requirements: (1) conciseness, they do not to exceed a given limit related to the number of images (key frames), (2) content coverage, they should maximize (minimize) the similarity (dissimilarity) between images for the selection of key frames. A dynamic summary, also known as video skim, consists of a sequence of images associated to their soundtrack. They are generally presented as a video clip or trailer and can be viewed as a preview of the original video, where unimportant shots and scenes are omitted. This kind of summary should respect the following requirements: (1) conciseness, they do not to exceed a given time limit, (2) content coverage, they should maximize the temporal distribution of original video, (3) visual consistency, must minimize the frequency of changes of scene.

There are many differences between static and dynamic summary. The static video summary can be obtained more quickly than the dynamic one because it is focused on the use of only visual features, derived from the images that compose the video itself, without taking into account information from the audio stream. Consequently, once identified the key frames, the creation of the storyboard is a simple activity: audio/video synchronization is not required. A further advantage provided by the static summary is related to the temporal order of the frames: the user is able to quickly understand the contents of a video by looking directly at the sequence of the selected frames. Concerning with dynamic video summary, there are other types of benefits. Dynamic summaries, compared to static ones, use the information coming from the audio stream in a rational way: if on one hand there is a high computational complexity, on the other hand there is a gain in terms of meaning provided by the audio stream.

By analyzing the state of the art related to multimedia summarization techniques, no evidences about summaries over courtroom proceedings are given. The main reasons behind this lack are related to the characteristic of the judicial domain: (1) courtroom recordings are usually charachterized by low quality of audio and video sources; (2) significant events occurring during a debate are not characterized by low level features and therefore we need to understand semantic concepts of interest; (3) a very high level of compression is expected, implying a summary with 2-5 keyframes (which is difficult to derive only from images). For this reasons a comprehensive approach for tackling the current constraints need to be defined. In this paper we are mainly addressing the problem of deriving a storyboard of a multimedia document coming from penal proceedings recordings, by proposing an external summarization technique based on the unsupervised clustering algorithm named Induced Bisecting K-Means. The main outline of this paper is the following. In section 2 the proposed multimedia summarization environment is presented. In section 3 the workflow for deriving a storyboard for the judicial actors is described. In section 4 details about the exploited clustering algorithm are given. Finally, in section 5 conclusions are derived.

2 Multimedia Summarization Environment

In order to address the problem of defining a short and meaningful representation of a debate that is celebrated within a law courtroom, we propose a multimedia summarization environment based on unsupervised learning. The main goal is to create a storyboard of either a hearing or an entire proceedings, by taking into account the semantic information embedded into a courtroom recording.
In particular, the main information sources exploited for producing a multimedia summary are represented by:

- automatic speech transcriptions that correspond to what is uttered by the actors involved into hearings/proceedings. The automatic transcription are provided by Automatic Speech Recognition (ASR) systems, investigated in [14] [15], trained on real judicial data coming from courtrooms. Since it is impossible to derive a deterministic formula able to create a link between the acoustic signal of an utterance and the related sequence of associated words, the ASR system exploits a statistical-probabilistic formulations based on Hidden Markov Models [17]. In particular, a combination of two probabilistic models is used: an acoustic model able to represent phonetics, pronounce variability, time dynamics (co-utterance), and a language model able to represent the knowledge about word sequences.
- automatic audio annotations coming from emotional states recognition (for example fear, neutral, anger). The emotional state annotations are derived through a framework based on a Multi-layer Support Vector Machine approach [18]. Given a set of sentences uttered by different speakers, a features extraction step is firstly performed in order to map the vocal signals into descriptive attributes (prosodic features, formant frequencies, energy, Mel Frequency Cepstral Coefficients, etc...). These features are then used to create a classification model able to infer emotional states of unlabelled speakers.
- automatic video annotations that correspond to what happen during a debate (for instance change of witness posture, new witness, behavior of a given actor). The motion analysis of judicial videos is based on a combinations of video processing algorithms, in order to achieve reliable localization and tracking of significant features. In order to analyze the motions taking place in a video, and to track gestures or head movements of given subjects (typically the witnesses), the optical flow is extracted as the moving points. Then active pixels are separated from the static ones using a kurtosis-based method and finally through a wavelet based approach extracting relevant

features. At this stage the link between low level features and a given set of relevant actions is performed through the induction of Bayesian learner.

The Multimedia Summarization Environment includes two different modules: the acquisition module and the summarization module.

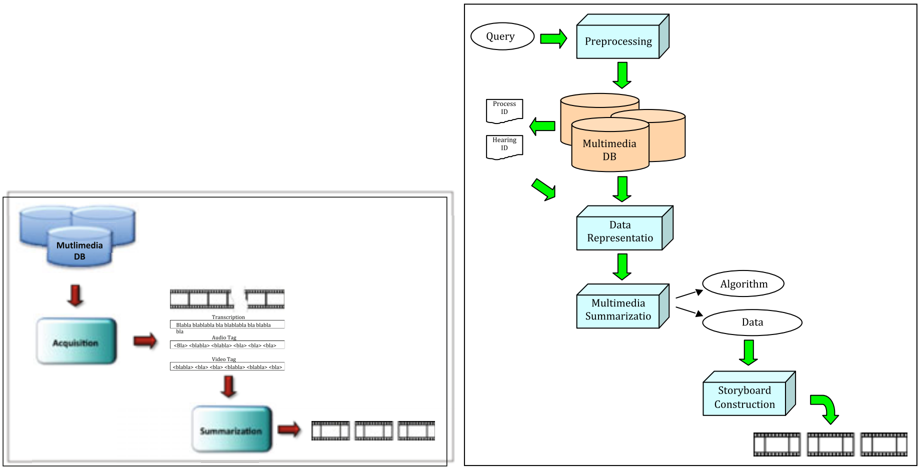
- The *acquisition module*, given a textual query specified by the end user, retrieves multimedia information from the Multimedia Database in terms of audio-video track(s), speech transcription and semantic annotations.
- The *summarization module* is aimed at producing a storyboard by exploiting the information retrieved by the acquisition module. The summary is created by focusing on maximally query-relevant passages and reducing cross-document redundancy.

A simple overview of the modules involved into the multimedia summarization environment is depicted in figure 1 (a).

3 Multimedia Summarization Workflow

In order to summarize a multimedia document according to the user needs, a query statement is specified to start the entire workflow (see figure 1 (b)).

The user query is specified at the graphic interface level, where a list of trials are available, in terms of keywords in which we are interested (whatever is uttered by the involved speaker, the emotional state of actors, etc...).



(a) Overview of the multimedia summarization macro-modules

(b) Overview of the multimedia summarization workflow

Fig. 1. Multimedia Summarizaion Environment

Once the query has been specified, it is submitted to the pre-processing module. The aim of this module is to optimize the user query by eliminating noise and by reducing the size of vocabulary, i.e. stop words removal and stemming are performed to enhance retrieval performance on transcription and annotations.

After the preprocessing activity the query is submitted to the retrieval module, which is aimed at accessing to the multimedia database, in order to identify all the information matching the user query: transcription of the debate, audio annotations and videos annotations. At this level, two possibilities are given to the end user: to summarize an entire trial or only those sub-parts of the proceedings that match the query. In the first case the user query is used to retrieve the multimedia documents related to a trial by executing a high-level skimming of the overall database. After this initial step all the clips of the retrieved hearing are considered for producing the summary. In the second case the query is used to scan the database in a more exhaustive way so that, within a given trial, only the audio, video and textual clips that completely match the user query are retrieved.

In both cases we refer to a (audio, video and textual) clip as a consecutive portion of a debate in which there is one speaker whom is active, i.e. there exist a sequence of words uttered by the same speaker without breaking due to other speakers. Indeed, a clip compreses a textual transcription for each speaker period with the corresponding audio/video tags.

The next step in the multimedia summarization workflow relates to data representation module. The aim of this module is to combine information coming from different sources in order to create a unified representation. This activity is performed through a feature vector representation, where all the information able to characterize the audio, video and textual clip of interest are managed as features and weights. Examples of features exploited by this representation are given by the textual transcription, the audio and video tag, the start and end time of the relevant sub-parts of the debate. In particular, two matrices are defined to be exploited by the summarization module: one matrix

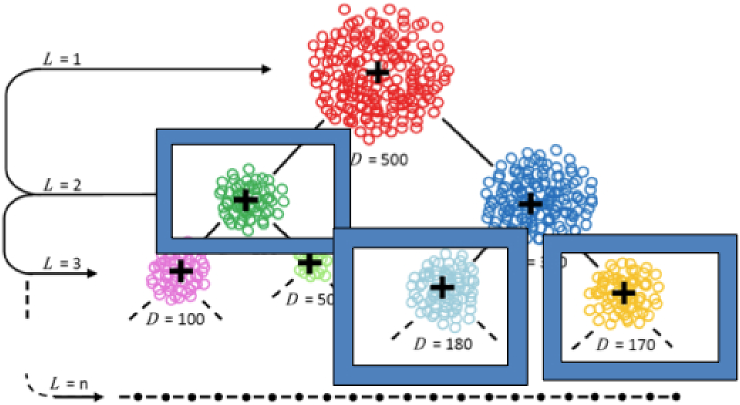


Fig. 2. Dichotimic tree generated by Induced Bisecting K-Means

associated with speech transcription and one matrix associated to the audio and video annotations. The first matrix, defined as numerical, represents textual transcription scoring, obtained through the TFIDF weighting technique [19]. A speech transcription segment associated to a single speaker is mapped into a row, while each term is mapped into a column. The second matrix, defined as binary, represents the presence or absence of a specific audio/video annotation associated to a transcription.

Starting from these two matrices, the multimedia summarization module may start the summary generation. The core component is based on a clustering algorithm named Induced Bisecting K-means [13]. The algorithm creates a hierarchical organization of (audio, video and textual) clips, by grouping in several clusters hearings (or sub-parts of them) according to a given similarity metric. This algorithm is able to build a dichotomic tree in which coherent concepts are grouped together, i.e. each cluster created by the algorithm contains a set of audio, video and textual clips representing similar concepts that are coherent with the user query (see figure 2).

The last step relates to the storyboard construction, where the final storyboard is derived from the dichotomic tree structure produced by the Induced Bisecting K-means algorithm. Given the dichotomic tree, a pruning step is performed in order to choose only those clusters that satisfy a given intra-cluster similarity requirements [20]. Suppose that the pruning activity after the Induced Bisecting K-means returns a set of clusters as reported in figure 3 where C1, C2 and C3 are the resulting clusters and the clips named $1, \ldots, 9$ represent the sub-parts of the debate. The storyboard construction activity considers the representative elements of each cluster (centroids) as the relevant clips for the summary. The storyboard is generated by presenting to the end user the first frame of each centroid, connected to the corresponding audio, video and textual

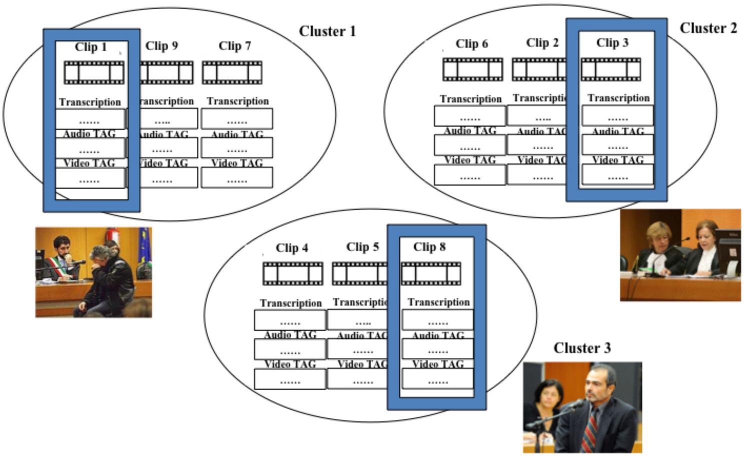


Fig. 3. Clustering output

information, references of the trial/hearing, start and end time of the segments and so on. By referencing figure 3, only the first frames related to segments 1, 3 and 8 (representative of the obtained 3 clusters) are presented to the end user as pictures that could be clicked to start the corresponding audio-video portion.

In the following subsection details about the core component of the multimedia summarization environment, i.e. the Induced Bisecting K-Means clustering algorithm, are given.

4 The Hierarchical Clustering Algorithm

The approaches proposed in the literature for hierarchical clustering were mostly statistical with a high computational complexity . A novel approach, Bisecting k-Means was proposed in [12], has a linear complexity and is relatively efficient and scalable. It starts with a single cluster of multimedia clips and works in the following way:

Algorithm 1. Bisecting K-Means

```
1: Pick a cluster S of clips $m_l$ to split
2: Set K as the number of clusters to be obtained
3: Select two random seeds which are the initial representative clips (centroids)
4: Find 2 sub-clusters S1 and S2 using the basic k-Means algorithm[1].
5: Repeat step 2 and 3 for ITER times and take the split that produces the clustering
   with the highest Intra Cluster Similarity (ICS)[1]
6:     $ICS(S_p) = \frac{1}{|S_p|^2} \sum_{m_i, m_j \in S_p} sim(m_i, m_j)$
7: Repeat steps 1, 2 and 3 until the desired number of clusters is obtained.
```

The major disadvantage of this algorithm is related to the requirements about the specification (a priori) of the parameters K and ITER. An incorrect estimation of K and ITER may lead to poor clustering accuracy. Moreover, the algorithm is sensitive to the noise tath may affect the computation of cluster centroids. Consider for instance N as the number of clips belonging to the cluster p and R as the set of their indices. The j^{th} feature of the cluster centroid - used by the k-Means algorithm during step 3 - is computed as $c_j^p = \frac{1}{N} \sum_{r \in R} m_{rj}$ where m_{rj} is the vectorial representation of the j^{th} feature of the i^{th} clip. Consequently, the centroid c_j^p may contain the contribution of noisy features that the pre-processing phase is not be able to remove. To overcome these two problems we exploit an extended version of the Standard Bisecting k-Means, named Induced Bisecting k-Means [13], whose main steps are described as follows:

[1] The similarity metric is a linear combination of the cosine similarity, for the numerical vectors concerned with transcriptions, and the jaccard similarity, for the binary vectors concerned with audio/video annotations.

Algorithm 2. Induced Bisecting K-Means

```
1: Set the Intra Cluster Similarity (ICS) threshold parameter $\tau$
2: Build a distance matrix A whose elements represents distance between couple of clips[2]
3: Select, as centroids, the two clips $i$ and $j$ s.t. $a_{ij} = \max_{l,m} A_{lm}$
4: Find 2 sub-clusters S1 and S2 using the basic k-Means algorithm
5: Check the $ICS$ of $S_1$ and $S_2$ as
6:    If the ICS value of a cluster is smaller than $\tau$, then reapply the divisive process to this set, starting form step 2
7:    If the ICS value of a cluster is over a given threshold, then stop. 6. The entire process will finish when there are no sub-clusters to divide.
```

The main differences of this algorithm with respect to the Standard Bisecting k- Means consist in: (1) how the initial centroids are chosen: as centroids of the two child clusters we select the clips of the parent cluster having the greatest distance between them; (2) the cluster splitting rule: a cluster is split in two subclusters if the Intra Cluster Similarity is smaller than the threshold value τ. Therefore, no input parameters K and ITER must be specified by the user. Our algorithm outputs a binary tree of clips, where each node represents a collection of similar clips. This dichotomic structure is then processed according to [20], in order to obtain a flat representation of clusters.

In order to perform an initial evaluation of the proposed multimedia summarization environmen, we considered 25 real proceedings characterized by a set of 3825 clips. A first analysis of the summary generated by our approach highlights two main peculiaries: high level of compression of the comprised multimedia documents, where two or three key frames per proceedings have been extracted, and high level of precision, i.e. the generated multimedia summaries contain the most important parts of the considered proceedings.

5 Conclusion and Future Work

In this paper a multimedia summarization environment has been presented in order to allow judicial actors to browse and navigate multimedia documents related to penal hearings/proceedings. The main component of this environment is represented by the summarization module, which creates a storyboard for the end user by exploiting several semantic information embedded into a courtroom recording. In particular, automatic speech transcriptions joint with automatic audio and video annotations have been used for deriving a compressed and meaningful representation of what happens into a law courtroom. Our work is now focused on creating a testing environment for a quality assessment of the produced storyboard.

[2] The distance metric is a linear combination of a cosine-based distance, for the numerical vectors concerned with transcriptions, and the jaccard distance, for the binary vectors concerned with audio/video annotations.

Acknowledgment

This work has been partially supported by the European Community FP-7 under the JUMAS Project (ref.: 214306).

References

1. Kim, J., Chang, H., Kang, K., Kim, M., Kim, H.: Summarization of news video and its description for content-based access. International Journal of Imaging Systems and Technology, 267–274 (2004)
2. Liang, C., Kuo, J., Chu, W., Wu, J.: Semantic units detection and summarization of baseball videos. In: Proc. of the 47th Midwest Symposium on Circuits and Systems, pp. 297–300 (2004)
3. Tjondronegoro, D.W., Chen, Y., Pham, B.: Classification of selfconsumable highlights for soccer video summaries. In: Proc. of the IEEE International Conference on Multimedia and Expo., pp. 579–582 (2003)
4. de Silva, G., Yamasaki, T., Aizawa, K.: Evaluation of video summarization for a large number of cameras in ubiquitous home. In: Proc. of the 13th Annual ACM International Conference on Multimedia, pp. 820–828 (2005)
5. Jaimes, A., Echigo, T., Teraguchi, M., Satoh, F.: Learning personalized video highlights from detailed MPEG-7 metadata. In: Proc. of the IEEE International Conference on Image Processing, pp. 133–136 (2002)
6. Takahashi, Y., Nitta, N., Babaguchi, N.: Video Summarization for Large Sports Video Archives. In: Proc. of the IEEE International Conference on Multimedia and Expo., pp. 1170–1173 (2005)
7. Agnihotri, L., Dimitrova, N., Kender, J.R.: Design and evaluation of a music video summarization system. In: Proc. of the IEEE International Conference on Multimedia and Expo., pp. 1943–1946 (2004)
8. Yang, H., Chaisorn, L., Zhao, Y., Neo, S., Chua, T.: VideoQA: question answering on news video. In: Proc. of the 11th Annual ACM International Conference on Multimedia, pp. 632–641 (2003)
9. Moriyama, T., Sakauchi, M.: Video summarization based on the psychological unfolding of drama. Systems and Computers in Japan, 1122–1131 (2002)
10. Rui, Y., Zhou, S.X., Huang, T.S.: Efficient access to video content in a unified framework. In: Proc. of the IEEE International Conference on Multimedia Computing and Systems, pp. 735–740 (1999)
11. Tseng, B.L., Smith, C.-Y.L.J.R.: Using MPEG-7 and MPEG-21 for personalizing video. IEEE Transactions on Multimedia, 42–52 (2004)
12. Steinbach, M., Karypis, G., Kumar, V.: A comparison of Document Clustering Techniques. In: KDD Workshop on Text Mining (2000)
13. Archetti, F., Fersini, E., Campanelli, P., Messina, E.: A Hierarchical Document Clustering Environment Based on the Induced Bisecting k-Means. In: Larsen, H.L., Pasi, G., Ortiz-Arroyo, D., Andreasen, T., Christiansen, H. (eds.) FQAS 2006. LNCS (LNAI), vol. 4027, pp. 257–269. Springer, Heidelberg (2006)
14. Lf, J., Gollan, C., Ney, H.: Cross-language Bootstrapping for Unsupervised Acoustic Model Training: Rapid Development of a Polish Speech Recognition System. In: Interspeech, Brighton, U.K., September 2009, pp. 88–91 (2009)

15. Falavigna, D., Giuliani, D., Gretter, R., Lf, J., Gollan, C., Schlter, R., Ney, H.: Automatic Transcription of Courtroom Recordings in the JUMAS project. In: Proc. of the 2nd International Conference on ICT Solutions for Justice, Skopje, Macedonia (September 2009)
16. Avgerinakis, K., Briassouli, A., Kompatsiaris, I.: Video processing for judicial applications. In: Proc. of the 2nd International Conference on ICT Solutions for Justice, Skopje (2009)
17. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings of the IEEE 77(2), 257–286 (1989)
18. Fersini, E., Messina, E., Arosio, G., Archetti, F.: Audio-based Emotion Recognition in Judicial Domain: A Multilayer Support Vector Machines Approach. In: Perner, P. (ed.) Machine Learning and Data Mining in Pattern Recognition. LNCS, vol. 5632, pp. 594–602. Springer, Heidelberg (2009)
19. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing & Management 24(5), 513–523 (1988)
20. Kashyap, V., Ramakrishnan, C., Thomas, C., Bassu, D., Rindflesch, T.C., Sheth, A.: TaxaMiner: An experiment framework for automated taxonomy bootstrapping. International Journal of Web and Grid Services 1(2), 240–266 (2005)

Comparison of Redundancy and Relevance Measures for Feature Selection in Tissue Classification of CT Images[⋆]

Benjamin Auffarth[1,⋆⋆], Maite López[2], and Jesús Cerquides[2]

[1] Institute for Bioengineering of Catalonia
C/Baldiri Reixac 4-6 (torre I), 08028 BCN, Spain
[2] Volume Visualization and Artificial Intelligence research group,
Departament de Matemàtica Aplicada i Anàlisi (MAIA), Universitat de Barcelona,
C/Gran Via, 585, 08007 Barcelona, Spain
bauffarth@el.ub.es,{maite,jcerquide}@maia.ub.es

Abstract. In this paper we report on a study on feature selection within the minimum–redundancy maximum–relevance framework. Features are ranked by their correlations to the target vector. These relevance scores are then integrated with correlations between features in order to obtain a set of relevant and least–redundant features. Applied measures of correlation or distributional similarity for redunancy and relevance include Kolmogorov–Smirnov (KS) test, Spearman correlations, Jensen–Shannon divergence, and the sign–test. We introduce a metric called "value difference metric" (VDM) and present a simple measure, which we call "fit criterion" (FC). We draw conclusions about the usefulness of different measures. While KS–test and sign–test provided useful information, Spearman correlations are not fit for comparison of data of different measurement intervals. VDM was very good in our experiments as both redundancy and relevance measure. Jensen–Shannon and the sign–test are good redundancy measure alternatives and FC is a good relevance measure alternative.

Keywords: feature selection; relevance and redundancy; distributional similarity; divergence measure.

1 Introduction

In biomedical image processing, it is difficult to classify organ tissues using shape or gray level information, because image intensities overlap considerably for soft tissue. Hence, features used for processing often go beyond intensity and include something what can be very generally referred to as texture (see [1]). The use of an adequate feature set is a requirement to achieve good classification results.

[⋆] This research was supported by the Spanish MEC Project "3D Reconstruction, classification and visualization of temporal sequences of bioimplant Micro-CT images" (MAT-2005-07244-C03-03).

[⋆⋆] Corresponding author.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 248–262, 2010.

Feature selection generally means considering subsets of features and eventually choosing the best of these subsets. The “goodness“ of feature subsets can be estimated by filters, such as statistical or information theoretic measures, or by a performance score of a classifier (*wrappers*). Currently many approaches to feature selection in bioinformatics are either based on rank filters (univariate filter paradigm) and thereby do not take into account relationships between features, or are wrapper approaches which require high computational costs.

Multivariate filter-based feature selection has enjoyed increased popularity recently [2]. The approach is generally low on computational costs. Filter–based techniques provide a clear picture of why a certain feature subset is chosen through the use of scoring methods in which inherent characteristics of the selected set of variables is optimized. In comparison wrapper-based approaches treat selection as a black-box and optimize the prediction ability according to a chosen classifier.

In feature selection, it is important to choose features that are relevant for prediction, but at the same time it is important to have a set of features which is not redundant in order to increase robustness. [3,4,5,6,7,8,9,10] have elaborated on the concepts of redundancy and relevance for feature selection. [11,4,9] presented feature selection in a framework they call min-redundancy max-relevance (here short mRmR) that integrates relevance and redundancy information of each variable into a single scoring mechanism.

Our data consist of slices in a 3-D volume taken from CT of bones, into which a tracing material was introduced[1]. Fig. 1 shows the alien biomaterial marked in white on the right and organic bone material (referred to henceforth as non-biomaterial in order to distinguish it from the introduced biomaterial). For the classification task, the introduced biomaterial is the target class and relatively small as compared to the non-target class. The volume centers around the introduced material and hence, percentage of biomaterial is greatest in the centers (around 10 percent), becoming less towards the exteriors.

In this article we present an experimental evaluation of several filters for computing redundancy and relevance. In section 2 we will introduce the concepts of redundancy and relevance and compare several measures. We put emphasis on non–parametric filters that are low on computational costs, using very simple density estimation. Section 3 describes experimental methodology and the results are presented in section 4. In section 5 we then discuss and draw some conclusions.

2 Feature Selection with Relevance and Redundancy

In feature selection the aim is to choose a subset of features in order to improve performance and efficiency with respect to a task (in our case classification) and to reduce noise. Information loss in the reduction of the feature space should be kept as small as possible so the resulting space can provide enough information for classification.

[1] Samples from the data set are available on the homepage of one of the authors: `http://www.maia.ub.es/~maite/out-slice-250-299.arff`

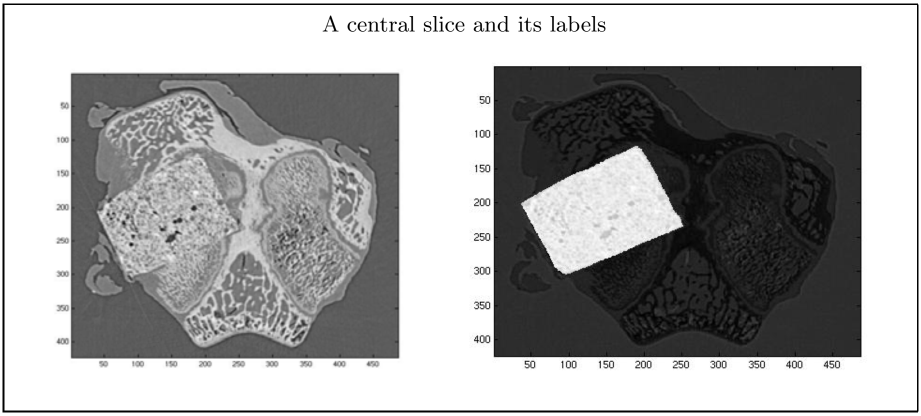


Fig. 1. A central slice in the 3-D volume (left) and its labels (right) marked white

Relevance measures the “goodness“ of the projection from individual attributes to labels. Redundancy measures how similar features are (or inversely, how much adding a feature to a given set of features contributes to prediction[2]). As such, both redundancy and relevance measures fall into the class of measures for *statistical dependence*, *distributional similarity*, or *divergence measure*.[3]

We will now outline several relevance and redundancy criteria based on mutual information, statistical tests, probability distributions, and correlation coefficients. Since we do not know, the true distribution of the data, we prefer non-parametric and model-free metrics. Non-parametric tests have less power (i. e. the probability that they reject the null hypothesis is smaller) but should be preferred when distributions could be non–gaussian. Furthermore non–parametric filters are generally more robust to outliers than parametric tests.

Within the context of feature selection, we will write targets as $Y \in C^N, |C| = d, C = \{c1, ..., c_d\}$ and denote feature i as X_i, with elements x_i^k.

2.1 Relevance Criteria

Relevance is distributional similarity between a continous feature vector and a target vector. In this article we consider the case of two classes (binary classification). Relevance criteria determine how well a variable discriminates between the classes. They are a measure between a feature and the class, i. e.

$$Rel(X, Y) \equiv \text{ how useful is } X \text{ for predicting } Y. \tag{1}$$

[2] Even though, redundancies can be n-ary relations of features, henceforth we will take redundancies to mean binary relations, i. e. between only two features, which is how it was used in [11,4,9].

[3] Shortly, while similarity and divergence are two different concepts, in this context, divergence or distance is taken to mean dissimilarity.

The relevance criteria that we discuss and use later in experiments are:

- Symmetric Uncertainty (SU)
- Spearman rank correlation coefficient (CC)
- Value Difference Metric (VDM)
- Fit Criterion (FC)

Of these, symmetric uncertainty was used before as a relevance criterion [5,7]. Symmetric uncertainty is symmetric and scaled mutual information [12]. Mutual information was used by [4,9] and by [3]. [13] used normalized mutual information for gene selection.

As for Spearman correlations, we did not find a prior publication that refers to it as a relevance criterion, but we thought it might be better to use a non-parametric measure instead of relying on linear correlations (Pearson product–moment correlations), which have been used before as relevance measure [14,8]. We did not use Pearson correlations because of their sensibility to extreme values, their focus on strictly linear relationships, and the assumption of gaussianity. For non-gaussian data rank–correlations should be preferred over Pearson correlations (see [15] on rank correlations and [16] as one of many recommendations to use Spearman correlations instead).

We show how a measure of probability difference, similar to one presented before as the "value difference metric" [17], can be adapted as a relevance criterion. We propose a new measure, which we call "fit criterion" which measures relevance similar to the z-score.

Value Difference Metric. We will refer to $\mathrm{p}(X)$ as the probability function of variable X, $\mathrm{p}([X|Y = c_i])$ as the probability function of X with target $Y = c_i$, and $\mathrm{p}(X = x)$ as the probability density of X at x.

We define a simple, continuous, monotonic function that measures overlap between two variables X_1 and X_2:

$$\left(\int |\mathrm{p}(X_1 = x) - \mathrm{p}(X_2 = x)|^q \, dx\right)^{1/q}, \text{ where } q \text{ is a parameter} \tag{2}$$

We chose $q = 1$, which has been used similarly by [17,18,19] under the name *value difference metric* as distance measure.

Given that the probabilities that X is equal to a given x for all possible values of x is 1, $\int \mathrm{p}(x)\, dx = 1$, total divergence would give the sum

$$\int \mathrm{p}(X_1 = x)\, dx + \int \mathrm{p}(X_2 = x)\, dx = 2 \tag{3}$$

In order to have a range between 0 and 1 we divided by 2. This gives a very intuitive, vertical distance between the probability mass functions.

$$\mathrm{VDM}(X_1, X_2) = \frac{1}{2} \int |\mathrm{p}(X_1 = x) - \mathrm{p}(X_2 = x)| \, dx \tag{4}$$

Our VDM relevance measure is based on the idea that conditional distributions of variables $\{p([X_i|Y = c_j])|j = 1, \ldots, d\}$ should be distinct from each other.

We define VDM relevance (to which we will refer to short as VDM) of a feature X and labels Y with two classes c_1 and c_2 as:

$$\mathrm{VDM}(X,Y) = \frac{1}{2}\int |\mathrm{p}(X = x|c_1) - \mathrm{p}(X = x|c_2)|\; dx \tag{5}$$

Fit Criterion. For a given point x a criterion of fit to one distribution X_1 could be defined as the points distance to the center of the distribution $\overline{X}_1$ in terms of the variance of the distribution var_{X_1}.

$$\frac{\left|x - \overline{X}_1\right|}{\mathrm{var}_{X_1}} \tag{6}$$

where $\overline{X}$ is a center of the distribution (as given e. g. by the mean or median[4]). and var denotes some measure of statistical dispersion, (e. g. the mean absolute deviation from the mean)

A decision criterion for whether a point x belongs to distribution X_1 or to distribution X_2 could be this:

$$\mathrm{FCP}(x, X_1, X_2) = \begin{cases} 1 \text{ if } \frac{\left|x-\overline{X}_1\right|}{\mathrm{var}_{X_1}} < \frac{\left|x-\overline{X}_2\right|}{\mathrm{var}_{X_2}} \\ 2 \text{ if } \frac{\left|x-\overline{X}_1\right|}{\mathrm{var}_{X_1}} > \frac{\left|x-\overline{X}_2\right|}{\mathrm{var}_{X_2}} \end{cases} \tag{7}$$

In the case that both distances were equal we chose arbitrarily.

We refer to FCP as the *fit criterion for a given point.*

More general for k distributions and a feature, this can be expressed as

$$\mathrm{FCP}(x, X) = \arg_{i=1\ldots,k} \min \frac{\left|x - \overline{X}_i\right|}{\mathrm{var}_{X_i}} \tag{8}$$

We now show the derivation of the decision boundary $\dot{x}$ that results from FCP given again two distributions X_1 and X_2. Our decision boundary $\dot{x}$ is at equal distance to both μ_{X_1} in terms of σ_{X_1} and μ_{X_2} in terms of σ_{X_2}.

$$\frac{|\mu_{X_1} - \dot{x}|}{\sigma_{X_1}} = \frac{|\mu_{X_2} - \dot{x}|}{\sigma_{X_2}} \tag{9}$$

We also know that $\dot{x}$ is between μ_{X_1} and μ_{X_2}. We assume $\mu_{X_1} \leq \mu_{X_2}$ and therefore $\mu_{X_1} \leq \dot{x} \leq \mu_{X_2}$ and resolve

$$\dot{x} = \frac{\mu_{X_1}\,\sigma_{X_2} + \sigma_{X_1}\,\mu_{X_2}}{\sigma_{X_2} + \sigma_{X_1}} \text{ (if } \mu_{X_1} \leq \mu_{X_2}) \tag{10}$$

Such decision boundaries ignore many of the characteristics of the distributions, but are unbiased between different distributions, because they do not take into

[4] The choice between mean and median should depend on characteristics of the data and the task. However, as the classical center of gravity the mean is preferable.

account prior class-probabilities. Note that the above expression loses meaning with long tails and with n-modal distributions ($n > 1$).

For the decision, whether x from distribution X_1 belongs to class c_1 or c_2 we write $\mathrm{FCP}(x, [X_1|Y = c_1], [X|Y = c_2])$.

For calculating relevance based on the FCP, we proceed with the conditional distributions $p([X_i|Y = c_1])$, X_i, where corresponding targets are equal to c_1, and for each point $x \in X_i$ we compute the FCP, i. e. the class which point x should belong to according to equation 8. This is then matched with the target labels and the percentage of correct classification by equation 8. We use this as a relevance criterion and call it "fit criterion" (short "FC"). Given data $[X|Y = c_i]$ of features $X = \left\{x_i^j \middle| i = 1 \ldots m, j = 1 \ldots N\right\} \subset \mathbb{R}^{Nm}$, where N is the number of points and m the number of features, and matching class labels $Y = \left\{y^j \middle| j = 1 \ldots N\right\} \in C^N$, we define the relevance fit criterion for binary class labels in Y and some feature X_k as:

$$\mathrm{FC}(X_k, Y) = \frac{1}{n} \sum_{i=1}^{N} 1_{\mathrm{FCP}(x_k^i, [X_k|y^i = c_1], [X_k|y^i = c_2]) = y^i}, \tag{11}$$

where 1 is an indicator function returning 1 (correct) or 0 (incorrect) depending on the correctness of the prediction by FCP. This relevance criterion takes the average accuracy of the separation by the σ–normalized distance from centers of distribution X_k given label c_1 and given label c_2, respectively.

2.2 Redundancy Criteria

Redundancy criteria measure similarity between the distribution of attributes and the distribution of labels[5].

Formally the redundancy between features X_1 and X_2 given class targets $Y \in C^N = \{c_1, \ldots, d\}^N$ can be written as

$$\mathrm{Red}(X_1, X_2, Y) = \frac{1}{d} \sum_{i=1}^{d} \varDelta\left([X_1|Y = c_i], [X_2|Y = c_i]\right), \tag{12}$$

where $[X_1|Y = c_i]$ denotes the distribution of feature 1, given class i (i. e. $\{X_1^l|\forall l, Y^l = c_i\}$, and $\varDelta$ one of the distributional similarity measures that will follow in this subsubsection. There could be more advantageous ways to combine the conditional metrics than the arithmetic mean as in equation 12, but we chose consciously a conservative one.

Relevance and redundancy measures are tests for the goodness-of-fit and as such, we could use similar or even the same functions for measuring redundancy and relevance. Given a relevance measure $Rel()$, features X_1 and X_2, and targets

[5] As such there exists abundant literature on goodness of fit however we did not find comparisons within the context of feature selection for pattern recognition.

$Y \in C^N$, we can define

$$\mathrm{Red}(X_1, X_2, Y) = \frac{1}{d} \sum_{i=1}^{d} \left(\mathrm{Rel}([X_1|Y = c_i], [X_2|Y = c_i])\right). \tag{13}$$

We used these redundancy criteria:

- Kolmogorov-Smirnov test on class-conditional distributions (RKSC)
- Kolmogorov-Smirnov test ignoring classes (RKSD)
- Redundancy VDM (RVDM)
- Redundancy Fit Criterion (RFC)
- Spearman rank correlation coefficients (RCC)
- Jensen-Shannon Divergence (RJS)
- Sign–test (RST)

Redundancy can be measured taking into account classes or without respect to a given class. For purpose of comparison, we include two redundancy criteria that differ only in whether or not they use class information, RKSC and RKSD. We compute all redundancy measures on the class conditional distributions except for RKSD. Recently, Zhang et al. found that taking class–specific correlations they obtained better results.

The Jensen-Shannon divergence is a symmetric and scaled version of the Kullback-Leibler divergence (sometimes: information divergence, information gain, relative entropy, which is an information theoretic measure of the difference between two probability distributions P and Q [20].

We will describe the redundancy VDM and the redundancy fit criterion in the following.

Redundancy Fit Criterion. Equation 11 gives the goodness of fit with respect to two classes, c_1 and c_2, averaged over all points of a feature X_k. The binary sequence behind the sum represents correct class attributions (hits, 1) and incorrect class attributions (misses, 0) for each point of a feature X_k. Let us write the indicator function (and binary vector) corresponding to feature X_k as $\mathrm{hits}_{\mathrm{X_k}} \in \{1, 0\}^N$, where N are the number of points of X_k. We define hits as:

$$\mathrm{hits} \begin{cases} 1 \text{ if } \mathrm{FCP}(x_k^i, \left[X_k|y^i = c_1\right], \left[X_k|y^i = c_2\right]) = y^i \\ 0 \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{14}$$

A very simple similarity measure between two features X_1 and X_2 given their binary sequences hits_{X_1} and hits_{X_2} could be the normalized sum of hits combined by binary operators:

$$\mathrm{RFC}_{X_1,X_2} = \frac{\sum(\mathrm{hits}_{X_1} \wedge \mathrm{hits}_{X_2}) \vee (\neg\mathrm{hits}_{X_1} \wedge \neg\mathrm{hits}_{X_2})}{N} \tag{15}$$

This formula quantifies the percentage of identically classified points. We will refer to this measure as the redundancy fit criterion“ (short “RFC“).

3 Experiments

We benchmarked the feature selection quality resulting from redundancy and relevance information combined by different selection schemes. Additionally we selected features based on unitary filters, i. e. based on either relevance or redundancy. We benchmarked first each relevance and redundancy criterion on its own by unitary filters, then all 28 combinations of mentioned relevance and redundancy measures with different selection schemes and random selection. We selected feature sets of different sizes ($S = [4, 8, 12, 16, 20, 30, 45, 60, 80, 100]$[6]).

We compared five basic feature selection schemes. In the simplest selection scheme, at each iteration we take the most relevant feature and discard all features for which redundancy with the newly chosen features exceeds a threshold. We iterate over these two steps until no features are left. This scheme was presented by [5] and we refer to it henceforth as "Greedy". Varying the redundancy thresholds we obtain a different number of features. As for the second selection scheme we order features by either $\frac{\text{rel}}{\text{red}}$ or $\text{rel} - \text{red}$ and choosing the first s. This schemes, presented by [9,4] were called minimum redundancy maximum relevance quotient (mRmRQ) and minimum redundancy maximum relevance difference (mRmRD), respectively. In [21] we presented a selection scheme based on an attractor network, which was thought to be capable of integrating more complex redundancy interactions between features (henceforth called Hopfield) and was comparable in performance to the mRmR framework. This is our third selection scheme. As our last selection scheme we rely on unitary filters which means either only relevance or only redundancy was taken into account. For the relevance case, the s most relevant features were used, and for the redundancy case, starting from the complete set of features, at each step the most redundant feature is removed until the desired numbers of features s are left. At last, we also compared a baseline of random selection.

We applied three classifiers for benchmarking. These were Naïve Bayes, GentleBoost, and a linear Support Vector Machine. As for Naïve Bayes we relied on our own implementation for multi-valued attributes using 100 bins. For GentleBoost we used 50 iterations. For SVM classification, we used libsvm [22]. Features were z-normalized and the cost function was made to compensate for unequal class priors, i. e. the weight of the less frequent class was set to $\max\left(\frac{\sharp(Y=c_2)}{\sharp(Y=c_1)}, \frac{\sharp(Y=c_1)}{\sharp(Y=c_2)}\right)$. We set the SVM complexity parameter C to 1 which seemed to be a good choice and in the right order of magnitude.

At each number of features – in order to have many validations at acceptable speed – we made 10 random samplings of size $n/10$ and for each sampling we did 5-fold cross-validation. As for random feature selection, we did 10 random samplings of the data of size $n/10$ and tested 10 random selections of features in 5-fold cross-validation.

The complete feature set consisted of 127 features. We included 10 features from the Laplacian Pyramid [23], 100 Gabor features [24] in 10 orientations

[6] This choice expresses an emphasis on feature sets of sizes ≤ 30 because that was were they were the greatest differences between the different methods.

and 10 scales, 9 features from luminance contrast [25], 7 features from texture contrast [26], and intensity.We added 50 useless variables (probes) which good feature selection methods should eliminate. 49 of these probes were random variables. 25 of those standard normal distributed, 24 uniformly distributed in the interval $(0, 1)$. The last probe was a variable of zeros.

The experiments and comparisons following in this section are therefore based on a set of 177 features and their respective relevance measures and mutual redundancies. Details on the methods can be found in [27].

4 Results

Within the scope of this article we focus on these questions:

1. What are the best measures of relevance and redundancy (RR)?
   (a) What is the best redundancy and relevance (RR) combination?
   (b) What is the best redundancy measure?
   (c) What is the best relevance measure?
2. Do class-conditional distributions give better redundancy estimations?

Question 1 concerns comparisons of relevance and redundancy measures. In particular this concerns comparisons of combinations of redundancy and relevance measures, and of redundancy measures and relevance measures, respectively, among themselves.

In subsection 2.2 we proposed to calculate redundancy criteria based on class-conditional distributions. As for question 2, we want to resolve whether this made sense, looking at RKSC and RKSD redundancy criteria which only differ in using class-conditional distributions and total distributions.

4.1 Statistical Evaluation

We used AUC as our performance measure. Following the recommendations of [28] we did not base our statistics on performances of single folds but took averages (medians[7]) over folds.

In table 1 redundancy and relevance combinations are compared over all classifiers, all numbers of features, and mRmRQ, mRmRD, and Hopfield. Tables 2 and 3 analyze redundancy measures and relevance measures, respectively, over

[7] According to the central limit theorem, any sum (such as e. g. a performance benchmark), if of finite variance, of many independent identically distributed random features will converge to a Gaussian distribution. This is however not necessarily to expect for only 5 values, i. e. from 5-folds of cross-validations. After finding partly huge differences between means and medians over cross-validations, in pre-trial runs, we decided to take the more robust median (which in case of normal distributions is equal to the arithmetic mean anyway). As for the error-bar, we plot the interquartile range (short: IQR), which is the difference between values at the first (25%) and the third quartile (75%).

all classifiers, numbers of features, mRmRQ/D, and Hopfield, and relevance or redundancy measures, respectively.

A difficulty with regard to the Greedy method is that it produces feature sets with an unpredictable number of features. We included all Greedy schemes in number-of-features specific comparison tables using a threshold of $\frac{|s_{\text{design}} - s_{\text{Greedy}}|}{s_{\text{design}}} \leq 0.1$.

We now explain the format of the result tables. The first column gives the name of the method, specified by selection scheme, redundancy, and relevance measures[8]. The second column indicates the rank of the method within methods compared in the same table. Ordering follows by mean rank of performance (third column). Median performance and interquartile range of the vector of performance scores (columns four and five) served for statistical comparisons by Friedman test and Nemenyi post-hoc test (F/N), and Wilcoxon Signed Rank Test (SR). One-to-one comparisons of methods by these statistical tests can be found in columns six and seven as win and loss scores (W/L) indicating statistical significance.

4.2 Redundancy and Relevance Measures

In table 1 you can find a ranking of RR combinations over all numbers of features and over mRmRQ/D and Hopfield.

The best combination was RVDM with FC. The table shows nearly coherent groupings by relevance measure. Everything including SU is clearly on the bottom. Also bad, but better than SU we find combinations with CC relevance. RFC and RCC redundancy seem worse than others, with RCC having greater deviation. A good redundancy measure seems to be RVDM.

In table 2 we see rankings of redundancy measures averaged (medians) over mRmRQ/D and Hopfield over all numbers of features. Here the clear winner is RJS, followed by RVDM and RST together with highly correlated RKSC and RKSD. RFC comes last, after RCC. Both had been only low correlated to the other measures (and higly negatively with each other).

A comparison of relevance measures we find in table 3. The statistics are again over mRmRQ/D and Hopfield and over all numbers of features. VDM and FC, which had been found highly correlating, are clearly the best relevance measures. CC comes before SU, which is the clear looser.

4.3 Class-Conditional Distributions

We used two redundancy criteria based on the Kolmogorov-Smirnov (KS) test, RKSC and RKSD. RKSD was computed based on the total distributions and RKSC on the class-conditional distributions, i. e.

$$RKSD(X_1, X_2) = KS(X_1, X_2) \tag{16}$$

[8] In the case of table 1 the average is taken over selection scheme.

Table 1. RR Combinations over mRmRQ/D and Hopfield, and over all Numbers of Features

	index	mean rank	median	iqr	F/N W/L	SR W/L
RVDM+FC	1	7.85	0.97	0.04	17/0	25/0
RCC+VDM	2	7.99	0.97	0.03	17/0	25/1
RVDM+VDM	3	8.38	0.97	0.04	18/0	24/0
RJS+VDM	4	8.63	0.97	0.04	17/0	24/1
RJS+FC	5	9.63	0.97	0.04	14/0	19/4
RKSC+VDM	6	9.89	0.96	0.07	17/0	19/4
RST+VDM	7	10.07	0.97	0.04	16/1	20/4
RKSD+VDM	8	10.14	0.96	0.07	16/2	15/6
RFC+VDM	9	10.39	0.97	0.03	16/2	16/4
RKSC+FC	10	10.68	0.96	0.09	16/0	14/7
RKSD+FC	11	10.73	0.96	0.09	16/1	14/7
RST+FC	12	11.07	0.97	0.08	15/3	16/7
RCC+FC	13	13.72	0.96	0.07	8/10	10/12
RJS+CC	14	13.94	0.96	0.05	9/12	13/10
RFC+FC	15	13.95	0.96	0.07	9/9	9/13
RST+CC	16	14.10	0.95	0.06	8/11	11/13
RVDM+CC	17	14.52	0.96	0.06	8/12	10/15
RCC+CC	18	15.42	0.95	0.05	7/14	10/8
RKSC+CC	19	15.62	0.95	0.08	9/13	9/17
RKSD+CC	20	16.08	0.95	0.08	8/14	8/18
RFC+CC	21	17.52	0.94	0.04	7/17	7/20
RJS+SU	22	17.92	0.94	0.09	6/21	6/21
RVDM+SU	23	19.60	0.87	0.16	3/22	3/22
RST+SU	24	21.28	0.88	0.13	4/23	4/23
RKSC+SU	25	21.51	0.86	0.15	3/23	3/23
RKSD+SU	26	21.98	0.86	0.15	2/24	2/24
RFC+SU	27	26.62	0.79	0.18	0/26	1/26
RCC+SU	28	26.79	0.84	0.17	0/26	0/27

Table 2. Redundancy over mRmRQ/D and Hopfield, and all Numbers of Features

	index	mean rank	median	iqr	F/N W/L	SR W/L
RJS	1	2.68	0.97	0.04	5/0	6/0
RVDM	2	2.94	0.96	0.04	3/0	5/1
RST	3	3.82	0.96	0.07	2/2	3/2
RKSC	4	4.23	0.95	0.08	2/2	2/3
RKSD	5	4.38	0.95	0.08	1/3	1/4
RCC	6	4.54	0.95	0.06	0/2	0/2
RFC	7	5.40	0.95	0.05	0/4	0/5

Table 3. Relevance over mRmRQ/D and Hopfield, and all Numbers of Features

	index	**mean rank**	**median**	**iqr**	**F/N W/L**	**SR W/L**
VDM	1	1.49	0.97	0.04	2/0	3/0
FC	2	1.88	0.97	0.06	2/0	2/1
CC	3	2.76	0.95	0.04	1/2	1/2
SU	4	3.86	0.86	0.12	0/3	0/3

and

$$RKSC(X_1, X_2, Y) = \frac{1}{d}\sum_{i=1}^{d} KS([X_1|Y = c_i], [X_2|Y = c_i]), \tag{17}$$

where KS refers to the p-values of the KS test.

We had introduced both RKSC and RKSD in order to test, whether it is better to use class-conditional distributions for redundancy estimation. They had a Spearman correlation coefficient of 0.96.

Table 2 shows the small difference between the two measures could have made a difference in performance with RKSC performing better than RKSD. The difference in performance is statistically significant according to the Wilcoxon test, but not significant according to the stricter Friedman and Nemenyi tests. We can conclude that estimations based on class-conditional distributions serve equal or better for redundancy measures than estimations based on the distribution totals.

5 Conclusions

In this article, we presented a framework for measuring redundancy and relevance of features and compared several measures. We present several measures of redundancy and relevance within this framework, including VDM and the fit criterion (FC) which helped us to select a feature set for our classification task. As for relevance and redundancy measures, while there cannot be any single universally best measure for all applications, we hope that our experimental comparison can give some hints as to the applicability and usefulness of some measures.

The comparison of redundancy measures and as well of relevance measures is complicated because of different scales and different levels of distinction For example, the KS-test gave very few different values, while RFC gave a broad variety of different values. Relevance measures differ greatly with respect to the importance they assign to different features. VDM and FC, and SU and CC demonstrated large correlations ($\rho > 0.65$). Relevance measures seem to concur on the relevance on some features, however there are huge differences with respect to others. In particular, we observed that CC and SU attribute lower relevance to some Gabor filters than to some probes. RKSC and RKSD (unsurprisingly because they are so similar) were found very highly correlating with one-another ($\rho = 0.96$). Both of them also were highly correlated with

RST($\rho > 0.8$). RCC correlated negatively with some measures, most markedly with RFC ($\rho = -0.61$).

As for the redundancy measures, the Jensen-Shannon Divergence, RVDM, and the sign-test were good. RFC which is based on the relevance measure FC may have been too simple. There are other options for redundancy fit criterion, for example, quantifying only the number of incorrectly classified points. Better options could also instead of binary sequences $\mathrm{hits}_{\mathrm{X_k}}$ involve continuous values between 0 and 1 that express confidence of assignment.

As for symmetric uncertainty, we did not optimize the density estimation beforehand and took the most simple and straightforward means we could imagine and which worked fine for the naive Bayes. We think that this density estimation affected SU. We concede that a more careful treatment may be necessary.

Of the other relevance measures, VDM and the fit criterion were the best. CC suffered from that it favored the zero-feature. Because of the formulation, Spearman rank correlation coefficients are unsuitable for comparisons between distributions with highly unequal scales, such as the case for comparing classes (set cardinality 2) and continuous features. The Pearson correlation coefficient suffers the same weakness [29]. We expect, the Kendall rank correlation coefficient (see [30]), another much used rank correlation, to have similar problems in dealing with distributions. Other correlation measures could bring an improvement, such as possibly [31].

RVDM and RFC performed very good as unitary filters. Integration of SU makes performance degrade in many cases with a given redundancy measure when compared to other relevance measures. RCC is a bad measure for redundancy; performance was worst when using only RCC (Red:RCC) and any information helped improve performance. RKSD was also bad, RKSC slightly better. Over the different integration schemes, the measures for redundancy and relevance differed in their contribution.

We computed normalized frequencies of probes for selection based on either only relevance or only redundancy (not shown). As for relevance measures, VDM and FC came before CC and SU (which corresponds to their performance ranking). As for redundancy measures, RCC lets slip in many probes, which seems to have caused the mediocre performances with RCC redundancy. RFC and RJS also were more tolerant to probes.

References

1. Vyas, V.S., Rege, P.: Automated texture analysis with gabor filters. GVIP Journal 6(1), 35–41 (2006)
2. Saeys, Y., Inza, I. n., Larrañaga, P.: A review of feature selection techniques in bioinformatics. Bioinformatics (August 24, 2007)
3. Mundra, P.A., Rajapakse, J.C.: SVM-RFE with Relevancy and Redundancy Criteria for Gene Selection. In: Rajapakse, J.C., Schmidt, B., Volkert, L.G. (eds.) PRIB 2007. LNCS (LNBI), vol. 4774, pp. 242–252. Springer, Heidelberg (2007)
4. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. 27(8), 1226–1238 (2005)

5. Duch, W., Biesiada, J.: Feature selection for high-dimensional data: A kolmogorov-smirnov correlation-based filter solution. In: Kurzynski, M., Puchala, E., Wozniak, M., Zolnierek, A. (eds.) Advances in Soft Computing, pp. 95–104. Springer, Heidelberg (2005)
6. Novovicová, J., Malík, A., Pudil, P.: Feature selection using improved mutual information for text classification. In: International Workshop on Structural and Syntactic Pattern Recognition (2004)
7. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. J. Mach. Learn. Res. 5, 1205–1224 (2004)
8. Knijnenburg, T.A.: Selecting relevant and non-relevant features in microarray classification applications. Master's thesis, Delft Technical University, Faculty of Electrical Engineering, 2628 CD Delft (2004)
9. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. In: Second IEEE Computational Systems Bioinformatics Conference, pp. 523–529 (2003)
10. Zhang, Y., Callan, J., Minka, T.: Novelty and redundancy detection in adaptive filtering. In: SIGIR 2002: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 81–88. ACM, New York (2002)
11. Zhou, J., Peng, H.: Automatic recognition and annotation of gene expression patterns of fly embryos. Bioinformatics 23, 589–596 (2007)
12. Witten, I., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques (2005)
13. Liu, X., Krishnan, A., Mondry, A.: An entropy-based gene selection method for cancer classification using microarray data. BMC Bioinformatics 6 (2005)
14. Yu, L., Liu, H.: Feature selection for high-dimensional data: A fast correlation-based filter solution. In: ICML, pp. 856–863 (2003)
15. Conover, W., Iman, R.: Rank Transformations as a Bridge Between Parametric and Nonparametric Statistics. AM. STAT. 35, 124–129 (1981)
16. Wu, G., Twomey, S., Thiers, R.: Statistical Evaluation of Method-Comparison Data. Clinical Chemistry 21, 315–320 (1975)
17. Stanfill, C., Waltz, D.: Toward memory-based reasoning. Communications of the ACM 29(12), 1213–1228 (1986)
18. Wilson, D.R., Martinez, T.R.: Improved heterogeneous distance functions. Journal of Artificial Intelligence Research 6(6), 1–34 (1997)
19. Payne, T.R., Edwards, P.: Implicit feature selection with the value difference metric. In: European Conference on Artificial Intelligence, pp. 450–454 (1998)
20. Lin, J.: Divergence measures based on the shannon entropy. IEEE Transactions on Information Theory 37, 145–151 (1991)
21. Auffarth, B., López-Sánchez, M., Cerquides, J.: Hopfield Networks in Relevance and Redundancy Feature Selection Applied to Classification of Biomedical High-Resolution Micro-CT Images, Petra Perner (2008)
22. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
23. Burt, P.J., Adelson, E.H.: The laplacian pyramid as a compact image code. IEEE Trans. Communications 31, 532–540 (1983)
24. Kovesi, P.D.: Edges are not just steps. In: Proceedings of the Fifth Asian Conference on Computer Vision, pp. 822–827 (2002)
25. Reinagel, P., Zador, A.: Natural scene statistics at center of gaze. Network: Comp. Neural Syst. 10, 341–350 (1999)

26. Einhäuser, W., Kruse, W., Hoffman, K.P., König, P.: Differences of monkey and human overt attention under natural conditions. Vision Research 46(8-9), 1194–1209 (2006)
27. Auffarth, B.: Classification of biomedical high-resolution micro-ct images for direct volume rendering. Master's thesis, University of Barcelona, Barcelona, Spain (2007)
28. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
29. Bollen, K., Bollen, K.: Structural equations with latent variables. Wiley, New York (1989)
30. Abdi, H.: The Kendall Rank Correlation Coefficient. In: Salkind, N.J. (ed.) Encyclopedia of Measurement and Statistics (2007)
31. Yilmaz, E., Aslam, J., Robertson, S.: A new rank correlation coefficient for information retrieval. In: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 587–594. ACM, New York (2008)

Quantile Regression Model for Impact Toughness Estimation

Satu Tamminen, Ilmari Juutilainen, and Juha Röning

University of Oulu, Intelligent Systems Group, PL 4500, 90014 University of Oulu, Finland

Abstract. The purpose of this study was to develop a product design model for estimating the impact toughness of low-alloy steel plates. The rejection probability in a Charpy-V test (CVT) is predicted with process variables and chemical composition. The proposed method is suitable for the whole production line of a steel plate mill, including all grades of steel in production. The quantile regression model was compared to the joint model of mean and dispersion and the constant variance model. The quantile regression model proved out to be the most effective method for modelling a highly complicated property at this extent.

Next, the developed model will be implemented into a graphical simulation tool that is in daily use in the product planning department and already contains some other mechanical property models. The model will guide designers in predicting the related risk of rejection and in producing desired properties in the product at lower cost.

Keywords: MLP, quantile regression, Charpy-V test, product design.

1 Introduction

Manufacturers in steel industry try to improve their competitiveness with different strategies. Small manufacturers may choose to compete with high quality and short delivery time instead of a large volume, but as a result the process control is more difficult to maintain to gain optimal properties to the product. The goal is to reduce the process variability, and it can be achieved with the assistance of design models for products or production. For a complicated phenomena it is much easier to build models that concern only specific products or subgroups of the production. However, the interpolation capability of these models is inadequate. Much more useful model can be achieved by including the whole process to modelling and this way allowing the transfer of information between products. It is much more difficult to build a model for the whole process, but there are powerful data mining methods that can help to achieve this goal.

Typically, one steel plant can have hundreds of products, and yet, the customers might make enquiries of new ones. The product design department will benefit from a model that has good interpolation capabilities, when responding to the customers' quality requirements. Rejections in qualification tests are very expensive for the company. As a result, the motivation to reduce the number of rejected plates has aroused interest to develop models that provide help for process planning.

Impact toughness (or notch toughness) is steel's mechanical property that describes how well does the steel resist fracturing at predefined temperature, when hard impact

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 263–276, 2010.

suddenly hits the object. The property is crucial for steel products that are used in cold and harsh environments *e.g.* ships, derricks and bridges. Qualification of the impact toughness requirements of a steel plate is verified with a Charpy-V test (CVT), which is a cost-effective material testing procedure. [1],[2] The test is performed on three different samples from every test unit, and the plate is accepted if the average of the measurements is higher than the requirement and none of the measurements is 30% below the requirement. When a product is designed, the risk of rejection in the CVT should be minimized. A model for rejection probability prediction would help in achieving this goal, but the modelling task is not easy. Three measurements should be transformed to one target variable that would preserve the information about the uncertainty caused by scattered measurements.

Transition behaviour is typical for ferritic steel qualities [3]. However, the impact toughness of these qualities can be affected by chemical composition and thermomechanical treatments. The effect of carbon concentration on transition behaviour is illustrated in Fig.1. Steel is ductile at higher temperatures (the area is called the upper shelf) and it gets brittle at low temperatures (the lower shelf). The transition temperature is determined from the average of the upper and lower shelves. When the carbon concentration is low, the transition region from ductile to brittle is narrow, the upper shelf is high, and the slope between the shelves is steep. An increase in the carbon concentration lowers the upper shelf and also widens the transition region. The effect of other alloying elements and process parameters on transition behaviour is similar (or reversed, if the parameter has a positive effect on impact toughness). The complicated interactions between these factors bring a challenge to modelling, as elements that are harmful alone can produce a desirable effect together with another component (*e.g.* nitrogen and aluminium).

Steel's behaviour in the transition region brings uncertainty to modelling, because in this area the force required to break the test bar can vary dramatically. The measurements can show upper shelf energy, lower shelf energy or something in between. Factors that raise the transition temperature (*e.g.* carbon, nitrogen, grain size) have a negative effect on impact toughness, as well. Furthermore, if the grain size is not uniform in the product, the transition temperature is affected by the biggest grain size instead of the average grain size. At room temperature steel can perform well in the impact toughness

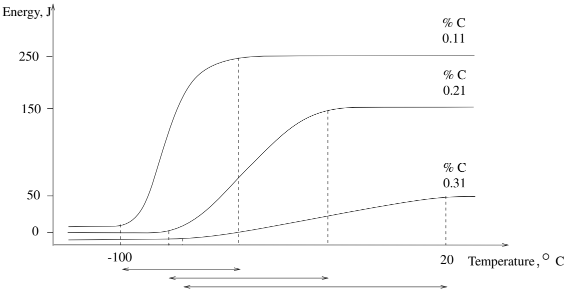


Fig. 1. Effect of carbon concentration on transition behaviour (temperature on the x-axis and absorbed energy on the y-axis)

test, but when the temperature falls, its performance weakens. [3] In this study, the most demanding steel qualities are tested at temperatures as low as -100°C.

Industrial process data is often heteroscedastic and non-Gaussian. In other words, it is not rare that the noise process of the model is input-dependent or that the dependent variable may be highly skewed. If the model simply estimates the conditional mean of the target data by minimizing the sum of squared errors (SSE) function, and ignores these conditions, the performance will be poor. Furthermore, when predicting extreme events, as the rejection in quality test, the conventional SSE-model will consistently under-predict these events. [4]

There are several possibilities to take into account this predictive uncertainty. Dispersion model joins together the mean and variance models and takes into account the heteroscedasticity. The quantile regression model enables to include the form of the distribution into prediction.

In the literature, the most common concern in impact toughness modelling is behaviour inside the transition region (the upper and lower shelf energies, the slope between them, and the ductile-to-brittle transition temperature) [5],[6]. Charpy-V modelling of weld seams is probably the most widely studied application area [7].

Both neural networks and traditional regression methods have been used in modelling, [8] but only a few of the studies have concentrated on how to predict the risk of disqualification and how to handle three measurements of every test unit. Golodnikov *et al.* used a quantile regression model for CVT modelling with a small data set. The test was performed on only one grade of steel and only at one test temperature. [9] The study of Golodnikov *et al.* did not consider the rejection probability in CVT, but the focus was on predicting the three CVT measurement values from the CVT distribution.

Quantile regression has been widely used by economists and ecologists, and for medical applications, survival analysis, financial economics, environmental modelling and detection of heterostcedasticity [10],[11]. Especially economical applications may involve very large data sets, but the method is rarely used for industrial applications. Nevertheless, many applications may benefit more from the estimation of extreme values instead of the mean. Methods for joint modelling of mean and dispersion in different industrial applications have been studied widely, [12],[13],[14] and the typical method is to perform heteroscedastic regression with generalised linear models (GLM).

In this study the goal was to develop a product design model for all grades of steel and test temperatures in production, so that the model can be utilized to predict the related risk of rejection in the CVT.

2 Joint Modelling of Mean and Dispersion

When not only the mean but also error variance is dependent on the explanatory variables, the assumption of constant variance is not satisfactory. If there were information about dispersion, distribution of the mean could be predicted in greater detail. [15],[16]

Dispersion modelling has been employed to analyze quality improvement experiments designed to find the process settings that minimize variance under given conditions. Models of dispersion can be used in tolerance design, for example, because the model points out the sources that produce variation in the process. [12]

The purpose of variance function estimation is to model the structure of the variances as a function of predictors [13]. The most common responses for variance modelling are $\hat{\epsilon}_i^2 = (y_i - \hat{\mu}_i)^2$ and $log\hat{\epsilon}_i^2$. In a case of a normally distributed response, squared residuals are suitable because of the result

$$\epsilon \sim N(0, \sigma^2) \Rightarrow \epsilon^2 \sim \mathrm{Gamma}(\sigma^2, 2), \tag{1}$$

where notation $y \sim \mathrm{Gamma}(\mu, s)$ means that y is Gamma- distributed with expectation μ and variance $s\mu^2$. When the response $log\hat{\epsilon}_i^2$ is used, the model is fitted using least squares. [15]

3 Quantile Regression Model

Because for some products the CVT measurements can be highly scattered, the rejection model should be able to recognize the form of the distribution. The CVT has two different rules for rejection and the impact toughness model should be able to recognize both of them. In some cases the model with average of the measurements fails to recognize the increased risk of rejection. It is much more informative to focus on the lower quantiles of the probability distribution of the measurements instead.

Quantile regression is a method that enables the estimation of conditional quantiles of a response variable distribution, and therefore provides information of not just the location of the distribution but also the shape. The model allows heteroscedasticity to appear in the data, and it is suitable with non-Gaussian distributions. Furthermore, the estimated median provides more robustness to large outliers than the ordinary least squares regression estimate of the mean. [10]

When estimating the sample mean with the ordinary least squares regression the objective is to minimize a sum of squared residuals, whereas the median is estimated by minimizing the sum of absolute residuals. Similarly, with quantile regression the objective is to minimize a sum of asymmetrically weighted absolute residuals, where positive and negative residuals are weighted differently. [17]

4 The Model

Multilayer perceptron networks (MLP) are commonly used for complex and nonlinear system modelling. In a basic form the MLP networks are used for estimating the sample mean, but they are suitable for fitting the quantile regression curves, as well. It has been proved that a network with one hidden layer and sigmoid activation functions can approximate any smooth function. However, sometimes two hidden layers are needed if discontinuities exist in the modelled function. This complicates optimization of the network, and initialization of the network will have a high impact on performance. [18],[19]

Two separate models were trained for the average of three CVT measurements (referred to as AVG_J for the joint model and AVG_C for the constant variance model, and AVG if the CVT level is considered before rejection probability calculation). The quantile regression model is referred to as QR.

In this study, the proposed AVG_J model is

$$\begin{aligned} \mu_i &= f(x_i, \beta) \\ \sigma_i^2 &= g(x_i, \mu_i, \tau) \\ y_i &= \mu_i + \sigma_i \epsilon_i \\ \epsilon_i &\sim N(0, 1). \end{aligned} \tag{2}$$

The functions f and g were modelled using MLP networks with model parameters β and τ.

The response of the mean model was used as one of the explanatory variables of the deviation model. The response for the deviation model was $\log \hat{\epsilon}^2$, which guarantees that the estimated variances are positive [13]. The predicted log-scale variance cannot be utilized without correction, because $E \log \epsilon_i^2 \neq \log \sigma_i^2$, and thus

$$\hat{\sigma}_i^2 = e^{\widehat{\log \epsilon_i^2} + 1.27}, \tag{3}$$

where $\widehat{\log \epsilon_i^2}$ is the prediction for the logarithm of the squared residual for the ith observation. [20],[15],[16] The estimated variance in equation (3) is used in rejection probability calculation. For AVG_C model $\sigma_i^2 = \sigma^2 \; \forall i$, where σ^2 is the sample variance.

The θ^{th} quantile of variable y is the value of $Q(\theta)$, for which $\mathrm{P}(y < Q(\theta)) = \theta$, and the conditional nonlinear quantile function is

$$Q_{y_i}(\theta | x_i) = g(x_i, \beta_\theta) \tag{4}$$

where β_θ is a vector of parameters dependent on θ. The full probability distribution of y can be approximated with quantile regression models corresponding to a range of values of θ $(0 < \theta < 1)$. [10]

When training a neural network model by minimizing a mean squared error (MSE) criterion, an estimation of the conditional expectation of the desired response is achieved.

$$\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{5}$$

Now, the conditional θ^{th} quantile will be produced if the optimization criterion is

$$\frac{1}{N} \sum_{i=1}^{N} [\delta(\hat{y}_i > y_i)(1 - \theta)(\hat{y}_i - y_i) + \delta(y_i > \hat{y}_i)\theta(y_i - \hat{y}_i)] \tag{6}$$

where $0 < \theta < 1$, $\delta(\hat{y}_i > y_i) = 1$ if $\hat{y}_i > y_i$, and $= 0$ otherwise. If $\theta = 0.5$, the conditional median quantile model will be produced. [21]

5 Utilizing the Model for Product Design

Based on customer's enquiry, the manufacturer should decide, what is the most economical way to produce the required properties to the product. Furthermore, when a steel manufacturer has hundreds of products and the volumes of the orders are small, inevitably, several products need to be manufactured from one melting. The model will help in designing a melting that is suitable for several products with their special quality requirements.

The product design group plans the chemical composition of the melting, possible treatments during melting (*e.g.* vacuum degassing) and some production requirements for heating, rolling and thermomechanical treatments for each product. The model will guide designers not to use too much working allowance when keeping the product within tolerances, and thus, to assure desired properties in the product at lower cost. If the steel mill has only a few products and a large volume of each grade, optimization of the process parameters is an easier task than in the case where the mill competes with high quality, short delivery time and a large product range. In the latter case, tools that help product design will directly reduce the number of rejections and delays.

Because the distribution of the standardized residuals was assumed to be normal for the AVG models, the probability of rejection can be calculated with

$$P_i = \Phi\left(\frac{L - \hat{\mu}_i}{\hat{\sigma}_i}\right) \tag{7}$$

where Φ is the cumulative normal distribution function, L is the requirement level, $\hat{\mu}_i$ is the impact toughness estimate and $\hat{\sigma}_i$ is the predicted deviation for observation i. The rejection probability risk level that will lead to a change in product design should be low enough to recognize the unsuccessful plates and high enough not to produce too many false alarms. Typically, $P_i = 0.05$ is used for most of the products.

To utilize the rejection probability information, quantile regression models are needed only for few desired probability levels. The model corresponding to the median is informative for the user, as well. Typically, quantiles $q = 0.05$ and $q = 0.01$ are sufficient levels for rejection probability for the whole production planning. The levels are selected, as earlier, by optimizing the true positive/false alarm -ratio. The CVT requirement of the product is compared with the estimated value of the quantile model, and model $q = 0.05$ indicates that 95% of the plates will have higher CVT value than estimated. Thus, the quantile will act as 0.05 rejection probability limit.

6 Data Collection

The data were collected from a Ruukki Metals, Raahe Works, Finland steel plate mill process in 2002-2008 and they consist of information on nearly 300,000 low-alloy steel plates and over 70 variables with relation to nearly 90% of the steel grades in production of the mill. Of the plates rejected after the CVT, 63% were rejected because of one too-low measurement and the rest because the average of the measurements was too low. The CVT was performed on a majority of the steel plates at −20°C, but the test area varied from +20°C to −100°C.

Careful pre-processing was performed in order to exclude defective observations and redundant variables, for example unnaturally low measurements and incorrectly performed tests. In industrial processes many variables can be highly correlated, but some of them are more reliable than others. With self organizing maps (SOM) the similar variables were recognized, and the most reliable ones were selected for modelling. The final data included 247,852 observations and 43 variables. The data were normalized into a range of $[-1, 1]$ before training.

7 Results

The MLP networks of the product design models were implemented with Matlab R2007a. The data were divided into training (50%), validation (25%), and test sets (25%). The independence of the data sets was verified by not allowing plates from the same melting to belong to different sets.

A resilient back-propagation algorithm with early stopping regularization was used for training, and hundreds of networks of different sizes with random initialization were trained. The best quantile regression network for CVT rejection prediction had two hidden layers with 54 and 14 neurons (the 0.05^{th} quantile) and for CVT level (the 0.5^{th} quantile) 59x24 neurons. The best AVG network for CVT had two hidden layers with 48x17 neurons and for variance 37x8 neurons.

In order to compare the performance of three models throughout the whole probability space, 20 quantile models from 0.00001^{th} to 0.999^{th} were trained, and the corresponding rejection probability levels were obtained.

If the results are viewed only from the CVT level prediction's point of view, the AVG model seem to work better. The mean squared error (MSE) as well as the mean absolute error (MAE) and the correlation between target and predicted value are better. The results of the models can be seen in Table 1.

Table 1. Results of the training data and the independent test set

	train	test
$\mathrm{MSE}_{\mathrm{AVG}}$	944.1	1001.1
$\mathrm{MAE}_{\mathrm{AVG}}$	23.3	23.9
$\mathrm{R}_{\mathrm{AVG}}$	0.91	0.90
$\mathrm{MSE}_{\mathrm{QR}}$	1193.8	1254.8
$\mathrm{MAE}_{\mathrm{QR}}$	25.4	26.1
R_{QR}	0.88	0.87

The overall correlation between the target and estimated values of the test set was 0.87 for the QR model and 0.90 for the AVG model, but correlations or mean squared errors do not represent the order from the application's point of view, as the interest of the user is in rejection probability estimation. The average of the measurements is easier to predict than the absolute values, as the AVG model reduces the impact of one single low measurement.

The scatter plot between the predicted AVG values and the model error is illustrated in the uppermost plot in Fig.2. The scatter plot between the predicted 0.5^{th} quantile and the model error is illustrated in the lower plot. The model error is the difference between the average of the measurements and the predicted value of the models. For both QR

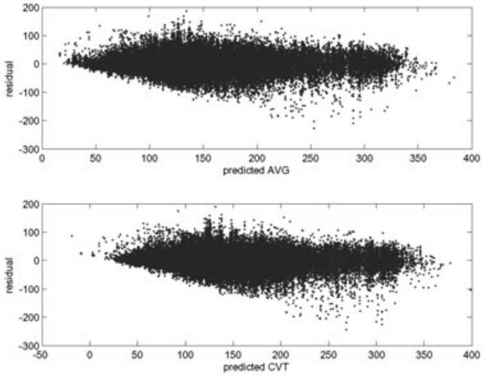


Fig. 2. Scatter plot between the predicted AVG and the model error and between the 0.5^{th} quantile and the model error for the independent test set

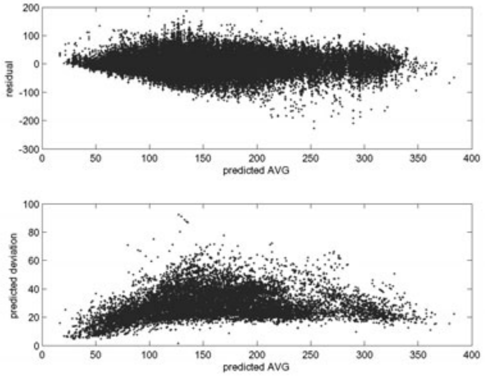


Fig. 3. Scatter plot between the predicted AVG and the model error and between the predicted AVG and the predicted deviation for the independent test set

and AVG models over 70% of the observations have a residual between $[-30, 30]$, and if the residuals between $[-50, 50]$ are observed, nearly 89% of the observations qualify into this group for both of the models. Observations with a residual of less than -150 in the lower right corner are plates that had a very low CVT value, but the model failed to recognize them. The number of them is 32 for the AVG model and 48 for the QR model. It can be seen that the average of the measurements decrease the impact of one low measurement, as the number of plates with poor residual is smaller. There are no similarities between these plates that could explain the poor result. There might be plates that were produced differently than planned (some of the critical process stages did not work as planned). The most probable explanation is the nature of the phenomenon itself. The existence of slag inclusions near the fracture causes low measurement values, and another test piece from the same plate could have much higher values.

The scatter plot between the predicted AVG values and the model error is illustrated in the uppermost plot in Fig.3. The scatter plot between the predicted AVG values and the predicted deviation is illustrated in the lower plot. A fitted model of the deviation shows that there are significant differences in the accuracy of the CVT-prediction, depending on the production method. The dependency indicates that the rejection probability estimation will benefit from the joint modelling.

Because the model was developed for all grades of steel in production, the results should be analyzed separately depending on the production method. Thus, the whole dataset was divided into six different groups. For AVG model four groups and for QR model five groups had observations with residual less than -150, but with both methods one of the groups had these observations nearly ten times more than other groups. This group has a high expected impact toughness, but the manufacturing process is very demanding. The three CVT measurements for most of these observations have a great variability, as well. The results indicate that the manufacturing process of this group is notstable and nothing specific cannot be pointed out causing the variability to the CVT results. The results of groups 2 and 5 are presented in detail, because they represent the largest groups in two very different production methods.

The motivation of this research was to develop methods for rejection probability estimation. Thus, the highest rejection probability values were observed at the typical rejection level 27 J. True rejections found with the highest probability values are shown

Table 2. Proportion of found true rejections in a test set when 2% or 10% of the highest rejection probabilities were selected

	2%	10%
QR	80%	93%
AVG_J	58%	80%
AVG_C	13%	48%

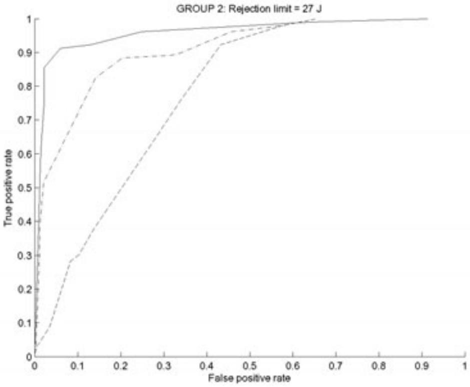


Fig. 4. ROC analysis in a test set for group 2 with a 27 J rejection limit (QR solid line, AVG_J dash-dot line, and AVG_C dashed line)

in Table 2 for each model. If 2% of the highest probabilities were selected, 80% of the rejections would have been found with QR model, 58% with AVG_J model and only 13% with AVG_C model. If 10% of the highest probabilities were selected, the QR model outperforms the others, as well.

Because the occurrence of the rejected plates is low, the accuracy of different models can be best examined with ROC (receiver operating characteristics) curves. The method will graphically display the trade-off between false-negative (1-true-positive) and false-positive rates obtained by varying the classification criteria [22]. The plates were rejected if the mean of three measurements was less than the rejection level or if the minimum value was lower than 30% of this level. The typical rejection level of 27 J for group 2 is illustrated in Fig.4 for the results of the test set. It can be seen that the rejected plates can be recognized very accurately without compromising with false rejections. The quantile regression model performs more accurately than the AVG models. The joint model for mean and variance performs significantly better than the model with constant variance. Group 2 makes up over 45% of the whole production, and hence the performance for this group is very important. The AUC (area under ROC) is a single-number measure for evaluating models, and these values are presented in Table 3. The values were calculated with four different rejection levels, and the results confirm the conclusions made from the ROC curves.

Table 3. AUC values in a test set for group 2 with different rejection levels

J	QR	AVG_J	AVG_C
20	0.9879	0.9824	0.8063
27	0.9601	0.9183	0.7831
40	0.9099	0.8402	0.7565
100	0.8955	0.8953	0.8935

The CVT requirements for products varied typically between 20 and 80 J, and although groups 5 and 6 had most commonly the rejection level 27 J, as well, it is more meaningful to analyze their results at a higher level. The ROC analysis for group 5 with a rejection level of 150 J can be seen in Fig.5. QR and AVG_J models perform slightly better than AVG_C model, but the overall performance is excellent. When the AUC values for different rejection levels were observed, none of the methods outperformed the others in every level. It was not possible to calculate the AUC value for this group at 27 J, because of the lack of low observations.

When the rejection level is raised, the performance of the joint model approaches the performance of the constant variance model. The amount of false rejections also starts to grow, but the rejections are still recognizable. The result can be seen in Fig.6 and in Table 4. When the rejection level is high, there is hardly any difference left between the QR and AVG models. Only two steel grades in group 2 require such high CVT values, though. When the overall performance of the models is compared, it can be seen that the QR model performs most accurately, and the variance model improves the accuracy of the AVG model in probability prediction.

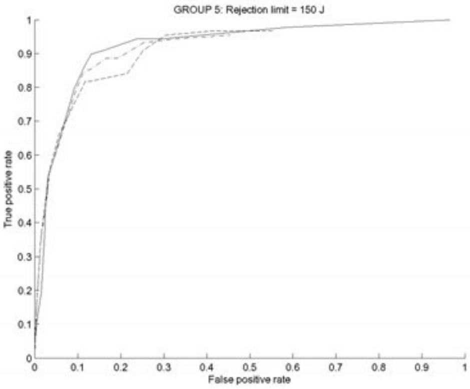


Fig. 5. ROC analysis in a test set for group 5 with a 150 J rejection limit (QR solid line, AVG_J dash-dot line, and AVG_C dashed line)

Table 4. AUC values in a test set for group 5 with different rejection levels

J	QR	AVG_J	AVG_C
27	-	-	-
40	0.9532	0.8958	0.9416
150	0.9241	0.9162	0.9128
200	0.9397	0.9463	0.9449

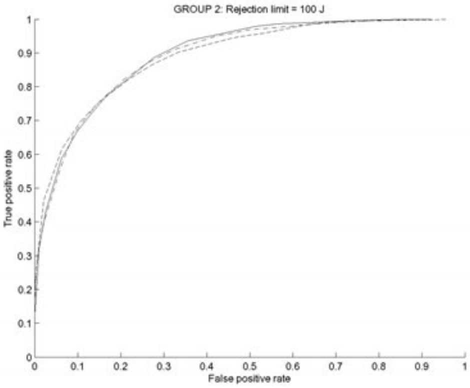


Fig. 6. ROC analysis in a test set for group 2 with a 100 J rejection limit (QR solid line, AVG_J dash-dot line, and AVG_C dashed line)

Table 5. AUC values in a test set for all the groups with 60 J rejection level

GROUP	QR	AVG_J	AVG_C
1	0.9739	0.9739	0.9247
2	0.8854	0.8649	0.8385
3	0.9255	0.9162	0.9207
4	0.8841	0.8499	0.8294
5	0.9145	0.8949	0.8828
6	0.9629	0.9548	0.9600

The performance of the models for all the product groups was observed at the 60 J rejection level. This level was selected, because it is the lowest possible rejection level that can be tested for all the products. The results can be seen in Table 5. There are differences in the performances in different groups, but QR model performs best in every product group.

8 Conclusions and Discussion

The data mining research group at the University of Oulu has studied the mechanical properties of steel plates for years, and models of tensile strength, yield strength and elongation have been developed earlier. [16] Because of the complicated nature of impact toughness, the first task in this study was to find out if the property could be modelled to a similar extent at all. A further motivation was to include the model in a simulation tool that the product design department uses to evaluate mechanical properties of a product and the probability of achieving requirements.

The study showed that it is possible to form a product design model for a whole product range, including all possible test temperatures. The QR model proved to be more efficient in CVT rejection prediction than the AVG model. For the AVG model, the variance model was used together with the mean model to calculate the rejection probability, and the joint model led to more reliable results than the model with constant variance. The research showed that the quantile regression method is very suitable for modelling product properties. The usability of the method is not restricted only to steel industry applications, but it is useful for other industrial areas, as well.

When the rejection level was raised different models began to perform more similarly, but in general, impact toughness rejection probability estimation clearly benefits from methods that take the heteroscedasticity into account. Because, majority of the products has a quite low rejection limit the quantile regression model is clearly the most accurate and the most beneficial method for product planning.

The number of models needing maintenance vary from a method to another. The number for QR model is three while for AVG_C model the number is one. The models need updating from time to time, because of the developments in the manufacturing process. The task is not too laborious, and the model accuracy need not to be compromised over the lesser amount of models.

The model will be implemented into a graphical simulation tool that is in daily use in the product planning department and already contains other mechanical property models. [23] The simulation tool is utilized to plan composition and production settings for product modifications and new products and to maintain the regulations for the production methods of existing products. According to the analysis of the results, it can be seen that most of the rejections could have been recognized with proposed model, and therefore it is expected that the number of rejections will be reduced when the model is entered into the product design.

Acknowledgment

The authors would like to thank Ruukki Metals, Raahe Works, Finland for providing the data and their expertise for the application. Further acknowledgments are given to the Finnish Funding Agency for Technology and Innovation (TEKES) and Infotech Oulu for supporting this research.

References

1. Tóth, L., Rossmanith, H.P., Siewert, T.: Historical background and development of the charpy test. In: François, D., Pineau, A. (eds.) From Charpy to Present Impact Testing. Elsevier Science Ltd., UK (2002)
2. Wallin, K., Nevasmaa, P., Planman, T., Valo, M.: Evaluation of the charpy-v test from a quality control test to a materials evaluation tool for structural integrity assessment. In: François, D., Pineau, A. (eds.) From Charpy to Present Impact Testing. Elsevier Science Ltd., UK (2002)
3. Lindroos, Sulonen, Veistinen: Uudistettu Miekk-ojan metallioppi. Otava, Finland (1986) (In Finnish)
4. Cawley, G., Janacek, G., Haylock, M., Dorling, S.: Predictive uncertainty in environmental modelling. Neural Networks 20, 537–549 (2007)
5. Haušild, P.: The influence of ductile tearing on fracture energy in the ductile-to-brittle transition temperature range. Materials Science and Engineering A335, 164–174 (2002)
6. Todinov, M.: Uncertainty and risk associated with the Charpy impact energy of multi-run welds. Nuclear Engineering and Design 231, 27–38 (2004)
7. Bhadeshia, H.: Neural networks in materials science. ISIJ International 39(10), 966–979 (1999)
8. Malinov, S., Sha, W., McKeown, J.: Modelling the correlation between processing parameters and properties in titaniumalloys using artificial neural network. ComputationalMaterials Science 21, 375–394 (2001)
9. Golodnikov, A., Macheret, Y., Trindade, A., Uryasev, S., Zrazhevsky, G.: Statistical modeling of composition and processing parameters for alloy development. Modelling and Simulation in Materials Science and Engineering 13(4), 633–644 (2005)
10. Koenker, R.: Quantile Regression. Cambridge University Press, USA (2005)
11. Yu, K., Lu, Z., Stander, J.: Quantile regression: Applications and current research areas. The Statistician 52(3), 331–350 (2003)
12. Engel, J.: Modelling variation in industrial experiments. Applied Statistics 41(3), 579–593 (1992)
13. Carroll, R., Ruppert, D.: Transformation and Weighting in Regression. Chapman and Hall, USA (1988)

14. Smyth, G., Huele, A., Verbyla, A.: Exact and approximate REML for heteroscedastic regression. Statistical Modelling 1(3), 161–175 (2001)
15. Juutilainen, I.: Modelling of Conditional Variance and Uncertainty Using Industrial Process Data. PhD thesis, University of Oulu, Finland (2006)
16. Juutilainen, I., Röning, J.: Modelling the probability of rejection in a qualification test based on process data. In: Proc. 16th Symposium of IASC (COMPSTAT 2004), Prague, Czech Republic, August 23-27, pp. 1271–1278 (2004)
17. Koenker, R., Hallock, K.: Quantile regression. Journal of Economic Perspectives 15(4), 143–156 (2001)
18. Bishop, C.: Neural Networks for Pattern Recognition. Oxford University Press, Inc., USA (1995)
19. Svozil, D., Kvasnička, V., Pospichal, J.: Introduction to multi-layer feed-forward neural networks. Chemometrics and Intelligent Laboratory Systems 39, 43–62 (1997)
20. Harvey, A.: Regression models with multiplicative heteroscedasticity. Econometrica 44(3), 461–465 (1976)
21. Saerens, M.: Building cost functions minimizing to some summary statistics. IEEE Transactions on Neural Networks 11(6), 1263–1271 (2000)
22. Dorling, S., Foxall, R., Mandic, D., Cawley, G.: Maximum likelihood cost functions for neural network models of air quality data. Atmospheric Environment 37, 3435–3443 (2003)
23. Laurinen, P., Tuovinen, L., Röning, J.: Smart archive: A component-based data mining application framework. In: Proc. 5th International Conference on Intelligent Systems Design and Applications, Wroclaw, Poland, September 8-10, pp. 20–26 (2005)

Mining for Paths in Flow Graphs

Adam Jocksch[1], José Nelson Amaral[2], and Marcel Mitran[3]

[1] Research in Motion, Waterloo, Canada
[2] Department of Computing Science
University of Alberta, Edmonton, Canada
[3] IBM Toronto Software Laboratory, Toronto, Canada

Abstract. This paper presents FlowGSP, a data-mining algorithm that discovers frequent sequences of attributes in subpaths of a flow graph. FlowGSP was evaluated using flow graphs derived from the execution of transactions in the IBM® WebSphere® Application Server, a large real-world enterprise application server. The vertices of this flow graph may represent single instructions, bytecodes, basic blocks, regions, or entire methods. These vertices are annotated with attributes that correspond to run-time characteristics of the execution of the program. FlowGSP successfully identified a number of existing characteristics of the WebSphere Application Server which had previously been discovered only through extensive manual examination. In addition, a multi-threaded implementation of FlowGSP demonstrates the algorithm's suitability for exploiting the resources of modern multi-core computers.

Keywords: Data mining, Flow Graphs, Compiler Implementation, Hardware Performance Monitors.

1 Introduction

Data from many domains can be represented as a flow graph with weights associated with the vertices and frequencies associated with the edges. A flow graph is a directed graph that may contain cycles. When the edge frequencies are normalized, they can be interpreted as the probability that flow will follow a given edge when leaving a vertex. This paper presents FlowGSP, a new data-mining algorithm for flow graphs whose vertices are annotated with binary attributes. The goal of FlowGSP is to discover sequences of attributes that occur often in subpaths of a large flow graph. A subpath is of interest if it occurs often or if its vertex weights are high. To the best of our knowledge, no other mining algorithm considers this dual nature of support for a subpath. Moreover, because of our motivating application, each vertex has *multiple attributes*, a characteristic that we believe has not yet been explored in graph mining. FlowGSP is an extension of Agrawal's and Srikant's Generalized Sequential Patterns (GSP) algorithm [20].

The motivation for the development of FlowGSP is the mining of data, collected by hardware profiling tools and compiler-introduced instrumentation, to discover interesting patterns in the execution of a computer program. Thus, the

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 277–291, 2010.

flow graphs that motivated the development of FlowGSP represent the control flow of a computer program, and therefore may contain cycles. In this domain, a subpath in the flow graph is of interest if it executes frequently or if it takes a relatively long time to execute. Moreover, there exists a partial order between vertices of the graph. In contrast, frequent-item-set mining algorithms, such as Apriori [1], are only able to mine independent events. Frequent-sequence mining algorithms, such as GSP [3], PrefixSpan [19], and WINEPI/MINEPI [14], take into account the context in which items in the database occur but rely on a total-ordering in the underlying data.

There has been significant work towards discovering patterns in topologically ordered data sets and specifically for data represented as a graph. Algorithms such as AGM [10], gSpan [23], Origami [7], and Gaston [17] all search for frequent subgraphs. Pawlak established connections between flow graphs and data mining but focused on flow graphs that model information flow in decision algorithms with the goal of discovering association rules. That study is, therefore, restricted to acyclic directed graphs [18].

The problem of finding interesting paths in a graph, or a collection of graphs, is a specialization of the more general problem of finding interesting subgraphs. Inokuchi *et al.* propose a method for transforming graph data into transactional form so that traditional basket analysis can be performed [11]. Yamamoto *et al.* present FMG, an algorithm that can discover frequent subgraphs in graphs with multiple attributes per vertex [22]. However, FMG does not mine graphs with weighted edges, weighted vertices, or weighted attributes. Inokuchi's method cannot detect patterns that occur over cycles in the graph. These shortcomings make these algorithms unsuitable for mining graphs generated by the execution of a computer program.

Hwang *et al.* present a method for mining frequent patterns in a directed graph constructed by tracing method calls in a Java program [8]. Each vertex in the unweighted, unlabeled, graph represents the entry into a new method in the Java$^{\text{TM}}$program. They do not include any attributes or performance information obtained from hardware profiling in their graph. They also collapse cycles in the graph into single vertices, whereas FlowGSP deals with cycles in the input graph.

Lee and Park, and Geng *et al.*, search for frequent subpaths in a database of paths between vertices in a graph [13,5]. A list of subpaths is an input to their algorithm. Their graphs contain only edge weights and do not have attributes associated with vertices. Their goal is to search for common subpaths. This is analogous to searching for the most-frequently-taken subpaths in a weighted graph. In contrast, FlowGSP searches for frequent sequences of attributes in a weighted and labeled graph.

The contributions of this paper are as follows:

– FlowGSP, a new mining algorithm that extends GSP [3] to search for frequent and/or costly paths in a vertex-weighted attributed flow graph.
– The presentation of a parallel implementation of FlowGSP as well as an evaluation of its performance.

- An use case, in the context of compiler development for large application servers, is presented as evidence of the practicality of FlowGSP.

Section 2 motivates FlowGSP. A formal description of the problem and FlowGSP is presented is sections 3 and 4, respectively. Sections 5 and 6 discuss the setup and results of experiments that confirm the potential of FlowGSP.

2 A Motivating Application

Many large enterprise applications have fairly "flat" execution profiles in which the execution time of the application is spread across a large set of procedures or methods. For instance, a transaction in a WebSphere Application Server [9] may execute millions of machine instructions. Typically, very few loops are observed and thousands of methods are invoked when processing a transaction. Given the wealth of data about each method executed, the task of discovering new opportunities for improving performance of the server is daunting [6].

For example, Nagpurkar *et al.* describe a methodology to reduce instruction cache latencies when running WebSphere Application Server [16]. While no single method accounts for more than 2% of execution time of the overall application, instruction-cache misses represent 12% of the program execution. The method with the most latency represents only 0.525% of the instruction-cache miss latency. To capture 75% of the overall instruction-cache latency, it is necessary to aggregate over roughly 750 methods. Thus, an strategy to address instruction cache latency requires global characterization of instruction-cache miss events.

This paper maps the problem of mining for repeated patterns of execution in a program to the mining of a flow graph. Units of execution are mapped to the vertices of the graph. Hardware or software measured events (*i.e.*, cache miss, instruction type, branch misprediction, *etc.*) are the attributes of these vertices. The edges represent the possibility that the execution will flow from one execution unit to another. These edges are annotated with frequency information. If the units of execution are assembly instructions or basic blocks, we have a control flow graph. If they are procedures, we have a call graph. Alternatives for the mapping into the edges of the graph may be bytecodes or even arbitrary single-entry single-exit regions.

FlowGSP is not the first attempt at using data mining to aid in the analysis of performance profiles. Moseley *et al.* developed Optiscope, an "optimization microscope" for examining hardware profiles [15]. Optiscope shares the same goal of presenting compiler developers with a more coherent view of the data contained in hardware profiles. However, Optiscope focuses on comparing two profiles of the same program run under different circumstances (compiler, optimization parameters, *etc.*) whereas FlowGSP focuses on the mining of data from a single execution of the program. Optiscope analysis is limited to basic database slicing operations.

FlowGSP was motivated by the problem of mining the flow graph generated by a large enterprise application. However, the algorithm can be applied to any mining application that can be mapped to an annotated flow graphs.

3 Flow Graphs with Weights, Frequencies, and Attributes

Let $G = \langle V, E, \alpha, \mathcal{A}, \mathcal{F}, \mathcal{W}, \mathcal{W}_{\alpha_i} \rangle$ be a flow graph such that:

- V is a set of vertices.
- E is a set of edges (v_a, v_b), where $v_a, v_b \in V$.
- α is the set of all possible attributes of vertices.
- $\mathcal{A}(v) \mapsto \{\alpha_1, ..., \alpha_k\}$ is a function mapping vertices $v \in V$ to a subset of attributes $\{\alpha_1, ..., \alpha_k\}, \alpha_i \in \alpha, 1 \le i \le k$.
- $\mathcal{F}(e) \mapsto [0, 1]$ is a function assigning a normalized frequency to each edge $e \in E$, *i.e.* $\sum\limits_{e \in E} \mathcal{F}(e) = 1$.
- $\mathcal{W}(v) \mapsto [0, 1]$ is a function assigning a normalized weight to each vertex $v \in V$, *i.e.* $\sum\limits_{v \in V} \mathcal{W}(v) = 1$.
- $\mathcal{W}_{\alpha_i}(v) \mapsto [0, 1]$ is a function assigning a normalized weight to an attribute α_i of a vertex v. It is required that $\alpha_i \in \mathcal{A}(v)$ and $\mathcal{W}_{\alpha_i}(v) \le \mathcal{W}(v)$.

A subpath $p \in G$ of length g is an ordered set of g vertices.[1] The notation $p[i]$ refers to the i^{th} vertex of p. For p to be a subpath there must be edges $(p[i], p[i + 1]) \in E$ for all $0 \le i \le g - 2$.

The edge frequencies and the vertex weights are assumed to be independent measures. Therefore, the algorithm uses two separate measures of support for a subpath: a frequency support and a weight support. The frequency support for vertex v is the sum of the frequencies of the incoming edges into v:

$$\mathcal{S}_{\mathcal{F}}(v) = \sum_{(v_a, v) \in E} \mathcal{F}(v_a, v)$$

The frequency support for a subpath p is the minimum of the frequency support of the edges that form p:

$$\mathcal{S}_{\mathcal{F}}(p) = \min\{\mathcal{F}(p[0], p[1]), \ldots \mathcal{F}(p[g - 2], p[g - 1])\}$$

The weight support for a subpath p is the minimum of the weight of the vertices that form p. However, not all attributes of a given vertex may contribute to a sequence. The subset of attributes of a vertex that contribute to a sequence is denoted as $\mathcal{A}_S(v)$. Thus, the support of a subpath with respect to a sequence of attributes S is:

$$\mathcal{S}_{\mathcal{W},S}(p) = \min_{0 \le j \le g-1} \begin{cases} \min\limits_{\alpha_i \in \mathcal{A}_S(p[j])} \mathcal{W}_{\alpha_i}(v) & \text{if} \mathcal{A}_S(p[j]) \ne \emptyset \\ \mathcal{W}(p[j]) & \text{if} \mathcal{A}_S(p[j]) = \emptyset \end{cases}$$

[1] A path through a graph usually refers to a path from source to sink. The mining algorithms discover *subpaths* connecting two arbitrary vertices.

3.1 Path versus Edge Profiling

FlowGSP mines a flowgraph that has only edge-profiling frequency information to try to discover frequent subpaths.[2] Precise execution paths cannot be derived from edge profiling [4]. Therefore the subpaths mined by FlowGSP are an approximation and may overestimate the actual support for subpath execution. This imprecision is mitigated because FlowGSP aggregates information from many occurrences of a subpath in the graph.

3.2 Attributes and Sequences

Each attribute of a vertex is assigned a weight representing the significance of the attribute. The rationale behind assigning a weight to attributes as well as to vertices stems from the motivating application: an attribute may be true in some but not all hardware sampling of the same instruction. The vertex weight represents the execution time spent on the instruction and an attribute weight represents the frequency of occurrence of the attribute. Thus, the weight of an attribute must be smaller than or equal to the weight of its vertex. Attributes with a weight of zero are omitted. Binary attributes are represented by assigning "true" attributes the same weight as the vertex on which they are located; "false" attributes are omitted. Enumerated attributes that can take one of r possible values are represented by r mutually exclusive binary attributes, where r is a known positive integer. Binary and enumerated attributes must have the same weight as the vertex on which they occur.

The mining algorithm will discover subpaths in a flow graph such that the vertices that form the subpath contain a sequence of sets of attributes. For instance, consider a flow graph G with $\alpha = \{a, b, c, d, e\}$ that contains a subpath formed by three vertices $p = \{v_1, v_2, v_3\}$. Assume that the following subset of attributes are associated with each of the vertices in this subpath: $\alpha_1 = \{a, e\}$, $\alpha_2 = \{b, c, d\}$, $\alpha_3 = \{a, d, e\}$. Then the subpath p contains the sequence of sets of attributes $S = \langle(a, e), (b, c, d), (a, d, e)\rangle$. The subpath p also contains any subsequence of sets of attributes derived from S by eliminating attributes from the subsets that form S. For instance, p contains $S' = \langle(a), (b, d), (a, e)\rangle$.

The sequence of sets of attributes contained by a subpath may skip vertices in the subpath. For instance, in the example above, the subpath p also contains the sequence $S'' = \langle(a, e), (a, d)\rangle$ even though the formation of this sequence skips the attributes of vertex v_2. The support for S'' is the support of p. Thus, for instance, if v_2 has the lowest weight in p, it still determines the weight support for S'' even though none of its attributes contributes to S''. A subpath is *minimal* with respect to a candidate sequence S if $p[0]$ and $p[g-1]$ contain part of the candidate sequence, *i.e.*, the first and last vertices in the subpath are not skipped. Henceforth, all subpaths are minimal.

[2] An alternative profiling technique would be to compute path profiling information during the execution of the program. Path profiling is much more expensive to collect.

Example

Figure 1 shows an example an EFG. In the original graph (left), vertices are annotated with frequencies, the annotation to the left of each vertex is the weight, the annotation to the right is the set of attributes in the node along with the attribute weight. In the graph to the right all the weights and frequencies are normalized. The graph on the right highlights two paths, $p_1 = \{v_1, v_3, v_7\}, p_2 = \{v_2, v_4, v_6\}$, that contain instances of the sequence $S_1 = \langle (a), (b), (e) \rangle$.

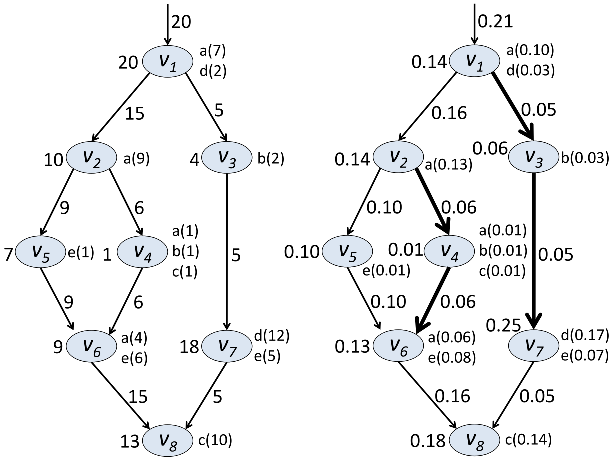


Fig. 1. An example of an EFG with weighted attributes: unnormalized (left) and normalized (right)

4 Mining Flow Graphs with Attributes

The goal of mining an EFG is to discover sequences of attributes that: (1) either happen frequently or occur in subpaths with high weight; and (2) happen frequently in subpaths with low weight or occur in subpaths with high weight that do not happen frequently.

The algorithm searches for finite sequences of sets of attributes $\alpha_0, \alpha_1, \alpha_2, \ldots$ that are associated with subpaths in the flow graph. Given a sequence S, the support for S is determined by the support for the subpaths that contain S. Let P_S be the set of subpaths in G that contain S. The frequency and weight support for S in G is then defined as:

$$\mathcal{S}_{\mathcal{F}}(S) = \sum_{p \in P_S} \mathcal{S}_{\mathcal{F}}(p) \qquad \mathcal{S}_{\mathcal{W}}(S) = \sum_{p \in P_S} \mathcal{S}_{\mathcal{W},S}(p)$$

To discover sequences that either happen frequently or result in a significant cost, the S_{max} support is defined as:

$$\mathcal{S}_{max}(S) = \max\{\mathcal{S}_{\mathcal{F}}(S), \mathcal{S}_{\mathcal{W}}(S)\} \tag{1}$$

The support to discover sequences for which the time and cost support are significantly different is:

$$S_{\Delta}(S) = |S_f(S) - S_w(S)| \tag{2}$$

Moreover, both the frequency and weight supports must be above set thresholds.

The supports for sequence $S_1 = \langle (a), (b), (e) \rangle$ in the example of Figure 1 is calculated as follows:

$$\mathcal{S}_{\mathcal{F}} = \mathcal{S}_{\mathcal{F}}(p_1) + \mathcal{S}_{\mathcal{F}}(p_2) = \min\{0.21, 0.05, 0.05\} + \min\{0.16, 0.06, 0.06\} = 0.11$$
$$\mathcal{S}_{\mathcal{W}} = \mathcal{S}_{\mathcal{W}}(p_1) + \mathcal{S}_{\mathcal{W}}(p_2) = \min\{0.10, 0.03, 0.17\} + \min\{0.13, 0.01, 0.08\} = 0.04$$
$$\mathcal{S}_{max}(S_1) = 0.11 \qquad \mathcal{S}_{\Delta}(S2) = 0.07$$

4.1 Algorithm Description

FlowGSP is an extension of GSP, and hence shares many of the same characteristics [20]. FlowGSP is an iterative, generate-and-test algorithm. At each iteration, a list of candidate sequences are generated, their frequency and cost support are calculated, and unfit sequences are discarded.

FlowGSP is analagous to enumerating every possible path through the EFG being mined, and then running a slightly modified GSP on the resulting database of paths. However, rather than explicitly enumerating every path which may or may not contain any candidate sequences, FlowGSP enumerates each path on the fly, and only as required. Generating paths dynamically also allows FlowGSP to mine for patterns in graphs for which it is not possible to enumerate all possible paths, for instance in an EFG that contains cycles.

The candidate sequences for the n^{th} generation are created by combining the surviving members of the $(n-1)^{st}$ generation according to the "apriori" principle [2].[3] A candidate survives if its support is above a set threshold. The algorithm is outlined by Algorithm 1.

GSP was chosen as the base algorithm for FlowGSP because GSP is simple, well-studied and understood, and descriptions of efficient implementation are available [20]. Moreover, with the goal of discovering all sequences of sets of attributes that meet a defined support criteria, it was more natural to extend a sequential pattern-mining algorithm to traverse a graph than to modify a graph-mining algorithm to find all such sequences.

4.2 Support Calculation

FlowGSP greedily looks for a candidate sequence starting at the current node, and conducts a breadth-first traversal (BFT) of the graph. FlowGSP uses a hash tree to search only a subset of the candidate sequences for each starting node [20]. Once a potential start point is found, the rest of the instance is greedily searched

[3] The "apriori" principle is as named by the original authors and has no connection to *a-priori* or *a-posteriori* inference.

Algorithm 1. FlowGSP

```
FlowGSP($G$, $g_{max}$, $w_{max}$, $n_{gen}$, $s_{Mthresh}, s_{\Delta thresh}$)
$G_1 \leftarrow Create\_First\_Generation(\alpha)$
$n \leftarrow 1$
while $G_n \neq \emptyset$ and $n < n_{gen}$ do
    for $v \in G$ do
        $C \leftarrow get\_candidates(v)$
        for $S \in C$ do
            $supports \leftarrow Find\_Paths(S, v, 0, g_{max}, w_{max})$
            for $(\mathcal{S}_{\mathcal{W}}, \mathcal{S}_{\mathcal{F}}) \in supports$ do
                $\mathcal{S}_{\mathcal{W}}(S) \leftarrow \mathcal{S}_{\mathcal{W}}(S) + \mathcal{S}_{\mathcal{W}}$
                $S_f(S) \leftarrow S_f(S) + \min\{S_f, S_f(v)\}$
            end for
        end for
    end for
    for $S \in G_n$ do
        if $\mathcal{S}_{max}(S) < s_{Mthresh}$ and $\mathcal{S}_{\Delta}(S) < s_{\Delta thresh}$ then
            Remove $S$ from $G_n$
        end if
    end for
    if $n < n_{gen} - 1$ then
        $G_{n+1} \leftarrow Make\_Next\_Gen(G_n)$
    end if
    $n \leftarrow n + 1$
end while
```

Algorithm 2. Algorithm to find all paths that contain a sequence S starting at a vertex v

```
$Find\_Paths(S, v, g_{\text{remain}}, g_{\text{max}}, w_{\text{max}})$
supports $\leftarrow Find\_Set(S[0], \emptyset, S, v, w_{max}, g_{\text{max}}, w_{\text{max}})$
if supports $\neq \emptyset$ or $g_{\text{remain}} \leq 0$ then
    return $\emptyset$
end if
for $v' \in children(v)$ do
    supports$'$ $\leftarrow Find\_Paths(S, v', g_{\text{remain}} - 1, g_{\text{max}}, w_{\text{max}})$
    for $(\mathcal{S}_{\mathcal{W}}, \mathcal{S}_{\mathcal{F}}) \in$ supports$'$ do
        $\mathcal{S}_{\mathcal{W}} \leftarrow \min\{\mathcal{S}_{\mathcal{W}}, \mathcal{W}(v)\}$
        $\mathcal{S}_{\mathcal{F}} \leftarrow \min\{\mathcal{S}_{\mathcal{F}}, \mathcal{F}((v, v'))\}$
        supports $\leftarrow$ supports $\cup \{(\mathcal{S}_{\mathcal{W}}, \mathcal{S}_{\mathcal{F}})\}$
    end for
    return supports
end for
```

Algorithm 3. Algorithm to find the next set of attributes in the sequence

```
$Find\_Set(s_{\text{left}}, s_{\text{found}}, S, v, w_{\text{remain}}, g_{\max}, w_{\max})$
supports $\leftarrow \emptyset$
if $s_{\text{left}} \subseteq \mathcal{A}(v)$ then
    if $|S| = 1$ then
        supports $= \{(W(v), \infty)\}$
        return supports
    end if
    for $v' \in children(v)$ do
        supports$' \leftarrow Find\_Paths(S[1, k-1], v', g_{\max}, g_{\max}, w_{\max})$
        for $(\mathcal{S}_{\mathcal{W}}, \mathcal{S}_{\mathcal{F}}) \in$ supports$'$ do
            $\mathcal{S}_{\mathcal{W}} \leftarrow \min\{\mathcal{S}_{\mathcal{W}}, \mathcal{W}(v)\}$
            $\mathcal{S}_{\mathcal{F}} \leftarrow \min\{\mathcal{S}_{\mathcal{F}}, \mathcal{F}((v, v'))\}$
            supports $=$ supports $\cup \{(\mathcal{S}_{\mathcal{W}}, \mathcal{S}_{\mathcal{F}})\}$
        end for
    end for
    return $supports$
else
    if $w_{\text{remain}} \leq 0$ then
        return $\emptyset$
    end if
    $s_{\text{found}} \leftarrow s_{\text{found}} \cup (\mathcal{A}(v) \cap s_{\text{left}})$
    $s_{\text{left}} \leftarrow s_{\text{left}} \setminus \mathcal{A}(v)$
    for $v' \in children(v)$ do
        supports‘ $\leftarrow Find\_Set(s_{\text{left}}, s_{\text{found}}, S, v', w_{\text{remain}} - 1, g_{\max}, w_{\max})$
        for $(\mathcal{S}_{\mathcal{W}}, \mathcal{S}_{\mathcal{F}}) \in$ supports$'$ do
            $\mathcal{S}_{\mathcal{W}} \leftarrow \min\{\mathcal{S}_{\mathcal{W}}, \mathcal{A}_S(v)\}$
            $\mathcal{S}_{\mathcal{F}} \leftarrow \min\{\mathcal{S}_{\mathcal{F}}, \mathcal{F}((v, v'))\}$
            supports $\leftarrow$ supports $\cup \{(\mathcal{S}_{\mathcal{W}}, \mathcal{S}_{\mathcal{F}})\}$
        end for
    end for
    return supports
end if
```

via depth-first search. When an instance of a sequence is found, its supports are added to the candidate's total supports.

The functions *Find_Paths* and *Find_Set*, outlined in Algorithms 2 and 3 respectively, perform the depth-first search described above. These functions are mutually-recursive; *Find_Set* locates the next set of attributes in the target sequence then calls *Find_Paths* to locate the rest of the sequence. *Find_Paths* then calls *Find_Set* once it has identified the potential start of the next set of attributes. Both methods return a set of $(\mathcal{S}_{\mathcal{F}}, \mathcal{S}_{\mathcal{W}})$ tuples, each tuple representig the supports of a subpath which contains the sequence S. The search terminates when either S has been located or the maximum gap or window size has been exceeded.

Two additional checks are required in order to ensure that all instances of the current sequence are counted correctly. First, the value of w_{remain} is ignored in *Find_Set* for the start of the first set of attributes in a sequence if the current vertex does not contribute at all to the sequence. If v is not the true starting vertex of the sequence then it does not make sense to begin calculating the support of the sequence starting at v.

Find_Set also aborts when searching for the first set of attributes in a sequence if it encounters a vertex v where $\mathcal{A}(v)$ contains all of the attributes from the set found thus far. The instance of the sequence that starts at v is the more precise of the two instances, and therefore it is that instance which should count toward the support of the sequence.

4.3 Candidate Generation

FlowGSP is an iterative generate-and-test algorithm. The candidate generation process is unchanged from Srikant *et al.* [20] however a short description is included here. At the end of each iteration, the next generation of candidates is generated from the surviving members of the current generation. A suffix and a prefix are created for each candidate sequence by removing the first and last attribute of the sequence respectively. A new sequence can be created by adding the attribute removed from the end of one sequence onto the end of another, but only if the prefix of the first sequence is equal to the suffix of the second.

4.4 Extensions

FlowGSP supports variable sliding-window sizes as well as variable gap sizes between elements in a sequence. A gap size of g_{max} means that up to g_{max} vertices can occur between vertices that match consecutive sets of attributes α_i, α_{i+1} in a sequence. A window size of w_{max} means that a set of attributes only has to match the union of the attributes of up to w_{max} consecutive vertices on the path. In both these cases, the smallest possible gap or window size is used when searching for instances of a sequence.

4.5 Dealing with Cycles

The subpaths found by FlowGSP may encompass multiple iterations over a cycle. To prevent looping indefinitely, FlowGSP maintains a list L of vertices that are the start of a subpath containing a candidate sequence. If FlowGSP encounters a vertex v that is in L, neither v nor its descendants can be the start vertex for a new subpath, and thus cannot be added to L. This restriction enforces that only vertices encountered during the first visit to the cycle start subpaths.

In each generation, FlowGSP searches for instances of sequences of a specified, finite, length. The gap and window sizes are also finite. Once a vertex that represents the potential start of an instance is located, the maximum number of vertices visited to uncover a sequence is $seqlength * w_{max} + (seqlength - 1) * g_{max}$.

Thus, given that (i) any subpath of interest is finite, and (ii) the start node of the subpath must be encountered during the first traversal of a cycle, FlowGSP is guaranteed to terminate even in the presence of cycles.

5 Experimental Setup

This evaluation of FlowGSP uses a 450 MB profile from a five-minute run of the WebSphere Application Server on the IBM z10$^{\mathrm{TM}}$architecture [21]. The compiler log produced 6 GB of data. The database contained 102,865 individual assembly instructions, 30,430 basic blocks, and 44,178 inter-basic block edges. The maximum number of attributes observed on a vertex was 70, with an average of 1.4 attributes per vertex. FlowGSP was implemented in Java and run on a dual quad-core AMD 2350 CPUs with 8 GB of RAM. The database containing the profiling data was hosted on an AMD dual-core CPU with 2 GB of RAM running IBM DB2® Database server.

Hardware profiles are collected through regular sampling of the machine state. Machine state includes which instruction is being executed and hardware events (instruction or data cache miss, Branch misprediction, transition lookaside buffer (TLB) miss, *etc.*). The total number of samples that *hit* an instruction correlates with the time spent on the instruction. This data is used to annotate the program CFG generated by the IBM Testarossa JIT compiler.

A multi-threaded implementation of FlowGSP takes advantage of the data-independence in the flow graph, which is structured at the level of a method. A single master thread coordinates the division of methods to be mined among work threads. When all workers have finished, the master prunes candidates with inadequate support and constructs the next generation of candidates. The master and workers communicate via shared memory. The data set for each generation is too large to fit in main memory and has to be re-fetched from the database. Because of the large number of methods in the WebSphere Application Server, the problem is naturally over-decomposed resulting in good load balancing.

6 Results

The experimental results presented in this section demonstrate that FlowGSP works in the context of mining execution paths in WebSphere Application Server, a large enterprise application.

6.1 Interesting Sequence Discovery

Before the development of FlowGSP, compiler developers were faced with the difficult challenge of identifying patterns in the execution paths of large enterprise applications using nothing but intuition and observation. This intuitive approach may lead to large investments in compiler development effort that may not necessarily pay off.

With the implementation of FlowGSP, an interesting *acid test* of the automatic approach is to discover patterns that had already been identified manually by compiler developers. FlowGSP passed this acid test because it was able to identify all the patterns that were known to the developers. Some of these patterns include:

- $\langle(Icachemiss, TLBmiss)\rangle, \mathcal{S}_{max} = 0.529$ indicates a high correlation between instruction cache misses and TLB misses on the host architecture.
- $\langle(Prologue, Icachemiss)\rangle, \mathcal{S}_{max} = 0.1175$ indicates a high occurrence of instruction cache Misses in the prologues of methods.
- $\langle(JITtarget, Icachemiss)\rangle, \mathcal{S}_{max} = 0.0935$ corresponds to a significant number of instruction cache Misses on the first instruction executed when a method is called from natively compiled code.[4] The level of support for this attribute pair is even more significant because the sequence $\langle(JITtarget)\rangle$ has $\mathcal{S}_{max} = 0.0935$.

Subsequently several unknown patterns were discovered and led to important new investigations of performance improvement opportunities in the compiler [12].

Table 1. Running times for FlowGSP by generation, in seconds. Each value reported is the mean of 10 runs with a 95% confidence interval according to the Student's t-distribution. All values rounded to the nearest second.

# of Threads	Execution time (s)					
	Gen. 1	Gen. 2	Gen. 3	Gen. 4	Gen. 5	Total
1 (baseline)	209 ± 2	411 ± 1	877 ± 3	3360 ± 27	16076 ± 99	20939 ± 112
2	168 ± 16	316 ± 1	567 ± 2	1837 ± 10	8672 ± 86	11581 ± 84
3	151 ± 1	297 ± 10	492 ± 1	1378 ± 7	6280 ± 64	8603 ± 65
4	148 ± 0	291 ± 1	462 ± 1	1166 ± 19	5384 ± 176	7455 ± 195
5	149 ± 0	292 ± 1	452 ± 1	1030 ± 4	4623 ± 34	6549 ± 37
6	150 ± 1	294 ± 3	456 ± 18	953 ± 21	4003 ± 28	5858 ± 58
7	150 ± 1	297 ± 1	452 ± 1	904 ± 5	3849 ± 33	5655 ± 35
8	148 ± 0	291 ± 0	445 ± 1	860 ± 4	3496 ± 47	5244 ± 49

6.2 Algorithm Performance

FlowGSP was run with different number of threads. Table 1 shows time to complete each generation and the total execution time for 1 to 8 threads. The total execution time reduces from 6.3 hours for the 1-thread instance to 1.9 hours for the 8-thread instance, representing a 70% improvement in execution time. This result is significant because it implies that within a day a production compiler environment could do multiple minings of a large application such as Websphere.

[4] In general, a different first instruction is executed when the method is called from interpreted code.

Table 2. Amount of time spent fetching data from the database for the first generation and breakdown of running time between sequential and parallel portions of FlowGSP

Number of Threads	Time Spent Fetching (%)	Execution Time (s)	
		Mining	Pruning/Joining
1	77.6	20851	82
2	86.2	11474	85
3	89.8	8512	86
4	91.8	7265	87
5	92.0	6459	87
6	94.0	5767	86
7	92.2	5562	89
8	95.7	5149	92

Table 2 shows that, on average, 90% of the time spent in the processing of the first generation is spent fetching data from the database. Thus it is no surprise that adding active threads to the computation does not significantly improve the performance of the early generations. Table 2 also shows that the amount of time spent on the sequential portion of the algorithm (pruning and joining) is independent of the number of threads. Therefore, it appears that the database server is currently the dominant bottleneck in our experimental setup.

7 Conclusion

FlowGSP is a novel algorithm that addresses the problem of mining for frequent sequences in subpaths of a flow graph that has weights and attributes associated with its vertices and frequencies associated with its edges. To the best of our knowledge no other existing algorithm is able to mine such flow graphs.

FlowGSP is evaluated using data collected from the just-in-time compilation and execution of IBM WebSphere Application Server, a state-of-the-art Java Enterprise Edition (JEE) application server broadly deployed in industry. A large set of hardware performance counters and compiler attributes were captured and associated to the assembly-code representation of WebSphere Application Server. FlowGSP was also able to quickly identify patterns in WebSphere Application Server profiles which had been previously identified through manual examination. These patterns demonstrate the potential of FlowGSP to fascilitate the discovery of new opportunities for code transformations.

Copyright and Trademarks

IBM, Websphere, z10, and DB2 are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both. The symbols (® or ™) on their first occurence indicates U.S. registered or common law trademarks owned by IBM at the time of publication. Such trademarks may also be registered or common law trademarks in other countries. Other company, product, and service names may be trademarks or service marks of others.

Acknowledgment

Thanks to support from the IBM Centre for Advanced Studies (CAS) and from the Natural Sciences and Engineering Research Council of Canada Collaborative Research and Development program. The IBM Testarossa JIT team provided expertise and extensive support for this research. Special thanks to Joran Siu and Nikola Grcevski for valuable assistance with data collection and analysis of results.

References

1. Agrawal, R., Imieliński, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD International Conference on Management of Data, Washington, DC, USA, pp. 207–216 (1993)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: International Conference on Very Large Data Bases (VLDB), Santiago, Chile, September 1994, pp. 487–499 (1994)
3. Agrawal, R., Srikant, R.: Mining sequential patterns. In: International Conference on Data Engineering (ICDE), Taipei, Taiwan, March 1995, pp. 3–14 (1995)
4. Ball, T., Mataga, P., Sagiv, M.: Edge profiling versus path profiling: the showdown. In: Principles of Programming Languages (POPL), San Diego, California, United States, pp. 134–148 (1998)
5. Geng, R., Dong, X., Zhang, X., Xu, W.: Efficiently mining closed frequent patterns with weight constraint from directed graph traversals using weighted FP-tree approach. In: Intern. Coll. on Computing, Communication, Control, and Management, Guangzhou City, China, August 2008, pp. 399–403 (2008)
6. Grcevski, N., Kielstra, A., Stoodley, K., Stoodley, M., Sundaresan, V.: Java just-in-time compiler and virtual machine improvements for server and middleware applications. In: Conf. on Virtual Machine Research and Technology Symposium (VM), San Jose, CA, USA, p. 12. USENIX Assoc. (2004)
7. Hasan, M.A., Chaoji, V., Salem, S., Besson, J., Zaki, M.: Origami: Mining representative orthogonal graph patterns. In: International Conference on Data Mining (ICDM), Washington, DC, USA, pp. 153–162 (2007)
8. Hwang, C.-C., Huang, S.-K., Chen, D.-J., Chen, D.T.K.: Object-oriented program behavior analysis based on control patterns. In: Asia-Pacific Conf. on Quality Software, Hong Kong, China, December 2001, pp. 81–87 (2001)
9. IBM Corporation. WebSphere Application Server (March 2009), `http://www-01.ibm.com/software/websphere/`
10. Inokuchi, A., Washio, T., Motoda, H.: An apriori-based algorithm for mining frequent substructures from graph data. In: Zighed, D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 13–23. Springer, Heidelberg (2000)
11. Inokuchi, A., Washio, T., Motoda, H., Kumasawa, K., Arai, N.: Basket analysis for graph structured data. In: Zhong, N., Zhou, L. (eds.) PAKDD 1999. LNCS (LNAI), vol. 1574, pp. 420–431. Springer, Heidelberg (1999)
12. Jocksch, A., Mitran, M., Siu, J., Grcevski, N., Amaral, J.N.: Mining opportuinities for code improvement in a just-in-time compiler. In: Compiler Construction (CC), Paphos, Cyprus (March 2010)

13. Lee, S.D., Park, H.C.: Mining frequent patterns from weighted traversals on graph using confidence interval and pattern priority. Intern. Journal of Computer Science and Network Security 6(5A), 136–141 (2006)
14. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovering Frequent Episodes in Sequences. In: Fayyad, U.M., Uthurusamy, R. (eds.) Knowledge Discovery and Data Mining (KDD), Montreal, Canada (1995)
15. Moseley, T., Grunwald, D., Peri, R.V.: Optiscope: Performance accountability for optimizing compilers. In: Code Generation and Optimization (CGO), Seattle, WA, USA (2009)
16. Nagpurkar, P., Cain, H.W., Serrano, M., Choi, J.-D., Krintz, R.: A study of instruction cache performance and the potential for instruction prefetching in J2EE server applications. In: Workshop of Computer Architecture Evaluation using Commercial Workloads, Phoenix, AZ, USA (2007)
17. Nijssen, S., Kok, J.N.: A quickstart in frequent structure mining can make a difference. In: Knowledge Discovery and Data Mining (KDD), Seattle, WA, USA, pp. 647–652 (2004)
18. Pawlak, Z.: Flow graphs and data mining. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 1–36. Springer, Heidelberg (2005)
19. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: PrefixSpan mining sequential patterns efficiently by prefix projected pattern growth. In: International Conference on Data Engineering (ICDE), Heidelberg, Germany, pp. 215–226 (2001)
20. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Advances in Database Techn., pp. 3–17. Springer, Heidelberg (1996)
21. Webb, C.F.: IBM z10: The next generation microprocessor. IEEE Micro 28(2), 19–29 (2008)
22. Yamamoto, T., Ozaki, T., Ohkawa, T.: Discovery of Frequent Graph Patterns that Consist of the Vertices with the Complex Structures. LNCS, pp. 143–156. Springer, Heidelberg (2008)
23. Yan, X., Han, J.: gSpan: Graph-based substructure pattern mining. In: International Conference on Data Mining (ICDM), Washington, DC, USA, p. 721 (2002)

Combining Unsupervised and Supervised Data Mining Techniques for Conducting Customer Portfolio Analysis

Zhiyuan Yao, Annika H. Holmbom, Tomas Eklund, and Barbro Back

Åbo Akademi University and Turku Centre for Computer Science (TUCS),
Joukahainengatan 3-5A, 20520 Turku, Finland
{Zhiyuan.Yao,Annika.H.Holmbom,Tomas.Eklund,
Barbro.Back}@abo.fi

Abstract. Leveraging the power of increasing amounts of data to analyze customer base for attracting and retaining the most valuable customers is a major problem facing companies in this information age. Data mining technologies extract hidden information and knowledge from large data stored in databases or data warehouses, thereby supporting the corporate decision making process. In this study, we apply a two-level approach that combines SOM-Ward clustering and decision trees to conduct customer portfolio analysis for a case company. The created two-level model was then used to identify potential high-value customers from the customer base. It was found that this hybrid approach could provide more detailed and accurate information about the customer base for tailoring actionable marketing strategies.

Keywords: Customer relationship management (CRM), customer portfolio analysis (CPA), Self-organizing maps (SOM), Ward's clustering, decision trees.

1 Introduction

For a long time, the focus of modern companies has been shifting from being product-oriented to customer-centric organizations. In the industry it is commonly held that maintaining existing customers is more cost-effective than attracting new ones, and that 20% of customers create 80% of the profit [1,2]. Reichheld and Teal [3] also point out that a 5% increase in customer retention leads to a 25–95% increase in company profit. Therefore, companies are focusing attention on building relationships with their customers in order to improve satisfaction and retention. This implies that companies must learn much about their customers' needs and demands, their tastes and buying propensities, etc., which is the focus of *Customer Relationship Management (CRM)* [4]. For CRM purposes, data mining techniques can potentially be used to extract hidden information from customer databases or data warehouses.

Data mining techniques can potentially help companies efficiently conduct *Customer Portfolio Analysis (CPA)*, which is the process of analyzing the existing and potential value of customers, thereby allocating limited resources to various customer groups according to the corporate strategy [4,5]. In this study, we propose a hybrid approach that combines the Self-Organizing Map (SOM)-Ward clustering [6,7] and decision trees for conducting CPA, aiming to create a more informative model for

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 292–307, 2010.

focused marketing efforts, compared to using either method alone. First, we use SOM-Ward clustering to conduct customer segmentation, so that the customer base is divided into distinct groups of customers with similar characteristics and behavior. This will allow us to identify the characteristics that separate high-spending customers from low-spending customers. Then, a decision tree technique will be used to further explore the relationship between customers' spending amounts and their demographic and behavioral characteristics. Finally, the trained decision tree model will be used to identify the segments with development potential, as well as customers in the group displaying mid-range spending that have similar characteristics as the high-spending customers. This group thus represents potential high-value customers if correctly activated. By extension, this type of analysis would allow companies to adjust their marketing efforts in order to better fit their customers' needs and demands, not only helping to enhance their relationship with important customers but also cutting down on advertising costs and improving the profitability of the entire customer base.

Although the SOM-based approach and Decision Trees have been used for market segmentation, classification, and data exploration problems individually, these two approaches have not to our knowledge previously been combined to perform CPA. A dataset of more than one million customers was used to create the models.

The remainder of this paper is organized as follows. Section two introduces the methodology (SOM-Ward and Decision Trees) and the data used in this study. Sections three and four document the training and analysis of the SOM-Ward and Decision Tree models respectively. In Section five, the trained Decision Tree model is used to analyze unclassified customers in order to identify their market potential. Section six presents our conclusions.

2 Methodology

2.1 The SOM and SOM-Ward Clustering

The SOM is a well-known and widely used unsupervised neural network that is able to explore relationships in multidimensional input data and project them onto a two-dimensional map, where similar inputs are self-organized and located together [8]. Additionally, as an unsupervised artificial neural network (ANN), the SOM is a data-driven clustering method. In other words, it works with very little *a priori* information or assumptions concerning the input data. Moreover, the SOM is able to compress the input data while preserving the topological relationships of the underlying data structure [8]. For these reasons, the SOM is considered an important tool for conducting segmentation tasks. The algorithm is well-known and will, therefore, not be further presented in this paper. Readers are referred to Kohonen [8] for details concerning the algorithm.

The SOM has been widely applied as an analytical tool in different business-related areas [9-11], including market segmentation [12-14]. In the above mentioned studies, the SOM is used alone, compared with, or used in conjunction with other clustering techniques for conducting market segmentation tasks. Vesanto and Alhoniemi [15] proposed a two-level approach, e.g., SOM-Ward clustering, for conducting clustering tasks. First, the dataset is projected onto a two-dimensional display using the SOM. Then, the resulting SOM is divided into groups. Lee et al. [14], adopting the two-level SOM (using SOM and K-means clustering), conducted a market segmentation of the Asian online game market and found that the two-level SOM is more accurate in classification than K-means clustering or the SOM alone. Samarasinghe [16], comparing two clustering methods (SOM-Ward and K-means clustering) drew the conclusion that SOM-Ward clustering resulted in better representations of the top and middle clusters than K-means alone.

As was previously mentioned, SOM-Ward clustering is a two-level clustering approach that combines the SOM and Ward's clustering algorithm. Ward's clustering is an agglomerative (bottom-up) hierarchical clustering method, which starts with a clustering in which each map node is treated as a separate cluster. The two clusters with the minimum distance are merged in each step until there is only one cluster left on the map [7]. SOM-Ward's clustering is a modification of Ward's clustering which limits cluster agglomeration to topologically neighboring nodes.

2.2 The Decision Tree

A Decision Tree is a supervised data mining technique that can be used to partition a large collection of data into smaller sets by recursively applying two-way and/or multi-way splits [17]. Compared to other data mining techniques, the Decision Tree has many advantages. First, as opposed to "black box" data mining techniques, the Decision Tree produces straightforward rules for classification and prediction purposes. Second, it is relatively insensitive to outliers [17] and skewed data distributions [18,19], and some decision tree algorithms are even capable of dealing with both numeric and nominal variables [19]. Thirdly, the decision tree is also a significant data exploration tool that can potentially be used to unveil the relationship between candidate independent and dependent variables. It can also be used to identify the significant variables for predicting the dependent variable [17]. These advantages make the decision tree applicable to a wide variety of business and marketing problems. For example, Fan et al. [20] adopted the decision tree to identify significant determinants of house prices and to predict these. Abrahams et al. [21] used decision trees to create a marketing strategy for a pet insurance company. Sheu et al. [22] adopted it to explore the potential relationship between important influential factors and customer loyalty. The findings of these studies inspire us to adopt the decision tree to explore the relationship between customers' purchase amounts and customers' demographic and behavioral characteristics, with special attention to the characteristics of high- and low-spending customers.

In this study, the CART algorithm [23] is employed to construct a binary classification tree. It is a tree-based classification method that uses recursive two-way partitioning to split the training records into segments where the records tend to fall into a specific class of the target variable. In other words, the records in the terminal nodes tend to be homogeneous with regard to the target variable. The CART algorithm employs an exhaustive search for the best independent variable for classification purposes at each split. First, the algorithm checks all possible independent variables and their potential values at each split. Then, the algorithm chooses an independent variable that maximizes within-node purity to split the node. We use the Gini index of diversity [23] to measure the improvement in purity at each split. For example, if all the records in a node fall into a specific class of dependent variable, the node is considered pure; however, if the records of each class are proportionate, the node is considered impure. This process is repeated until some user-specified criteria are met, e.g., the maximum tree depth, the minimum number of records in a node or the minimum change in within-node purity improvement [17]. When the tree growing process ends, there is a unique path from the root to each terminal node. These paths can be considered a set of if-then rules that can be used to classify the records.

2.3 The Data

The case company is a national retailer belonging to a large, multiservice Finnish corporation. The corporation uses a common loyalty card system which offers cardholders various discounts and rewards for purchases. The cardholder is required to provide basic personal information in order to register for the loyalty card, and their transactional information is collected and recorded in the system. The dataset, containing 1,480,662 customers, was obtained through the loyalty card system, and contained sales information from several department stores in Finland, for the period 2006-07. The dataset consists of ten variables that fall into two categories: *demographic* and *behavioral* variables.

The demographic variables consist of the following:

- Age
- Gender: 0 for male, 1 for female.
- Mosaic group: The Mosaic group is a socio-economic ranking system that builds upon 250-by-250 meter map grid cells covering all the populated areas of Finland. Each map grid contains an average of seven households. The ranking system combines census data with marketing research data to classify the whole population of Finland into nine groups: A, B, C, D, E, F, G, H, and I. Each map grid can be assigned to one of the nine groups. The households living in the same map grid can then be described in terms of socio-demographics, such as education, lifestyle, culture, and behavior.
- Mosaic class: Based upon the Mosaic group, the Mosaic class divides the nine Mosaic groups further into 33 subclasses.
- Estimated probability of children: This variable divides households into ten groups of equal size, based upon the probability of them having children living in the same household. A higher value in this variable indicates that the family is more likely to have children living at home. Possible values are from one to ten.
- Estimated income level: Predicts customers' income level. The higher the value, the wealthier the household is considered. Possible values are one, two and three.

The behavioral variables consist of the following:

- Loyalty point level: Based on the average spending amount per customer in the corporate chain (the case company is one service provider in the corporate chain), this variable divides customers into five classes: zero, one, two, three, and four. A higher value in loyalty point level is an indication of a customer's larger spending amount in the entire corporate chain.
- Customer tenure: Number of years since the customer's registration.

- Service level: Measures how many service providers in the corporate chain the customer has used in the last 12 months.
- The spending amount: Records the total spending amount of each customer during the period 2006-2007.

3 The SOM-Ward Model

3.1 Training the SOM-Ward Model

The training of the SOM-Ward Model was carried out using Viscovery SOMine 5.0 (http://www.viscovery.net/), which is based upon the batch SOM algorithm [8]. SOMine is a user friendly SOM implementation with a number of analytical tools embedded, including automated two-level clustering using three clustering algorithms, i.e., SOM-Ward, Ward and SOM Single Linkage [24].

To begin with, we preprocessed data to ensure the quality and validity of the clustering result. The Mosaic group is a categorical variable that is not orderly ranked. Since the SOM requires numeric input, we converted the Mosaic group into nine binary variables (either 0 or 1). In addition, the Mosiac class variable was excluded from the SOM-Ward Model as each of the 33 sub-classes of the Mosaic class would have required a dummy variable, which would made visualization extremely difficult.

Assigning a higher priority factor (default is 1) to some variables can be used to give them additional weight and importance in the training process [8], while reducing the priority factor can be used to achieve the opposite. If the priority of a variable is set to zero, the variable has no influence on the training process. We assigned the priority factor of spending amount to 1.4, aiming to give it more influence in the training process. In the pilot tests, it was discovered that the Mosaic group binary variables dominated the segmentation result, leading to clusters exclusively defined by a particular Mosaic group. Therefore, the priority factor of the Mosaic group was set to 0.1. Thus, the Mosaic group data had little influence on the segmentation result, but their distributions in the segments can be investigated when the map has been trained. The priority factors for estimated probability of children and estimated income level were set to 0.5, considering that both variables are based upon estimates and might thus involve some uncertainties. In addition, in order to achieve a more interpretable segmentation result, we slightly adjusted the priority factors of the other variables as well, again based upon the results of the pilot tests. The priority factors of age, gender, service level and customer tenure were adjusted to 1.1, 0.9, 0.9 and 0.8, respectively. Finally, the data were scaled in order to make sure that no variable received undue scale-related bias and to ease the overall training process. Total spending amount and customer tenure were scaled according to range while the rest of the variables were normalized by variance. This step was done automatically by the software. No transformation (e.g., sigmoid or logistic) was applied.

SOMine requires very few parameters for training, mainly because of the batch training process used. The user is only required to provide the *map size, map ratio* and *tension* [24]. The default map size is 1,000 nodes. By comparing a set of maps trained in the pilot tests, we chose a map containing 600 nodes to visualize the result. A smaller map is better suited for clustering [25]. The tension parameter is used to specify the neighborhood interaction. A lower tension will result in a map that adapts more to the data space, resulting in a more detailed map. On the other hand, a higher tension tends to average the data distribution on the map. Based upon the pilot test results, a map generated with the default setting (0.5) was chosen.

3.2 Analysis of the SOM-Ward Model

The characteristics of each segment were identified by examining the variables' component planes (displayed in **Fig. 1**), which show the distributions of each variable across the map. The colors of the nodes in the component planes visualize the value distribution of each variable. Cool colors (blue) indicate low values, while warm ones (red, yellow) indicate high values. Values are indicated by the color scales under the component. For instance, high spending customers were mainly found in Segments One and Two, while long-standing customers are mainly found in Segment Five. In addition to the component planes, two bar charts also illustrate the characteristics of each segment (**Fig. 2** and **Fig. 3**). The height of a bar measures the extent to which the mean value of a variable in a segment deviates from that of the entire data set. The unit of the x-axis is the standard deviation of the entire data set. In this way, both the component planes and the bar charts can visually represent the important characteristics of each segment. A description of each segment follows.

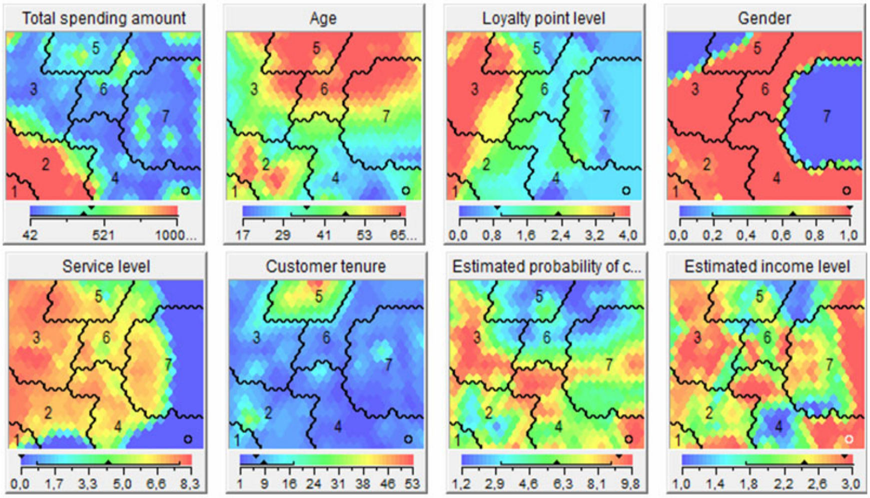


Fig. 1. The component planes of the map

Segment One: Exclusive customers

There are 25,425 customers in Segment One, accounting for 1.7% of the whole customer base. According to **Fig. 1** and **Fig. 2**, the average total purchase amount in this segment is the highest among the seven segments. These customers are mainly female, having a relatively high loyalty point level. **Fig. 3** shows that the customers belonging to this segment are most likely to belong to Mosaic groups A, C, D, and E.

Segment Two: High spending customers

In this segment, there are 177,293 customers, accounting for 12.0% of the customer base. **Fig. 1** and **Fig. 2** show that that most of them, mainly female, are high spending customers. Some of them are around 60 years old, and some have a high loyalty point level. A large percentage of them display a high service level, indicating that they also use many other service providers in the corporate chain. **Fig. 3** reveals that customers belonging to this segment are likely to be from Mosaic groups A, C, D, and E.

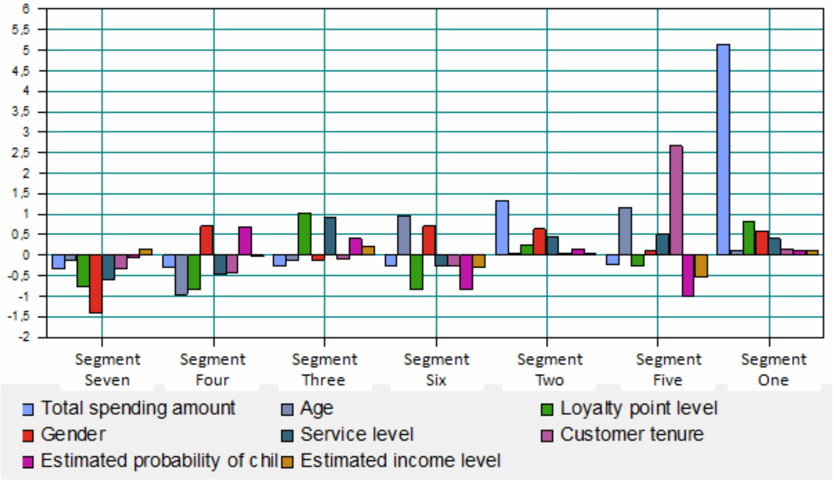


Fig. 2. Bar chart illustrating the characteristics of each segment

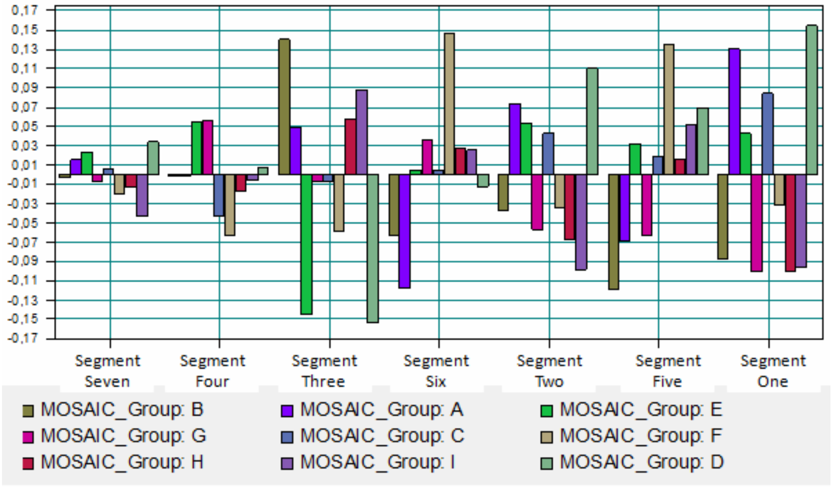


Fig. 3. Bar chart illustrating the proportion of each mosaic group in each segment

Segment Three: Customers with high loyalty point level
There are 283,265 customers in this segment, accounting for 19% of the customer base. **Fig. 1** shows that they have a very high loyalty point level. They use many other service providers in the corporate chain, but their spending amount in the case company is not large. The probability of these customers having children at home is high. **Fig. 3** shows that customers in this segment are likely to be from Mosaic groups B, H, and I.

Segment Four: Relatively young female customers
There are 303,588 customers in this segment, accounting for 20.5% of the customer base. **Fig. 1** and **Fig. 2** show that these customers are mainly females who are much younger than the average of the customer base, and some have a large probability of having children in the same household. However, their spending amount, loyalty point level, service level and customer tenure are below average.

Segment Five: Long-standing customers of the corporate chain
There are 116,537 customers in this segment, accounting for 7.9% of the customer base. **Fig. 1** reveals that these customers have been customers of the corporate chain for a long time. They widely use other service providers in this corporate chain, but their spending amount in the case company is not high. Compared to other segments, these customers are older and their estimated probability of having children living at home is small. However, some of them have a very high loyalty point level. **Fig. 3** indicates that it is likely that these customers belong to Mosaic groups D, F, and I.

Segment Six: Relatively old female customers
Segment Six has 227,194 customers, accounting for 15.3% of the customer base. **Fig. 1** shows that these customers are comparatively senior to those in other segments. Although they are senior in age, they are not long-standing customers of the corporate chain and their spending amount is not large. They are mainly female, and have a low probability of having children living in the same household. **Fig. 3** shows that these customers often belong to Mosaic group F.

Segment Seven: male customers
There are 347,410 customers in Segment Seven, accounting for 23.5% of the customer base. **Fig. 1** shows that these customers are mainly male. They have a low spending amount, and some of them have a high estimated income level.

4 The Decision Tree Model

4.1 Training the Decision Tree Model

The training of the Decision Tree Model is carried out with The PASW Decision Trees module (http://www.spss.com/statistics/).

A binary target variable, i.e., the variable of high- and low-spending customers, is created for the Decision Tree Model. From the analysis of the SOM-Ward Model, we found that the customers in Segments One and Two, i.e., those who spend much in the company, account for 13.7% of the customer base. Therefore, we arranged the customers in sequence, from the lowest to highest according to their total purchase amounts. The top 13.7% of the customers are labeled high-spending customers, and the bottom 13.7% are labeled low-spending customers. Customers in the middle, whose spending amount are not clearly high or low, were excluded from the training set. These customers will be further analyzed in Section 5.

Compared to that of the SOM-Ward Model, the data preprocessing of the Decision Tree Model is much simpler. First, the CART algorithm is able to construct a decision tree model by training continuous and/or categorical predictor variables [26]. Next, the CART algorithm uses surrogates to handle missing values on independent variables [17]. Thus, observations containing independent variables that include missing values are not excluded from the training process. Instead, other independent variables that are highly correlated with the independent variable containing missing values are used for classification. Lastly, the decision tree is relatively insensitive to outliers and skewed data distributions [17]. The above factors reduce the data preprocessing efforts required for training the decision tree. In the Decision Tree Model we used such variables as Mosaic class, which are not easily used in the SOM-Ward Model.

The maximum number of levels in the tree was limited to five and the minimum number of records in a node was set to 1,000, in order to prevent the Decision Tree from becoming very complex. A complex model, from which too many rules are extracted, would not only make the rules hard to generalize into actionable marketing strategies, but would also increase the risk of overfitting. The final model was chosen based upon ten-folds cross validation.

4.2 Analysis of the Decision Tree Model

Appendix 1 shows the created Decision Tree Model. The tree grows from left to right, with the root node (0) located on the left and terminal nodes on the right, with each non-terminal node having two child nodes. The set of two child nodes represents the answers to the decision rules for splitting the records, and these rules are printed on the lines connecting each node to its child nodes. The variable used to split the root node (node 0) is gender. The algorithm compares the classification results produced by all independent variables, and gender is found to lead to the largest improvement of within-node purity. Female customers are in the upper branch of node 0 and male customers are in the lower branch. At node 2, we find that restricting the records to female customers leads the percentage of high-spending customers to increase from 50% to 60.7%. On the other hand, at node 1, restricting the records to male customers leads the percentage of low spending customers to increase from 50% to 77.2%. After the initial split, the decision tree uses loyalty point level to further divide node 1 and node 2 into nodes 3 and 4, and nodes 5 and 6, respectively. The tree shows that a customer with a higher loyalty point level is more likely to be a high-spending customer. For example, when comparing nodes 5 and 6 (the children nodes of node 2), we find that the proportion of high value customers in node 6, i.e., female customers with a loyalty point level above 1, increases compared to that of node 1, while the proportion of high value customers in node 5 decreases compared to that of node 2. The same pattern also appears at the splits of nodes 3, 12, 28 and 30. In addition, the splits at nodes 4, 6, and 26 clearly show that customers belonging to Mosaic groups A, C, D, and E are more likely to spend more. Moreover, the splits at nodes 9 and 13 indicate that customers belonging to Mosaic classes 11, 12, and 17 are more likely to spend more in the company.

As shown in Appendix 1, the process of recursive partitioning does not stop until the tree grows to the terminal nodes. The paths to these terminal nodes describe the rules in the model. The primary focus of this analysis is to identify the characteristics of high- and low-spending customers. Therefore, among all the terminal nodes, we will select four terminal nodes with the highest percentage of high-spending customers, and two terminal nodes with the highest percentage of low-spending customers. The paths from the root node to the six selected terminal nodes are interpreted as characteristics that identify high-spending and low-spending customers.

High-spending customers – Group One: Node 27

This node has 21,526 customers, out of which 92.3% are high-spending customers. The characteristics of the customers in this node are, in order of importance:

1. They are female.
2. Their loyalty point level is larger than 1.
3. They belong to Mosaic classes 11, 12, and 17.

High-spending customers – Group Three: Node 50

This node has 22,172 customers, 83.4% of which are high-spending customers. Their characteristics are:

1. They are female customers.
2. Their loyalty point level is larger than 3. (We combined the rules applied at nodes 2 and 28, because the loyalty point level is used twice.)
3. They belong to Mosaic classes 2, 3, 9, 10, 13, 14, 15, and 16.

High-spending customers – Group Four: Node 19

This node has 1,205 customers, out of which 82.8% are high-spending customers. Characteristics of the customers in this node are:

1. They are male.
2. Their loyalty point level is larger than 3.
3. They belong to Mosaic classes 11, 12, and 17.

Low-spending customers – Group Five: Node 41

This node has 7,144 customers, out of which 98.3% are low-spending customers. Characteristics of the customers in this node are:

1. They are female.
2. Their loyalty point level is less than or equal to 3.
3. They are less than 18.5 years old.

It is also noted that node 23 (the parent node of node 41) also has a very large percentage of low-spending customers. It restricts the age to less than or equal to 20.5.

Low-spending customers – Group Six: Node 31

This node has 7,624 customers, out of which 96.9% are low-spending customers. The characteristics of the customers in this node are:

1. They are male.
2. Their loyalty point level is less than or equal to 1.
3. Their customer tenure with the corporate chain is less than 4.5 years.
4. They are less than 26.5 years old.

It is also noted that node 15 (the parent node of node 31) also has a very large percentage of low-spending customers. However, it has no restriction on age.

5 Customer Portfolio Analysis

Based upon the results of the SOM-Ward and the Decision Tree analyses, we divide the customer base into three groups: high-spending customers, low-spending customers, and customers with development potential. The SOM-Ward model shows that there are seven segments. They are:

1. Exclusive customers
2. High spending customers
3. Customers with high loyalty point level
4. Relatively young, female customers
5. Long-standing customers of the corporate chain
6. Relatively old, female customers
7. Male customers

The map shows that Segments One and Two are high-spending customers. Our purpose is to now identify which of the Segments Three, Four, Five, Six, and Seven have development potential in terms of spending amounts. We will do this by identifying segments that consist of customers displaying similar characteristics as those in Segments One and Two.

We use the decision tree to identify the characteristics that can tell high-spending customers from low-spending ones. The confusion matrix of correct and incorrect classifications in **Table 1** illustrates the accuracy of the decision tree model.

Table 1. The prediction performance of the decision tree

Observed	Predicted		
	Low-spending customers	High-spending customers	Percent Correct
Low-spending customers	125,743	75,685	62,4%
High-spending customers	34,745	166,683	82,8%
Overall Percentage	39.8%	60.2%	72.6%

Table 1 shows that the overall accuracy of the model is 72.6%. 82.8% of the high-spending customers are correctly classified, while only 62.4% of the low spending customers are correctly classified. As our main objective was to build a model that can identify potential high-spending customers, the cost of incorrectly classifying a high-spending customer as a low-spending one is higher than the cost of incorrectly classifying a low-spending customer as a high-spending one. Therefore, this accuracy rate is acceptable.

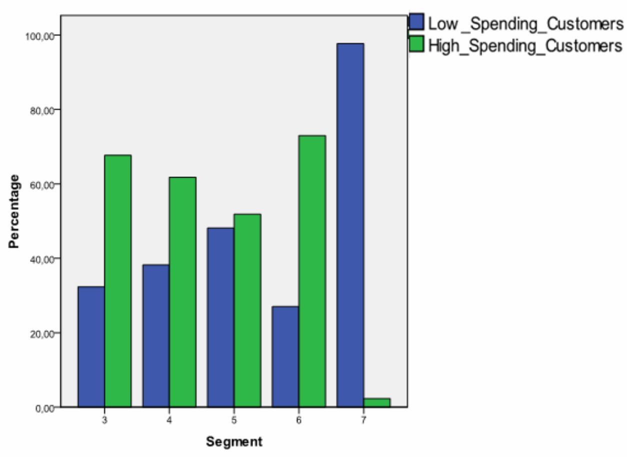


Fig. 4. Percentages of customers in each segment identified as having development potential

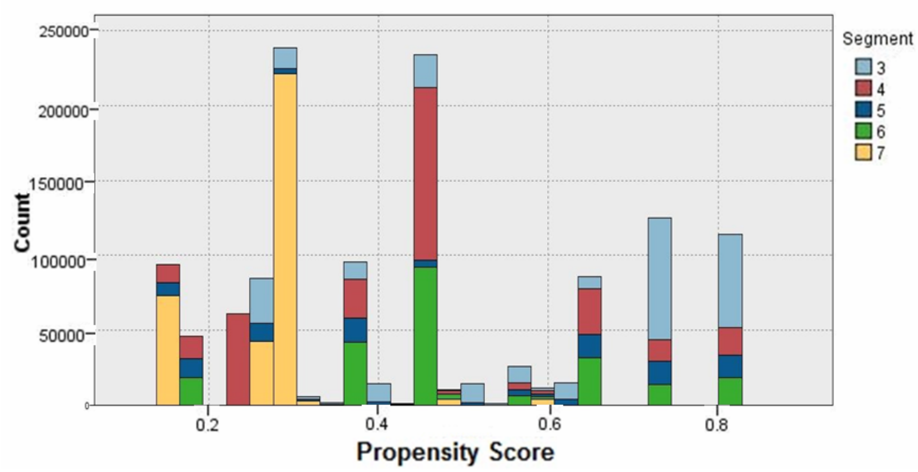


Fig. 5. The propensity scores for the different segments

We then ran all the cases in Segments Three, Four, Five, Six, and Seven through the decision tree model that was created. The ones that we can identify as possessing the same characteristics as the high–spending customers will be our potential group. As the clustered bar chart in **Fig. 4** indicates, the customers in Segments Six and Three have more development potential than the customers in the other segments. Each terminal node is a mixture of high-spending customers and low-spending customers. The predicted value is the category with the highest proportion of cases in the terminal node for each case. Therefore, it is possible to use the model to predict these unclassified cases using propensity scores. The propensity score ranks the likelihood of the prediction from 1 (likely to be a high-spending customer) to 0 (not likely to be a high-spending customer). For example, if an unclassified case has the same characteristics as node 27 in the decision tree model, it will be assigned a propensity score of 0.923, as 92.3% of the cases in node 27 are high-spending customers. The Histogram of the distribution of propensity scores of Segments Three, Four, Five, Six, and Seven is shown in **Fig. 5**.

This figure reveals that the customers in Segment Three are most likely to be potential high value customers, while the customers in Segment Seven are least likely to be potential high value customers. After running all of the data outside of the 27.4% (high and low spending) that were used to train the decision tree, we obtain a list of those customers with their propensity scores or/and predictions appended.

6 Conclusions

A hybrid approach combining unsupervised and supervised data mining techniques has been proposed to conduct customer portfolio analysis. SOM-Ward clustering was first used to conduct customer segmentation. Then, the decision tree was employed to gain insight into whether there are significant determinants for distinguishing between high- and low-spending customers. The results of the two models are then compared and combined to perform customer portfolio analysis, i.e., to identify the high- and low-spending customers, as well as customers with development potential. Each model possesses advantages and disadvantages of its own.

As an unsupervised data mining technique, SOM-Ward clustering is a good tool for exploratory analysis, as is the case when no *a priori* classes have been identified. The SOM is a very visual tool and possesses strong capabilities for dealing with nonlinear relationships, missing data, and skewed distributions. However, while the clusters produced using unsupervised methods may be good for gaining an understanding of the customer base, they are not necessarily actionable in terms of marketing strategy as they are not based upon any identified target or aim. In addition, using detailed nominal data (e.g., 33 Mosaic classes) is a problem when using the SOM, as binary variables must be constructed for each potential class. This easily clutters the map and heavily influences training results.

Decision trees, on the other hand, are tailored to a specific purpose by using a supervised learning approach. The decision tree is also a very robust method, easily capable of dealing with difficult data, and requires less data preprocessing and setting of parameters than the SOM. However, the starting point of supervised learning inevitably requires more *a priori* knowledge than unsupervised learning, making the knowledge gained using the SOM potentially very important.

The results of the analysis demonstrate that the combined method of the SOM-Ward clustering and the Decision Tree can potentially be effective in conducting market segmentation. The information provided by the combined model is more detailed and accurate than that provided by either model used alone, thus more actionable information about the customer base for marketing purposes could be retrieved.

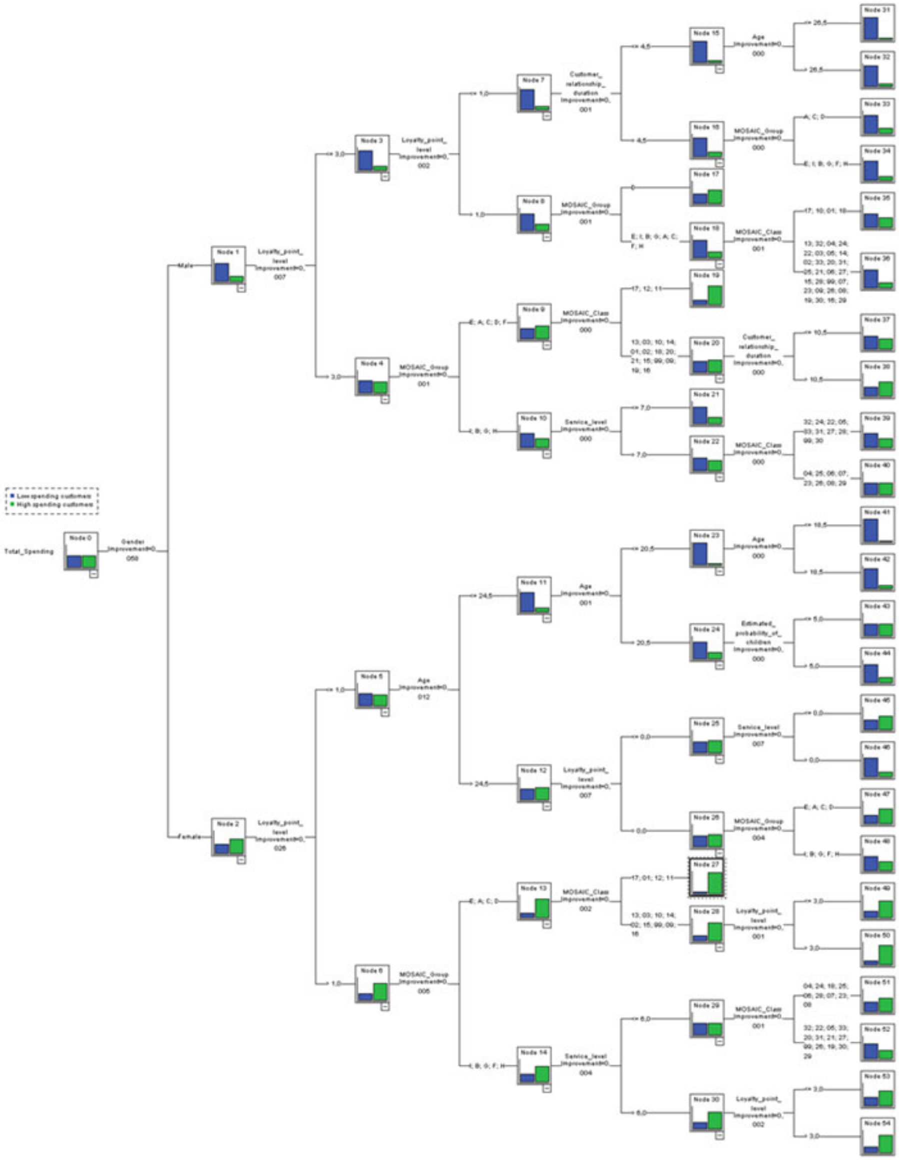
Acknowledgements. The authors gratefully acknowledge the financial support of the National Agency of Technology (Titan, grant no. 40063/08) and the Academy of Finland (grant no. 127656). The case organization's cooperation is also gratefully acknowledged.

References

1. Kim, S., Jung, T., Suh, E., Hwang, H.: Customer Segmentation and Strategy Development Based on Customer Lifetime Value: A Case Study. Expert Systems with Applications 31, 101–107 (2006)
2. Park, H., Baik, D.: A Study for Control of Client Value using Cluster Analysis. Journal of Network and Computer Applications 29, 262–276 (2006)
3. Reichheld, F.F., Teal, T.: The Loyalty Effect: The Hidden Force Behind Growth, Profits, and Lasting Value. Harvard Business Press, Boston (2001)
4. Buttle, F.: Customer Relationship Management Concepts and Tools. Butterworth-Heinemann, Oxford (2004)
5. Terho, H., Halinen, A.: Customer Portfolio Analysis Practices in Different Exchange Contexts. Journal of Business Research 60, 720–730 (2007)
6. Kohonen, T.: Self-Organization and Associative Memory. Springer, New York (1988)
7. Ward, J.H.: Hierarchical Grouping to Optimize an Objective Function. Journal of the American Statistical Association 58, 236–244 (1963)
8. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (2001)
9. Kaski, S., Kangas, J., Kohonen, T.: Bibliography of Self-Organizing Map (SOM) Papers 1981-1997. Neural Computing Surveys 1, 102–350 (1998)
10. Oja, M., Kaski, S., Kohonen, T.: Bibliography of Self-Organizing Map (SOM) Papers: 1998-2001 Addendum. Neural Computing Surveys 3, 1–156 (2003)
11. Deboeck, G.J., Kohonen, T.: Visual Explorations in Finance with Self-Organizing Maps. Springer, Berlin (1998)
12. Holmbom, A.H., Eklund, T., Back, B.: Customer Portfolio Analysis using the SOM. In: Proceedings of 19th Australasian Conference on Information Systems (ACIS 2008), pp. 412–422 (2008)
13. D'Urso, P., Giovanni, L.D.: Temporal Self-Organizing Maps for Telecommunications Market Segmentation. Neurocomputing 71, 2880–2892 (2008)
14. Lee, S.C., Gu, J.C., Suh, Y.H.: A Comparative Analysis of Clustering Methodology and Application for Market Segmentation: K-Means, SOM and a Two-Level SOM. In: Zhong, N., Ras, Z.W., Tsumoto, S., Suzuki, E. (eds.) ISMIS 2006. LNCS (LNAI), vol. 4203, pp. 435–444. Springer, Heidelberg (2006)
15. Vesanto, J., Alhoniemi, E.: Clustering of the Self-Organizing Map. IEEE Trans. Neural Networks 11, 586–600 (2000)
16. Samarasinghe, S.: Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition. Auerbach Publications, Boca Raton (2007)
17. Berry, M.J.A., Linoff, G.S.: Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management. Wiley Publishing Inc., Indianapolis (2004)
18. Murthy, S.K.: Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey. Data Mining and Knowledge Discovery 2, 345–389 (1998)
19. Rokach, L., Maimon, O.: Data Mining with Decision Trees: Theory and Applications. World Scientific Publishing, Singapore (2008)
20. Fan, G.Z., Ong, S.E., Koh, H.C.: Determinants of House Price: A Decision Tree Approach. Urban Studies 43, 2301–2315 (2006)
21. Abrahams, A.S., Becker, A.B., Sabido, D., D'Souza, R., Makriyiannis, G., Krasnodebski, M.: Inducing a Marketing Strategy for a New Pet Insurance Company using Decision Trees. Expert Syst. Appl. 36, 1914–1921 (2009)

22. Sheu, J.J., Su, Y.H., Chu, K.T.: Segmenting Online Game Customers - the Perspective of Experiential Marketing. Expert Syst. Appl. 36, 8487–8495 (2009)
23. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Wadsworth, Pacific Grove (1984)
24. Deboeck, G.J.: Software Tools for Self-Organizing Maps. In: Deboeck, G.J., Kohonen, T. (eds.) Visual Explorations in Finance using Self-Organizing Maps, pp. 179–194. Springer, Berlin (1998)
25. Desmet, P.: Buying Behavior Study with Basket Analysis: Pre-Clustering with a Kohonen Map. European Journal of Economic and Social Systems 15, 17–30 (2001)
26. Hill, T., Lewicki, P.: Statistics: Methods and Applications: A Comprehensive Reference for Science, Industry, and Data Mining. StatSoft, Inc., Tulsa (2006)

Appendix 1. The Decision Tree Model



Managing Product Life Cycle with MultiAgent Data Mining System

Serge Parshutin

Riga Technical University, Institute of Information Technology,
1 Kalku Str., Riga, Latvia, LV-1658
serge.parshutin@rtu.lv

Abstract. Production planning is the main aspect for a manufacturer affecting an income of a company. Correct production planning policy, chosen for the right product at the right time, lessens production, storing and other related costs. The task of choosing a production policy in most cases is solved by an expert group, what not an every company can support. Thus a topic of having an intelligent system for supporting production management process becomes actual. The main tasks such system should be able to solve are defining the present Product Life Cycle (PLC) phase of a product as also determining a transition point - a moment of time (period), when the PLC phase is changed; as the results obtained will affect the decision of what production planning policy should be used.

The paper presents the MultiAgent Data Mining system, meant for supporting a production manager in his/her production planning decisions. The developed system is based on the analysis of historical demand for products and on the information about transitions between phases in life cycles of those products. The architecture of the developed system is presented as also an analysis of testing on the real-world data results is given.

Keywords: MultiAgent system, Agents, Data Mining, Product Life Cycle Management.

1 Introduction

Constantly evolving computer technologies are becoming more and more an inherent part of successful enterprise management and keeping its activity at a high level. Different institutions are trying to reduce their costs by fully automatising certain stages of manufacturing process as well as introducing various techniques intended for forecasting certain market indicators that impact general manufacturing process. Different statistical methods are employed as well, though an increasing interest in computational intelligence technologies and their practical application can be observed ever more.

The present research focuses on studying the ability to create the MultiAgent system that will integrate several Data Mining technologies in order to support a human's decision in a real-world problem.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 308–322, 2010.

A task of product life cycle phase transition point forecasting can serve as an example of such problem where both Data Mining and Decision Support technologies should be applied. From the viewpoint of the management it is important to know, in which particular phase the product is, as it will have an impact on the production planning policy that will be chosen [10]. In the case of determined demand changing boundaries, typical for a maturity phase, it is possible to apply cyclic production planning policy [2], whereas for the introduction and decline phases an individual planning is usually employed, as the demand is less stable comparing to the maturity phase. The chosen right production policy will have a positive influence on the state of a company.

As the technologies are evolving, the variability of products grows, making manual monitoring of PLCs a difficult and costly task for companies. Thus an alternative of having an autonomous intelligent system that monitors market data, creates and automatically updates lists of products for what it is reasonable to consider a production planning policy update or replacement, becomes more valuable.

The managing a PLC and forecasting the phase change period is one complicated non-linear task. The situation on market changes dynamically, therefore the system, designed for such task, should be able to function in dynamically changing environment. The Agent Technology is one of the modern technologies that can be applied for building an intelligent system for monitoring a dynamic environment.

This paper proposes a model of MultiAgent system that ensures the solving of the aforementioned task as well as provides an analysis of system testing results.

2 Problem Statement

Any created product has a certain life cycle. The term ”life cycle” is used to describe a period of product life from its introduction on the market to its withdrawal from the market. Life cycle can be described by different phases: traditional division assumes such phases like introduction, growth, maturity and decline [9]. For products with conditionally long life cycle, it is possible to make some simplification, merging introduction and growth phases into one phase - introduction.

An assumption that three different phases, namely, introduction, maturity and end-of-life are possible in the product life cycle, gives us two possible transitions. The first transition is between introduction and maturity phases and the second - between maturity and product's end-of-life.

From the side of data mining [4,6,7], information about the demand for a particular product is a discrete time series, in which demand value is, as a rule, represented by the month. A task of forecasting a transition points between life cycle phases may be formulated as follows. Assume that $D = \{d_1, \ldots, d_i, \ldots, d_n\}$ is a dataset and $d = \{a_1, \ldots, a_j, \ldots, a_l\}$ is a discrete time series whose duration equals to l periods, where $l \in L = \{l_1, \ldots, l_h, \ldots, l_s\}$ and varies from record to record in the dataset D. For simplification, the index of d is omitted. Time series

d represents a particular phase of a product life cycle, say introduction. Assume that for a particular transition, like introduction to maturity, a set of possible transition points $P = \{p_1, \ldots, p_k, \ldots, p_m\}$ is available. Having such assumptions the forecasting of a transition point for a new product, represented by a time series $d' \notin D$, will start with finding an implication between historical datasets D and P, $f : D \to P$; followed by application of found model to new data.

3 Structure of the MultiAgent System

The developed system contains three main elements - Data Management Agent, Data Mining Agent and Decision Support Agent, shown in Figure 1. The system intentionally was designed simple and general, as to give the possibility to apply it not only to PLC phase transition point forecasting task, but also to any other task that encapsulates the clustering of time series with special markers and forecasting a marker value for a new object, represented only by a time series.

To avoid misunderstandings of several terms the definitions of an *Agent*, *Intelligent Agent*, *Multiagent System* and *Controlled Agent Community* are given in scope of the present research.

There is no universally accepted definition of the term agent, nevertheless, some sort of definition is important: An agent is a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives [13]. The present definition fully meets all objectives of a term *Agent* in the present research.

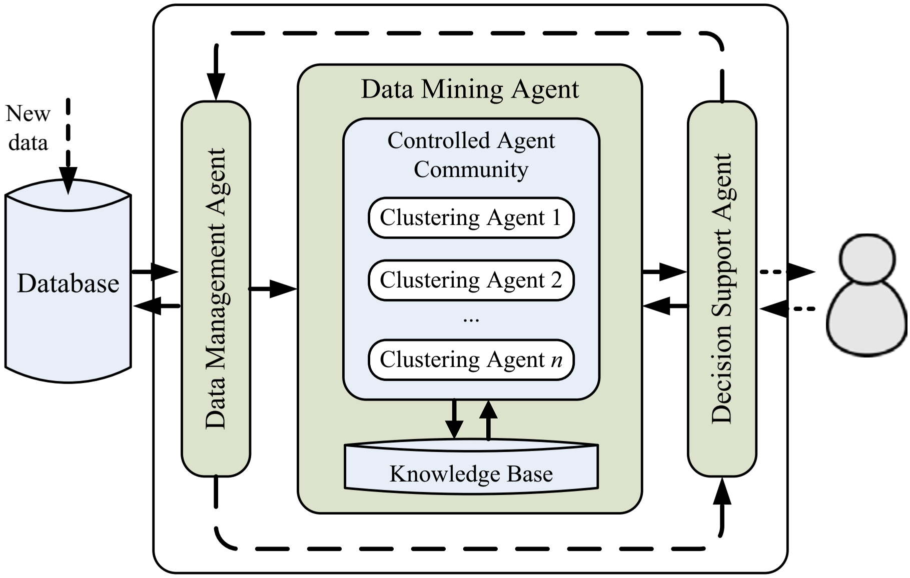


Fig. 1. MultiAgent system

An intelligent agent can be defined as an agent that is not only capable of autonomous action, but also can learn - extract and gather knowledge about the environment it is situated in. The ability to learn allows an intelligent agent to build a model of an environment.

The MultiAgent system contains a number of agents and intelligent agents, which interact with one another through communication. The agents are able to act in an environment; different agents have different 'spheres of influence', in the sense that they will have control over - or at least be able to influence - different parts of the environment [13].

Main difference between a MultiAgent system and an agent community is that in agent community agents act in the environment, but do not interact with one another. Having such definitions the agents of the MultiAgent system (Figure 1) can be described as follows.

Data Management Agent. The Data Management Agent performs several tasks of managing data. It has a link to the database that contains the product demand data and regularly is updated. The Data Management Agent handles the preprocessing of the data - data normalization, exclusion of obvious outliers and data transformation to defined format before sending it to the Data Mining Agent. Such preprocessing allows to lessen the impact of noisiness and dominance of data [3,12]. The transformed data record displays the demand for a product, collected within known period of time, the length of which is set by the system - day, week, month, etc. Each data record is marked with one or both markers - transition indicators; namely, *M1* indicates the period when product switched from Introduction phase to Maturity phase; and *M2* indicates the period when product switched from Maturity phase to End-of-Life phase. Marks on transitions can guarantee that a model will be build; if we have patterns of transitions in historical data, then, theoretically, in presence of a model for generalisation, we are able to recognise those patterns in new data. Assigning markers to the demand time series is one process that currently is done manually by a human expert.

As the database is regularly updated, the Data Management Agent monitors the new data and at a defined moment of time forwards the subset of new data to the Data Mining Agent for updating the knowledge base.

Data Mining Agent. The Data Mining Agent is an intelligent agent. The actions, performed by the Data Mining Agent, cover such processes as initialization of a training process, performing system training and testing processes, imitation of On-line data flowing during system training and testing, Knowledge base creation and maintaining it up-to-date.

The Data Mining Agent contains a knowledge base that is connected with the controlled community of clustering agents. Clustering agents mine knowledge from data and gather it in the knowledge base. Each clustering agent either can handle records, with equal duration l or can proceed with time series with different durations. Which option will be selected depends on the total load distribution policy, currently defined by the user. Let us illustrate the load distribution. Assume that the duration of discrete time series in dataset D varies from 4 to 9 periods, thus $l \in L = \{4, 5, \ldots, 9\}$. In this case, total system load will consist of six values that time series duration can take. Let us assume that the load has to be distributed uniformly over clustering agents at the condition that an individual load on separate agent should not exceed three values that is $q = 3$. Under such conditions, two clustering agents will be created by the moment of system initialisation. The first clustering agent, ca_1, will process time series of duration $l \in \{4, 5, 6\}$; the remaining time series with duration of 7, 8 and 9 periods will be sent to the second clustering agent, ca_2.

The option of having the On-line data flowing procedure becomes necessary due to specifics of a chosen system application environment. In the real-world situation, the information about the demand for a new product is becoming available gradually: after a regular period finishes, new demand data appear. Due to that the system must be trained to recognize transition points that will occur in the future having only several first periods of the demand time series available.

The algorithm employed is executed taking into account the following details. Preprocessed by the Data Management Agent time series d with duration l, containing demand data within introduction or maturity phase and the appropriate marker (*M1* or *M2* respectively) with value p, is sent to the Data Mining Agent. Having that minimal duration of a time series should be l_{min} and greater, the algorithm of On-line data flowing procedure will include these steps:

1. Define $l^* = l_{min}$;
2. Process first l^* periods of a record d with a marker value p;
3. If $l^* < l$ then increase value of l^* by one period and return to step 2; else proceed to step 4;
4. Finish processing record d.

According to the chosen policy of system load distribution the fraction of a time series with marker is directed to the clustering agent responsible for processing the time series of a specific duration.

Each clustering agent implements a specific clustering algorithm, capable of processing times series of different duration. The present research considers a possibility of using a Gravitational Clustering algorithm - the G-Algorithm [5] for extracting knowledge from data.

Decision Support Agent. The Decision Support Agent is the element that uses the knowledge base of the Data Mining Agent to forecast a transition points for new products as also to analyse the alternatives of using cyclic or non-cyclic planning policies for particular products. The Decision Support Agent follows a certain algorithms, whose examples are provided in the subsection 3.1.

3.1 System Functioning Algorithm

The system functioning algorithm consists of two main stages - System Learning and System Application. The System Learning stage contains two inner steps

- System Training followed by System Testing and Validation. The System Application stage is the stage when the Decision Support Agent receives requests from the user and following certain algorithms, described in current subsection, provides the user with certain information.

System Learning. The system learning process is fired when the Data Management Agent sends prepared data to the Data Mining Agent with "Start initial learning" or "Update" command. The "Start initial learning" command is sent when the system is launched for the first time. This will fire the process of determination and setting of basic system's parameters: the number of clustering agents, learning coefficients, number of iterations. The number of clustering agents in the agent community, n, is calculated empirically. Given a policy assuming uniform distribution of general load among the agents, formula (1) can be used to calculate the number of clustering agents.

$$n = \left\lceil \frac{|L|}{q} \right\rceil \ , \tag{1}$$

where q - each clustering agent's individual load; $\lceil\cdot\rceil$ - symbol of rounding up.

After the number of clustering agents n is calculated, for each clustering agent ca_i an interval of time series durations $[l_{i,min}; l_{i,max}]$ is set. The records with duration $l \in [l_{i,min}; l_{i,max}]$ will be processed by an agent ca_i. Given a uniform load distribution, equation (2) can be used for setting the bounds.

$$\begin{cases} i = 1, & l_{i,min} = l_{min} \ , \\ i > 1, & l_{i,min} = l_{i-1,max} + 1 \ ; \end{cases} \\ l_{i,max} = l_{i,min} + q - 1 \ . \tag{2}$$

As the main parameters are set, Data Mining Agent processes each of the received records imitating the On-line data flowing. Fractions of time series are forwarded to the corresponding clustering agent ca_i, defined by the $[l_{i,min}; l_{i,max}]$ interval, calculated with equation (2). Each clustering agent will process an individual part of the data.

Data clustering process is based on the Gravitational Clustering Algorithm, described in [5]. The G-Algorithm is based on the Gravitational Law and the Second Newtont's Motion Law and allows to find clusters in data without any predefined information about the number of clusters. The main aspects of the algorithm is that there is a gravitational force among all objects in the dataset. The more similar objects are - the smaller is the distance between them, and the stronger is the gravitational force - which means that similar objects are moving towards each other forming clusters.

To start using the G-Algorithm some major parameters must be set. First one will be the Universal gravitational constant G, the value of which equals to $6.67 * 10^{-11}$. Analysing the results in [5] it is possible to conclude that the value of G should be chosen carefully for each dataset, as there is no "universal" value for this parameter, that works for all datasets. The big value for G will result in that only one cluster will be created and vice versa - too low value will not allow the G-Algorithm to create any clusters. In parallel with G a decay for a gravitational force, ΔG, should be stated, which also can be equal to zero. In that case the gravitational force remains the same during all iterations.

The number of iterations for the algorithm is set empirically. Another option that can be used as an alternative to a defined number of iterations is to launch algorithm until only one merged object remains and save status each Z iterations. This will give an information to analyse and come up with a suitable number of iterations for a current dataset.

Finally the Merging Distance - ϵ, should be defined. This parameter defines the minimal distance between two object at which they can be merged into a single object. The value of ϵ can be either defined as a number or as a percentage of a maximal distance between any two objects in the dataset. With respect to [5] the most suitable merging distance was equal to 0.0001, what will be near 0.01% of maximal distance between any two objects in the dataset, used by authors of [5]. If ϵ is set as a percentage of a maximal distance, then during the first iteration the actual value for this parameter can be calculated and saved for futher application.

Initially all objects in the dataset have their masses equal to 1. The mass of a merged object can either remain 1 or become equal to the sum of masses of the objects that were merged. This will influence the gravity force between objects. The force exerted from one object x over another object y can be expressed with equation (3).

$$F(t) = \frac{G \cdot m_x \cdot m_y}{\|\overrightarrow{d(t)}\|^2} \ , \tag{3}$$

where G is the gravitational constant, m_x and m_y are the masses of the two objects and $\overrightarrow{d(t)}= y(t) - x(t)$, is the vector that defines the direction of the force.

Therefore, the movement equation of an object x under the influence of the gravitational field of an object y are:

$$x(t + \Delta t) = x(t) + v(t)\Delta t + \overrightarrow{d(t)} \ \frac{G \cdot m_y \cdot \Delta t^2}{2 \cdot \|\overrightarrow{d(t)}\|^3} \tag{4}$$

Assuming that an object velocity at moment t, $v(t)$, is a zero vector and $\Delta t = 1$, a simplifications to equation (4) can be made [5]. Equation (5) can be used as a simplified one for calculating a change in position of an object x induced by a force exerted from object y to object x.

$$x(t + 1) = x(t) + \overrightarrow{d(t)} \ \frac{G \cdot m_y}{2 \cdot \|\overrightarrow{d(t)}\|^3} \tag{5}$$

Note how the vector $\overrightarrow{d(t)}$ that defines direction of the force, is calculated. When the load is $q = 1$, that is when each clustering agent is processing discrete time series with a certain fixed duration, the vector $\overrightarrow{d(t)}$ is calculated as the difference between vectors of equal length (6).

$$\overrightarrow{d(t)} = \overrightarrow{y(t)} - \overrightarrow{x(t)} \qquad (6)$$

In other cases when $q \geq 2$, the direction vector of the force is calculated using a specific approach, suggested in the scope of present research. Assume that the force is exerted from y to x. In case when durations of time series are equal, that is $l_x = l_y$, the direction vector of the force is calculated using equation (6). In cases when $l_x < l_y$ or $l_x > l_y$ a presented algorithm is used:

1. Find a minimal duration l' among l_x and $l_y : l' = \min\{l_x, l_y\}$;
2. Use equation (6) and only first l' periods of time series x and y to calculate vector $\overrightarrow{d}$.

In other words in case when durations of time series x and y are not equal, the vector $\overrightarrow{d}$ is calculated only in l' dimensions, where $l' = \min\{l_x, l_y\}$.
Let us demonstrate an example of such case. Assume that duration of time series x, l_x, is equal to 4 periods and duration of time series y, l_y, is equal to 6 periods. Vector $\overrightarrow{d}$ represents direction of the force exerted from object y to object x. The minimal duration, l', will be equal to the duration of time series x, that is $l' = l_x = 4$. For calculation of $\overrightarrow{d}$ all 4 periods of time series x and only first 4 of 6 periods of time series y will be used. That is $\overrightarrow{d}$ will be calculated in 4 dimensions.

When the clustering is finished the clusters must be formed and the Data Mining Agent launches this process. The parameter α defines the minimal cluster size - the minimal number of objects in one cluster. Setting $\alpha = 2$ will result in that any object, created by merging 2 or more single objects, will be treated as a cluster. Other objects that do not match the parameter α condition will be treated as outliers. Each cluster contains the next data:

1. The ID of the clustering agent, where the cluster was formed;
2. The time series, representing the centroid of all objects in the cluster;
3. Statistics, giving the picture of what transition points are present in the cluster and what part of objects has a certain transition point. This statistics will be used for choosing a preferable transition point for cluster.

Taking into account the point that a single cluster can contain time series with different durations (for cases with $q \geq 2$), the next strategy is applied for a cluster centroid calculation:

1. Number of periods in a cluster centroid is equal to the maximal duration of times series in that cluster;
2. For each period i repeat:
   (a) Set SV_i as a summarized value of period i from all m time series with a value in period i;
   (b) Set a centroid value in period i equal to SV_i divided by m.

Clusters will be included in the knowledge base. The knowledge base contains a number of levels, equal to the number of clustering agents in the Controlled Agent Community. Each level contains clusters from a clustering agent with corresponding number.

System evaluation. To evaluate the precision of transition point forecasts made by the system, two criteria are employed: Mean Absolute Error - MAE, to evaluate the accuracy of the system and Logical Error to evaluate whether decisions made by the system are logically correct.

The Mean Absolute Error (MAE) is calculated using formula (7).

$$MAE = \frac{\sum_{i=1}^{k} |p_i - r|}{k} \quad i = [1, 2, \ldots, k] \ , \tag{7}$$

where k - the number of records used for testing; p_i - real value of the key parameter for record d_i; r - the value of the key parameter forecasted by the system.

Logical error provides information about the logical potential of the system. To calculate the logical error, it is necessary to define logically correct and logically incorrect decisions. As applied to the task of forecasting product life cycle phase transition period, logically correct and logically incorrect decisions are defined:

1. Assume that a discrete time series d has a duration equal to l_d, but the value of the key parameter - the period of product life cycle phase transition, is $p = p_d$, where $p_d > l_d$. This statement means that a real transition period has not come yet. Due to that, logically correct decision is to forecast a transition period r_d, where $r_d > l_d$. Logically incorrect decision in this case will be if $r_d \leq l_d$.
2. Assume that a discrete time series d has a duration equal to l_d, but the value of the key parameter - the period of product life cycle phase transition, is $p = p_d$, where $p_d \leq l_d$. This statement gives evidence that real transition moment has already come. Due to that, logically correct decision could be forecasting transition period r_d, where $r_d \leq l_d$. And logically incorrect decision will take place if $r_d > l_d$.

The statement that at $r_d = l_d$ transition has occurred can be considered correct as the availability of data about some period in record d shows that the period is logically finished and, consequently, the transition - if any was assumed in this period - is occurred.

Functional aspects of the Decision Support Agent. The Decision Support Agent can perform its actions either by receiving a request from a user, or in autonomous mode, with defined interval of time (at the end of each period) reporting the decision analysis results. Products that are analysed by Decision Support Agent are products that still are evolving on the market. The list of such products is monitored by the Data Management Agent.

Either in autonomous mode or by request from a user the Decision Support Agent starts the process, displayed in Figure 2. The depicted process includes

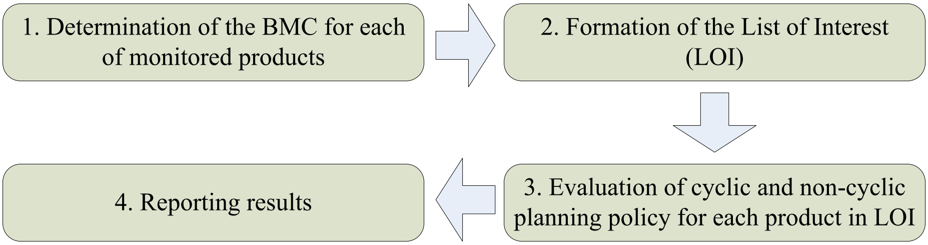

1. Determination of the BMC for each of monitored products
2. Formation of the List of Interest (LOI)
3. Evaluation of cyclic and non-cyclic planning policy for each product in LOI
4. Reporting results

Fig. 2. Decision Support Agent functioning diagram

three main steps - Determination of the Best Matching Cluster (BMC) for each of the evolving products; Formation of the List of Interest (LOI), the list of products for which it would be reasonable to reconsider the planning policy; Evaluation of a cyclic and non-cyclic planning policy for each product in the List of Interest. And finishes with the fourth step - Reporting the results of the evaluation to a user. Let us describe the processes hidden behind each of the main steps.

Step 1: Determination of the BMC for each of the evolving products. The Decision Support Agent sends a request to the Data Management Agent and receives a dataset containing evolving products. Each record is preprocessed and formatted by the Data Management Agent. As the dataset is received it being sent to the Data Mining Agent with command ”Find the Best Matching Cluster”.

For each product the Data Mining Agent first finds a clustering agent ca_i that processes demand time series with duration same to the product time series has. This will give an information about the level in the knowledge base on which the BMC will be searched. Then the Data Mining Agent finds the Best Matching Cluster among cluster on the specified level of the knowledge base. As the BMC is found for each product, the Data Mining Agent sends a list of found BMCs to the Decision Support Agent.

The information from the BMC contains a list of possible transition points for each product - for the products in the introduction phase the *M1* transition point is supplied and *M2* transition point - for products in the maturity phase. This is where the Formation of the LOI begins.

Step 2: Formation of the List of Interest. The Best Matching Cluster may contain different information for products and several cases are possible:

1. The simplest case (*C1*) occurs when the Best Matching Cluster contains only one possible transition point. In this case the Decision Support Agent assumes this transition point as the preferable one and follows a solution (*S1*) containing three major rules:
   (a) If $l < p$ and $p - l > \theta$ Then: Product remains monitored and is not included in the List of Interest;
   (b) If $l < p$ and $p - l \leq \theta$ Then: Product is included in the List of Interest;

   (c) If $l \geq p$ Then: Product is included in the List of Interest.

   Where l is the duration of the demand time series in periods; p - a forecasted transition point; and variable θ stores the minimal threshold of interest for either including the product in the List of Interest or not.

2. The case (*C2*) occurs when the BMC contains more than one possible transition points for a product, but one of transition points has an expressed appearance frequency f. An appearance frequency may be stated as expressed if it exceeds some threshold, like 50%. In such case the solution (*S2*) will be that the Decision Support Agent accepts the transition point with an expressed f as preferable one and follows rules from the *S1* solution.
3. The third possible case (*C3*) occurs when the BMC contains several possible transition points, but there is no one with an expressed appearance frequency present. For this case several solutions are possible:
   (a) Solution *S3*. The Decision Support Agent queries the Data Mining Agent if the BMC was found on the last (highest) level of the knowledge base. If so, then the Decision Support Agent follows the next two rules:
      i. If only one transition point has the highest (not expressed) f, then select it as a preferable one and follow rules from solution *S1*;
      ii. If several transition points have the highest f, then select a transition point with a minimal value as preferable one and follow rules from solution *S1*. Current rule application example would be if transition points with highest f are 6th, 8th and 10th period then the Decision Support Agent will choose the 6th period, as it is the minimal one.

   (b) Solution *S4*. If the solution *S3* was not triggered, then the Decision Support Agent follows the next strategy:
      i. Send a request to the Data Mining Agent to find for a current product the Best Matching Clusters among clusters in knowledge base levels that are higher than a level with the current BMC;
      ii. The Data Mining Agent searches for BMCs in the knowledge base using only first l' periods, where $l' = l_i$ and l_i is the duration of the demand time series of the current product;
      iii. The Data Mining Agent returns the list of BMCs to the Decision Support Agent;
      iv. Decision Support Agent adds the current BMC to the list, selects the most matching one and checks which of three cases - *C1*, *C2* or *C3*, is triggered. If case *C1* or *C2* is triggered, then the Decision Support Agent just follows the rules from solutions for those cases (solutions *S1* and *S2* respectively). In case when the *C3* is triggered again, the Decision Support Agent follows rules from the *S3* solution.

      Example of such situation may be described as follows. Assume that the Controlled Agent Community contains two clustering agents: first one, ca_1, processes time series with duration 4, 5 and 6 periods; second clustering agent, ca_2, is processes time series with duration 7, 8 and 9 periods. Respectively the knowledge base will contain two levels for clusters from each of clustering agents. Time series of a current product has duration equal to 6 periods. Time series with such duration are processed by the agent ca_1. Due to that, the BMC will be searched among clusters on the first level of the knowledge base. Assume that case $C3$ occurred and solution $S4$ was triggered. In that case the Data Mining Agent will search for a BMC on the second level of the knowledge base (as it is the only higher level than the one with initial BMC), but only using first 6 periods for distance calculation. The list of BMCs will contain two objects. Finally the most similar BMC will be selected from the list of BMCs, the preferable transition point will be selected and the forecast will be made.

If at the end of formation of the List of Interest the list is empty then the Decision Support Agent bypasses the third step and reports that products, for which it would be reasonable to reconsider the planning policy, were not found. In case when LOI contains at least one product the Decision Support Agent starts processes in the third step.

Step 3: Evaluation of cyclic and non-cyclic planning policy for each product in LOI. At this step the Decision Support Agent measures an expenses of using cyclic or non-cyclic planning policy for each product in the List of Interest. As stated in [1,2,8] the measure of Additional Cost of a Cyclic Schedule ($ACCS$) may be used for those purposes. The $ACCS$ measures the gap between cyclic and non-cyclic planning policies, and is calculated by formula (8).

$$ACCS = \frac{CPPC - NCPPC}{NCPPC} \ , \tag{8}$$

where $CPPC$ is the Cyclic Planning Policy Cost and $NCPPC$ - the Non-Cyclic Planning Policy Cost.

As the third step is finished the user receives analysis results of the products from the List of Interest.

4 Gathered Results

The fact that the data describes real life process and marks of transitions were given by experts implies that some noisiness in data is present.

The obtained dataset contains 312 real product demand time series with minimal duration equals to 4 and maximal - to 23 periods. Each time series contains the demand during the introduction phase of a specific product and is marked with *M1* marker. To normalize the data, the Z-score with standard deviation normalization method was applied. As the true bounds of the demand data in the dataset are unknown and the difference between values of various time series is high, the chosen normalization method is one of the most suitable ones.

Figure 3 displays an example of time series data used in experiments. As can be seen, the time series differs not only in duration, but also in amplitude and its pattern.

A specific tool for testing the proposed MultiAgent system was developed using MS Visual Basic 2008. The system was tested using a 10-fold cross-validation

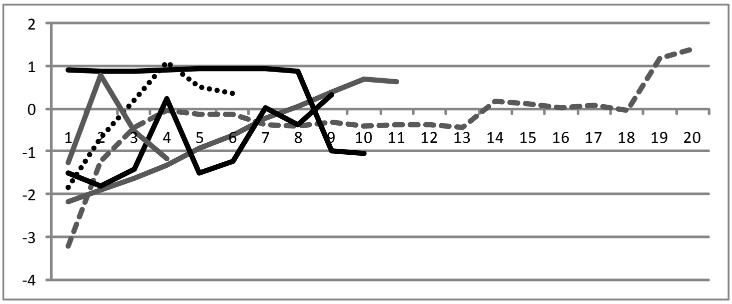


Fig. 3. Example of normalized data

technique. Eight experiments with different conditions were performed, totally giving 80 system runs. In all experiments the minimal number of elements in the cluster was 2 elements and the minimal merging distance was set as 0.5% of a maximal distance found during the first iteration. The total number of iterations remained 2000, the system was tested every 10 iterations. Table 1 supplies the experimentally gathered results.

As may be seen from the results the Logical Error (LE), calculated by the algorithm described in paragraph *System Evaluation* of subsection 3.1, is high and exceeds 50%. The LE parameter was chosen as additional one to evaluate the system and comparing to the Mean Absolute Error (MAE) has less weight in system evaluation. The results in table 1 show that in experiments, where each clustering agent (CA) was able to proceed time series of three different durations ($q = 3$), MAE was less comparing to experimnerts with $q = 1$. The reason for such case may be that with $q = 3$ the clusterring community has less clustering agents, but nthe number each CA can process is larger. That supports the discovery of important associations in data. The minimal average value of MAE lies close to 2 periods and in individual cases the system vas able to succeed MAE less than 2 periods. Comparing to the results, acheved in paper [11] by Self-Organising Maps, the new presented system was able to lessen the Mean Absolute Error.

Table 1. Gathered results

Exp.	G	Masses after merging	CA load	MAE_{min}	MAE_{avg}	LE_{min}	LE_{avg}
1	0.0007	Remain 1	$q = 1$	2.33	2.72	64.8%	69.4%
2	0.0007	Remain 1	$q = 3$	1.88	2.24	57.6%	68.7%
3	0.0007	Summed	$q = 1$	2.57	2.91	54.0%	68.5%
4	0.0007	Summed	$q = 3$	1.74	2.35	61.9%	69.2%
5	0.00007	Remain 1	$q = 1$	2.21	2.59	65.7%	70.4%
6	0.00007	Remain 1	$q = 3$	1.75	2.07	59.7%	69.1%
7	0.00007	Summed	$q = 1$	2.37	2.84	59.3%	69.4%
8	0.00007	Summed	$q = 3$	1.83	2.16	58.4%	68.7%

In most cases the best results were gathered in experiments, where $q > 1$ and masses of the objects after merging remained equal to 1. This lessens the chance of appearance of a Black Hole that will dominate in the environment, and increases the chance that discovered clusters will contain important associations.

Analysing the gathered results it is possible to conclude that system was able to find associations in data and to apply a created model to forecast a transition point for a new product with a certain precision.

5 Conclusions

For the practitioners of management of the product life cycle the knowledge, which describes in which phase the product currently is and when the transition between phases will occur, is topical. Such knowledge, in particular, helps to select between the cyclic and non-cyclic production planning policy, as also to manage the assortment of products in stock.

In this paper, the task of forecasting the transition points between different phases of product life cycle is stated, and the structure of a MultiAgent Data Mining system, which helps to solve this task, is presented and described. The functional aspects of all agents in the system - Data Management Agent, Data Mining Agent and Decision Support Agent, are described. Experimentally gathered results show that the created MultiAgent Data Mining system has its potential and can process real demand data, create a model on the basis of historical data, forecast possible transition points and theoretically report an analysis of expenses for cyclic and non-cyclic planning policies.

For the future research it is necessary to examine the developed system on the data from different production fields, and, which is also important, to have a response from practitioners who will use these systems. The examination of other technologies for creating the knowledge base in the Data Mining Agent will be performed.

Another important moment is that the modest data volume that was used for practical experiments, is related to the fact, that it is necessary to have transition marks in historical data from experts and practitioners. The more products, the more complicated for human is to make all these marks - in practice the amount of marked data will always be limited. Due to that one more point of future research is creating a system with ability to extract valuable knowledge from a small set of marked data.

Acknowledgements

This work has been partially supported by the European Social Fund within the project "Support for the implementation of doctoral studies at Riga Technical University" (Agreement No. 2009/0144/1DP/1.1.2.1.2/09/IPIA/VIAA/005).

References

1. Campbell, G.M.: Cyclic assembly schedules for dynamic demands. IIE Transactions 28(8), 643–651 (1996)
2. Campbell, G.M., Mabert, V.A.: Cyclical schedules for capacitated lot sizing with dynamic demands. Management Science 37(4), 409–427 (1991)
3. Chakrabarti, S., Cox, E., Frank, E., et al.: Data Mining: Know It All. Morgan Kaufmann, San Francisco (2009)
4. Dunham, M.: Data Mining Introductory and Advanced Topics. Prentice-Hall, Englewood Cliffs (2003)
5. Gomez, J., Dasgupta, D., Nasraoui, O.: A new gravitational clustering algorithm. In: Proceedings of the SIAM International Conference on Data Mining (SDM), pp. 83–94. Society of Industrial and Applied Mathematics, Philadelphia (2003)
6. Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)
7. Hand, D.J., Mannila, H., Smyth, P.: Principles of Data Mining. The MIT Press, Cambridge (2001)
8. Kamath, N., Bhattacharya, S.: Lead time minimization of a multi-product, single-processor system: A comparison of cyclic policies. International Journal of Production Economics 106(1), 28–40 (2007)
9. Kotler, P., Armstrong, G.: Principles of Marketing, 11th edn. Prentice-Hall, Englewood Cliffs (2006)
10. Merkuryev, Y., Merkuryeva, G., Desmet, B., Jacquet-Lagreze, E.: Integrating analytical and simulation techniques in multi-echelon cyclic planning. In: Proceedings of the First Asia International Conference on Modelling and Simulation, pp. 460–464. IEEE Computer Society, Los Alamitos (2007)
11. Parshutin, S., Aleksejeva, L., Borisov, A.: Forecasting product life cycle phase transition points with modular neural networks based system. In: Perner, P. (ed.) Advances in Data Mining: Applications and Theoretical Aspects. LNCS (LNAI), vol. 5633, pp. 88–102. Springer, Heidelberg (2009)
12. Tan, P.-N., Steinbach, M., Kumar, V.: Introduction to Data Mining. Pearson Education, London (2006)
13. Wooldridge, M.: An Introduction to MultiAgent Systems. John Wiley & Sons, Chichester (2005)

Modeling Pricing Strategies Using Game Theory and Support Vector Machines

Cristián Bravo, Nicolás Figueroa, and Richard Weber

Department of Industrial Engineering
Universidad de Chile
{cbravo,nicolasf,rweber}@dii.uchile.cl

Abstract. Data Mining is a widely used discipline with methods that are heavily supported by statistical theory. Game theory, instead, develops models with solid economical foundations but with low applicability in companies so far. This work attempts to unify both approaches, presenting a model of price competition in the credit industry. Based on game theory and sustained by the robustness of Support Vector Machines to structurally estimate the model, it takes advantage from each approach to provide strong results and useful information. The model consists of a market-level game that determines the marginal cost, demand, and efficiency of the competitors. Demand is estimated using Support Vector Machines, allowing the inclusion of multiple variables and empowering standard economical estimation through the aggregation of client-level models. The model is being applied by one competitor, which created new business opportunities, such as the strategic chance to aggressively cut prices given the acquired market knowledge.

1 Introduction

Among the diverse decisions taken by companies, pricing is one of the most important. Decision makers do not only have a product's or service's price as a tool to affect demand, but also several marketing actions (e.g. mailings or call centers). The final consumer decision is thus influenced by market prices as well as by the stimuli he or she has been subject to.

The dynamics that these elements define can be modeled by game theory [8] which proposes results based on a solid economical background to understand the actions taken by agents when maximizing their benefit in non-cooperative environments. In companies, however, for more than twenty years data mining has been used to retrieve information from corporative databases, being a powerful tool to extract patterns of customer response that are not easily observable.

As of today, these two approaches (i.e. data mining and game theory) have been used to describe similar phenomena, but with limited interaction between each other. This work attempts to combine these approaches thus exploiting both the strong economical background used by game theory to model the relations that define competitive actions, as well as sophisticated data mining models to extract knowledge from the data companies accumulate.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 323–337, 2010.

In this model a customer-level, highly detailed demand estimation is introduced, built from Support Vector Machines that can handle a large number of variables from different sources, in contrast with common economics estimations. This demand is used to empower a market-level model based on game theory that details the situation the companies in the market are in, delivering an integrated picture of customers and competitors alike.

This work is structured as follows. Section 2 presents the game theoretic model used for this problem. In Section 4 the demand model is introduced, followed by technical details on Support Vector Machines (SVMs) which is the main technique utilized. The following section presents results obtained for a financial company. Finally, conclusions are drawn in Section 7. Possible future work is outlined in section 8.

2 Competition as a Game

Prior to the definition of game dynamics presented in this work, three definitions are necessary to fully understand the proposed model.

Definition 1. *A* strategy *s_j of a player j corresponds to a complete plan of actions, selected from a set of possible actions S_j that determines his or her behavior in any stage of the game. The player may, instead of using a fixed action s_j, define a probability distribution for the set S_j to determine his or her actions, this probability distribution p_j is called a* mixed strategy*.*

Definition 2. *Let p_j be the strategies for a set of J players in a given game. A* Nash Equilibrium *is a vector $p^* = (p_1^*, \ldots, p_J^*)$ containing the strategies of the players such that no player has incentives to change his or her particular strategy. If $S_j(p)$ is the payout for player j, then a Nash Equilibrium is such that*

$$S_j(p^*) = \max_{p_j} S(p_1^*, \ldots, p_j, \ldots, p_J^*) \quad \forall j \in \{1, \ldots, J\} . \tag{1}$$

Definition 3. *A* Perfect Sub-Game Equilibrium *is a refinement of the Nash Equilibrium concept where the state is an equilibrium to the game, and also is an equilibrium to all the sub-games that can be constructed from the original one.*

With these concepts at hand we can now define our game. Studies of competition dynamics are usually limited to a game theoretic framework where the players are the companies in the market under analysis. For this particular approach, Nash - Bertrand specification is useful, where players (companies) compete using prices as strategic variables, a reasonable assumption when quantity is flexible when compared to the different levels of demand [6].

In this context, the Nash - Bertrand equilibrium in a one-stage game has only one stable equilibrium: perfect competition, where each player fixes its price according to his marginal cost. However, Friedman [5] argued that when these games are played for a long (infinite) span of time, then every possible configuration of utilities that falls in an “acceptable” or “rational” range will be a perfect sub-game equilibrium.

The previous result, known as *folk theorem*, has an interesting interpretation: from the game theoretic point of view, companies that compete monthly for a fixed (or stable enough) set of customers fall in strategy configurations that will always result in a new equilibrium. If one agent modifies one of its decision variables then, under the assumption of rationality of the players, the new reached state will be a perfect sub-game equilibrium. Then it is useful to look for models that determine, given that one or more conditions are modified, *which* equilibrium will be obtained.

For this theorem to be applicable, the set of strategies must be non-empty. Rotemberg and Saloner[10] define a set of assumptions that are fulfilled in most markets, including the one in this application, that assure the existence of at least one equilibrium.

We will follow the steps of Sudhir *et al* [13] to define the model. Suppose there are J firms in the market with N_t customers in each period t; the firms must fix prices p_{jt}, marketing actions between L available (given by $x_{jt} \in \{0,1\}^L$) and face a cost vector c_{jt}. Under these conditions, Vilcassim, Kadiyali and Chintagunta [16] postulate that the marketing budget does not influence pricing, because it corresponds to a fixed cost. In this work it is considered that marketing strategies are determined *a priori*. This assumption seems realistic since usually marketing budgets and actions are planned at the beginning of each year whereas prices are fixed on a monthly basis.

Each period, the companies maximize the following expression:

$$\max_{p} \; N_t(p_{jt} - c_{jt})S_{jt}(p, x, \chi) \tag{2}$$

Where S_{jt} is the market share which has as inputs the price vector p, all observable marketing actions x, and the observable market heterogeneity given by $\chi \in \mathbb{R}^{I \times n}$ that is intrinsic to each company's customer database and is observable by the players using their respective databases. This assumption means that future utilities are infinitely discounted, being supported by the fact that even though the firm wishes to maximize its future benefits, managers usually prefer short-term goals, implying decisions such as price determination for a certain month, not long-term price fixing. The objective function to be maximized can be represented as a discounted sum, as done e.g. by Dubé and Manchanda [3]. The approach proposed in our paper simplifies the study and is centered on the determination of demand patterns.

To estimate the different cost functions, a matrix of cost factors C_t will be used as input along with a parameter vector λ_j that will be estimated from the firm's data: $c_{jt} = \lambda_j \cdot C_t + \varepsilon_t$ where ε_t is the error that occurs when using this method. The conditions of first order from (2) lead to the price definition for each firm:

$$p_{jt} = c_{jt} - \frac{S_{jt}}{\partial S_{jt}/\partial p_j} \tag{3}$$

The second term on the right hand side of (3), called *Bertrand margin*, must be adjusted by a parameter to allow deviations from the theoretical equilibrium.

The parameter that adjust equation (3) will be named κ_j, and corresponds to a numerical measure of how competitive the market is.

This parameter κ_j is of utmost importance, because it indicates the deviation respect to the equilibrium of each company. κ_j, with values between zero and one, indicates exactly how efficient is each company that is being modeled. Values close to one indicate efficiency (near-optimal behavior) and values close to zero represent the lack of it. It allows to identify the companies that are being inefficient and are subject to aggressive behavior from their competitors. To obtain the final model expression, it is necessary to replace the expression for costs and to include κ_j in (3):

$$p_{jt} = C_t \cdot \lambda_j - \kappa_j \frac{S_{jt}}{\partial S_{jt}/\partial p_j} + \varepsilon_t \tag{4}$$

The prior expression is identical to the one presented in previous works, since it represents the classical solution of Bertrand's competition with deviation assuming variable costs. The specification of the market share S_{jt} is where the present work differs from the usual economical modeling, because demand is modeled based on SVMs from disaggregated data at a customer level.

In general, aggregated data is used to model demand, according to the specification of Dubé *et al.* [2], but this approach does not take into account the real drivers for customer decisions, because the aggregated data usually corresponds to variables that indirectly interpret demand. In this work market share is modeled using the direct effects (prices), the indirect effects (marketing strategies) and the customer characteristics, expanding the spectrum used so far and attaining a buying propensity on a case by case basis.

This approach allows to handle a large number of variables in an efficient manner and also permits to construct a demand function with strong statistical support and generalization power, simultaneously providing a high level of detail.

3 Support Vector Machines

Support Vector Machines, the technique used to model demand, is based on the concepts of statistical learning created by Vapnik and Chervonenkis [14] around 1960. Pairs (x, y) are considered, in which an object $x \in X$ is represented by a set of attributes (the "input space" X) and $y \in \{-1, 1\}$ represents the class of the object. If a function $f : X \to Y$ exists that assigns each element in X to its correct class, the error incurred when approximating f can be measured in two ways. The empirical risk R_{emp} is the error accounted when approximating f from a sample set $M \subseteq X$ and the structural risk R is the error incurred in the whole set X. Modelers would like to minimize R, but only observe R_{emp}.

Statistical learning theory establishes bounds for the structural risk based on the empirical risk and a property of the family of functions used to determine classes, called "VC dimension". Defining the margin between a set and a hyperplane as the minimum distance between the hyperplane and the elements of the set, there is a unique hyperplane that maximizes the margin to each of the classes [15]. Furthermore, VC dimension is decreasing as the margin increases, so maximizing the margin and simultaneously minimizing the empirical risk is equivalent to minimizing the structural (real) risk of performing classification using the function f. These types of functions are called “classifiers based on hyperplanes” and possess functional forms given by:

$$f(x) = sgn\left[(w \cdot x) + b\right], \quad w \in R^n,\, b \in R,\, x \in X \tag{5}$$

A support vector machine is then a hyperplane built from a sample $M \subseteq X$ using an efficient algorithm to train it. The first interesting aspect has to do with the properties of the set X: to build the hyperplane one must possess an inner dot product in the space defined by X, and to maximize the margins is necessary that the elements in M are linearly separable. In order to by-pass this the “kernel trick” can be used, that starts by considering a function $\Phi : X \to \Phi(X)$ with $\Phi(X)$, the “feature space”, a Hilbert space [14] that possesses a defined dot product. In this particular space the separation of the classes can be done linearly, because for any finite - dimension space there is a higher dimensional space such that a non-linear separation in the input space becomes a linear separation in the feature space.

An interesting property of the previous definition is that, in order to construct an SVM, only the value of the dot product in the feature space is necessary, not the explicit form of $\Phi(x)$, so a function $K : X \times X \to \mathbb{R}$ can be used, where $K(x_1, x_2) = \Phi(x_1) \cdot \Phi(x_2)$. Function K is known as a kernel function, hence the “kernel trick”. Unfortunately, not all functions that behave as a dot product are kernels, there is an additional property that must be fulfilled, called the Mercer condition. In short, the condition states that the kernel function must be able to represent the dot product for every point in the space X. For a detailed explanation of these conditions, as well as examples of known kernel functions, the reader is referred to [1].

An SVM is then defined when solving the optimization problem that maximizes the margin between the classes, considering the following quadratic optimization problem:

$$\begin{aligned} \min_{w,b} \frac{1}{2}\left\|w\right\|^2 + \eta \sum_{j=1}^{M} \epsilon_j & \\ s.t. \quad y_j\left[K(w, x_j) + b\right] \geq 1 - \epsilon_j, &\quad j = 1, \ldots, m \\ \epsilon_j \geq 0, &\quad j = 1, \ldots, m \end{aligned} \tag{6}$$

The solution of this problem corresponds to hyperplane defined by the normal vector w and the distance to the origin given by b that maximizes margin and minimizes the error incurred. Slack variables ϵ_j account for the fact that not all problems are linearly separable, so the restrictions are relaxed by using these variables to consider an error in the classification. The objective function must account for both effects - minimum error and maximum margin - at the same time, which is done by including a relative weight η, that allows the modeler to balance both goals. The normal vector obtained by this problem is built from a weighted sum of a set of samples from the set M [12], subset named "support vectors" of the hyperplane f, hence the name of the technique. Since the solution of SVMs is composed of a subset of the original set, the solution of the SVM problem is sparse, which also gives numerical advantages over other types of models.

3.1 One-versus-All SVM

SVMs are by definition binary operators. However, some extensions have been developed that allow for multiclass classification, which are of interest for this paper. In particular, One-versus-All (OVA) [9] method will be used, that has been tested as one of the simpler, yet complete, approach to determine multiclass labels.

The main idea of the OVA approach is to train K SVMs, one for each class defined by object j's label $y_j \in \{1, ..., K\}$. In this case, the continuous output of each SVM is used (function (5) without the sign function), that represents the distance to hyperplanes with signs associated to which side of the hyperplane the example is in. The classifier is then given by $f(x) = (f_1(x), \ldots, f_K(x))$ and the class y is determined by the index of the maximum value in vector $f(x)$:

$$y_i = \begin{cases} 1 \; i = arg\max_k\{f_k\} \\ 0 \qquad \text{otherwise} \end{cases} \tag{7}$$

4 Customer-Level Demand and Aggregated Model

For the data mining twist to model (4), the market share S_{jt} must be defined. We propose to use SVMs for this task, which offers two main advantages:

1. Allows the use of atomic data: Econometric estimations usually employ aggregated data and general indicators as regressors. It is in the interest of both researchers and end-users to exploit the large quantity of data that exist in today's companies' databases. To model this phenomenon at a customer level is therefore of high relevance.
2. Allows to generalize demand: Data mining approaches model demand without the assumptions about the capacity to observe customers' characteristics. Instead, they are based on the patterns that each customer leaves about his or her behavior in the company databases. This allows to consider empiric demands based on the customers' actions (atomic model) and econometric models of the behavior of the firm (aggregated model), hence empowering both approaches.
3. Possesses methodological advantages: SVMs are a powerful mathematical model to approximate almost any type of phenomena. In particular, the problem (6) does not possess local minima, increasing the confidence of the solution.

The use of SVMs instead of Support Vector Regression (SVR) is justified because the demand must be estimated in a per-customer basis, since demand is not modeled by considering a continuous function of aggregated results, but by a set of different customers taking separated decisions.

To develop the final model, consider a database with customer attributes χ over T periods of time and a matrix of marketing actions $X \in \{0,1\}^{L \times T}$ directed to customers. Finally, one must assign labels indicating whether the customer chose the company, its competitors or neither (no-sale) for each one of the periods. Then, each customer can be represented by a vector of attributes consisting of their personal characteristics ($\chi_i \in \mathbb{R}^n$), the prices he or she observed at that particular time ($p = (p_1, \ldots, p_J) \in \mathbb{R}_+^J$) and the marketing strategies realized to the set of customers ($x_t \in \{0,1\}^L$): (x_i, p, χ_t). This data plus the labels associated to the firms ($y_i \in \{-1,1\}^{J+1}$) allow to train an SVM in OVA approach (section 3.1) in order to obtain, for each firm J, the predicted amount of customers that chose it in each period t and also the number of customers that do not choose any firm. There are then $J+1$ SVMs that model the tendency to buy (or not to buy) for each customer.

$$\begin{aligned} f_j(x_i, p, \chi_t) = sgn\left[k\left((w_j^x, w_j^p, w_j^\chi), (x_i, p, \chi_t)\right) + b_j\right], \\ j = \{1, \ldots, J\}, i = \{1, \ldots, N\} \end{aligned} \qquad (8)$$

The market share for a given period is the known market share from the previous period ($S_{(j,t-1)}$), adjusted by the new number of customers in period t minus the number of customers that are no longer in the captivity of the company at the end of period $t-1$ (e_{t-1}), plus the fraction of customers that are selected by the SVMs as customers of the company in period t:

$$\begin{aligned} S_{j,t}(p, x, \chi) &= S_{j,\widehat{(t-1)}} + \frac{\sum_{i \in N_t} f_j(p, x_i, \chi_t)}{N_t} \\ S_{j,\widehat{(t-1)}} &= \frac{S_{j,(t-1)} \cdot N_{t-1} - e_{t-1}}{N_t} \end{aligned} \qquad (9)$$

Equation 9 is a demand function that includes relevant customer-describing variables and also prices and information about strategic actions (e.g. the prices and marketing actions).

It is now necessary to estimate the derivative of (9) $\partial S_j(p, x, \chi)/\partial p_j$, which will be numerically estimated obtaining the number of customers that change their choice due to a price change. In particular, the secant method [4] will be used for the derivation, approximating it through moving the price in a small quantity Δp_j.

$$\begin{aligned} \frac{\partial S_j(p, x, \chi)}{\partial p_j} \approx \frac{1}{2}[&\frac{S_j(p^+, x, \chi) - S_j(p, x, \chi)}{\Delta p_j} \\ &+ \frac{S_j(p, x, \chi) - S_j(p^-, x, \chi)}{\Delta p_j}] \end{aligned}$$

$$
\begin{aligned}
&= \frac{S_j(p^+, x, \chi) - S_j(p^-, x, \chi)}{2\Delta p_j} \\
&= \frac{\sum_{i=1}^{N_t} [f(p^+, x_i, \chi_t) - f(p^-, x_i, \chi_t)]}{2\Delta p_j} \qquad (10) \\
\text{with } p^+ &= (p_1, \ldots, p_j + \Delta p_j, \ldots, p_J), \\
p^- &= (p_1, \ldots, p_j - \Delta p_j, \ldots, p_J)
\end{aligned}
$$

Now, replacing (9) and (10) in (4) the final model is obtained:

$$
p_{jt} = \lambda_j \cdot C_t + \kappa_j \, \frac{\hat{S}_{j,(t-1)} + \frac{\sum_{i=1}^{N_t} f(p, x_i, \chi_t)}{N}}{\frac{\sum_{i=1}^{N_t} [f(p^+, x_i, \chi_t) - f(p^-, x_i, \chi_t)]}{2\Delta p_j}} + \epsilon_j \qquad (11)
$$

An interesting feature of (11) is, even though the full expression is non-linear, the final estimation is done linearly. To train this model, we propose the following three-step procedure:

1. Construction of integrated database: We need three different kinds of data, an internal database to construct the matrix χ with customer data, information about the competitors, for example whether they have performed a commercial or other visible activity on the customers, stored in matrix X and finally the observed prices must be collected to construct the price matrix P. These prices are usually available to public.
2. Train SVMs on the integrated database: It is necessary to determine the different expected market shares for the competing companies. Expression (9) allows a customer to choose more than one company at the same time, an

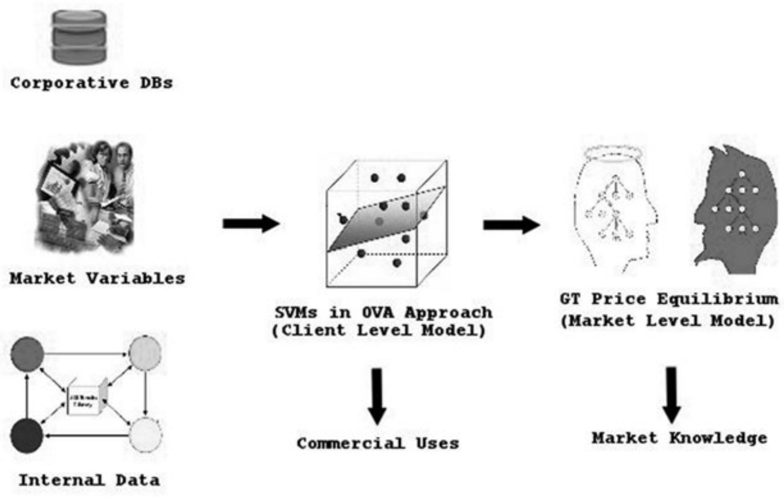


Fig. 1. Model diagram

event that actually can happen in most markets. This case can, however, be extracted from the database or considered as participation anyway, although this is a debatable step.

3. Estimate the parameters in (11): To estimate the resulting linear regression, cost regressors are needed, and can be found through global indicators. In this case, indicators such as the Producer Price Index (PPI) from Chile's National Institute of Statistics and other widely available global indicators were used to model the marginal cost for each company.

To adjust the final model and considering that the resulting problem is linear, least squares or a maximum likelihood method such as the generalized method of moments can be used. The input data are the monthly cost regressors, the observed shares and the estimation of the derivative from equation (10) for each period of time measured. The schematic application of the proposed model is illustrated in Figure 1.

5 Benchmark Model

In order to check the performance of the proposed approached, an artificial neural network (ANN) [11] will be used as a benchmark of the performance when estimating demands. Neural networks are composed of neurons (nodes) that are organized in layers. Each neuron receives as input all the outputs from the previous layer, and applies a specific weight and a transfer function to this input, to then pass this result to the neurons in the next layer. The first layer (input layer) only consists of weights and each neuron is associated to one input variable of the dataset. The final layers is called output layer, and presents the final result in any specified format, while the layers between the input and output layers are called hidden layers. The configuration of the neural network consists of:

- Hidden Layers: The number of hidden layers, and the number of neurons in each hidden layer must be decided. It has been stated [7] that only one hidden layer is necessary to approximate any bounded function, while two are necessary to approximate any unbounded one. In this experiment, since a probability will be estimated, only one hidden layer will be used.
- Output Layer: The output layer must have an output function that presents the results in a logical format for the problem being modeled. In this case, a softmax function must be used, given by:

$$p_i(x) = \frac{exp(\beta_j \cdot x)}{\sum_{j'} exp(\beta_{j'} \cdot x)} \tag{12}$$

  with β_i a vector of output parameters associated to each class j, and x the vector of variables.
- Transfer functions: The functions that determine the input of each layer must be decided. In this case, linear functions were used, representing the final output as a multinomial logistic function, typically used for this type of problems.

– Training parameters: Depending on the software used to train a neural network, several parameters must be defined to determine convergence. The number of epochs (times the neuron is presented with the data) is one common attribute.

In order to obtain the number of epochs and the number of neurons of the hidden layer, a grid-search process was conducted, as described in section 6.

To perform a comparison on the performance to approximate demand, the results of this ANN are aggregated by simply adding the probability that each customer chooses any of the companies, adjusted in the same way as equation 9. This gives the expectation of the demand for the period, as desired.

6 Experimental Results

To apply this model in a real-life situation, data from a well-known local company was available to the authors. The company offers a complete line of financial services, and between them are loans that are directly discounted from the customer's income. This database has some advantages that make it perfect for this particular problem:

– The products placed by any competitor are known: The company in study has access to the other companies that also place products to a given customer, because the discount of the loan's installments is done through the company. This allows knowing in real-time when the customer has chosen the company's competitors and the company in study, fulfilling the most restrictive assumption in the model.
– The market is highly concentrated: 93% of total loans are concentrated in three companies, with the rest offering the remaining 7%, allowing them to behave as an oligopoly and possessing some market influence. The companies with market share of 7% will be referred to as one single company (company O) for simplicity of study.

The market is then formed by four companies (E, A, H and O) that struggle to acquire N customers, each one of them characterized by the variables described in section 2. In particular, 80 variables were studied, characterizing 100,000 different customers over 18 months. The variables came from different sources:

– Internal Databases: possessing demographic data, the income for the customers and the shares for the companies.
– External Databases: Information about prices and cost regressors, which came from Chilean Central Bank, the National Institute of Statistics (INE) and the organism that supervises the companies in the market, called Superintendence of Social Security (SUSESO).
– Generated Variables: Some indicators were built from global income, debt, specific per-company debt and so on, in order to improve the results of the models and to attempt to discover new relationships between the variables.

The variables were selected utilizing a complete study that maximizes the contact of the modeler with the variables and so the extracted knowledge. This process begins by eliminating variables highly concentrated in one single value or with a high rate of missing values, then it continues with a process of univariate feature selection, where variables that possess no univariate discriminating capacity are eliminated. The capacity is obtained from simple and widely known χ^2 and K-S tests, supposing that independence from the objective variable in a univariate way also implies independence from that objective in a multivariate environment. Finally, over-adjusted classification trees are built under the hypothesis that if the variable does not appear in any levels of the tree then it would likely not appear in a different multivariate model. This procedure has been tested previously by our team with good results.

Finally, a set of 20 discriminating variables was achieved, being the input for the data mining models. The available database consists of approximately 100,000 registers and is highly unbalanced, considering that company E has a market share of 50%.

In order to perform the experiments, a search must be performed for optimal parameter setting. 20% of the database was reserved for such task and to ensure the model was able to cover all classes two precautions were applied:

- The samples were artificially balanced using an adjustment parameter that grants a value of one to all the elements that are in the set associated with the class with less cases and a value of $\frac{\text{cases-minor-class}}{\text{cases-mayor-class}}$ to the elements of the class with more registers associated. This is done for each SVM, considering they are in an OVA scheme.
- An *ad-hoc* error function was used to measure performance of the particular parameter configuration, that balances the errors in each class (e_m). This function multiplies the errors for each class, so only solutions that represent all classes are considered. Considering $e_c, c \in \{E, A, H, O, NB\}$ the errors per class, the error function is given by (13). The reader should note that the "No-Buy" class is included ("NB") that consists of all the customers that choose not to buy in a particular period in any company.

$$e_m = \prod_{c \in \{\text{E,A,H,O,NB}\}} e_c \tag{13}$$

- Finally the error is averaged over 3-sample cross-validation, with the error for each parameter set being the average of the errors from equation (13).

With these steps the optimal parameter setting was found and used to train the model in the remaining 80% of the sample.

The elements of this sample were divided in five different subsets to perform cross-validation once again, to reduce the sample error, also keeping additionally 20% off the training for testing. Table 1 displays the results for each company, consisting on the per-class error, which is close to 10% in average, this being a satisfactory measure. The benchmark model performs well below this index, with errors around 15%. Since SVMs allow to fine tune each class performance,

Table 1. Class Error for each company

Class	Class Error (SVM)	Class Error (ANN)
E	21,53%	22,75%
A	4,37%	13,78%
H	1,89%	10,72%
O	1,91%	8,98%
No Buy	21,61%	43,78%

Table 2. Accuracy of aggregated results

Company	SVM Error	ANN Error
E	16.40% ± 11.50%	43.22% ± 19.38%
A	19.60% ± 5.86%	30.38% ± 17.77%
H	29.00% ± 9.80%	23.99% ± 15.16%
O	19.90% ± 8.38%	26.10% ± 15.53%

the model offers more chances to improve the result obtained, as is reflected in this experiment.

Table 1 showed the results on an individual customer level, but in this work we require the estimation of aggregated demand, which is not at all common in data mining models, that usually are at an atomic level. The results were aggregated using equations (9) and (10), with results close to an 80% of accuracy (Table 2), which also is highly satisfactory and supports the use of data mining models for aggregated estimations. The benchmark model once again is outperformed by the SVM results, as is expected, since the results from the client level model should be somewhat transferred to the estimation of demand functions, also, the standard deviation of the model is higher, which indicates that the SVM is capable of capturing a wider range of different patterns.

The final step is to estimate the game theoretic model from equation (11). The cost factors used consider the cost of life (Consumer Price Index, IPC), the maximum interest rates allowed, and price indexes to producers, salary indexes and others from the sources previously indicated. The dataset consists on weekly data and includes the previous variables, plus the prices (target) charged by each company in that week, and the estimated demands divided by the derivative. The regression was run using the software package SPSS and feature selection was performed using backward and forward selection, conciliating both approaches by keeping the feature combination that performed best in the sample. The results are highly satisfying (table 3) with over 95% of accuracy in average and low standard error, which once again supports the use of this kind of models to predict price changes.

The efficiency coefficient κ_j is of particular interest, because it represents how efficient company j is when fixing its price, a well-known result in game theory.

Table 3. Results for regression on prices

Company	R	**Adjusted** R^2	**Std. Error**	κ_j
E	0.973	0.796	0.016	0.684
A	0.919	0.758	0.054	0.757
H	0.961	0.903	0.031	0.096
O	0.994	0.981	0.02	0

Companies with higher market shares are more efficient, establishing that the most important drivers of price changes are changes in demand and competition. Companies that are less efficient, on the other hand, present smaller values, which indicate that their main drivers to fix prices are their observed costs and their lack of interest (or capacity) to take demand into account. This result is really interesting because it establishes a quantitative measure of the different companies' market position in a given market and goes beyond the results each single approach - data mining and game theory - could provide.

7 Conclusions

The model introduced in this work provides a novel tool to find market equilibria and to determine the expected market share when modifying strategic variables. Moreover, demand is modeled in terms of directly measurable variables such as price, and in terms of indirect variables such as marketing strategies that the company employs and in terms of the customers' characteristics. This provides a more profound knowledge regarding the customers' attitude towards the different companies. The model offers an integrated view of the elements that define the respective market, integrating the available knowledge, providing a major advantage over the use of a single technique.

In general, the use of models based on successions of games represents an effective alternative to measure the effects of changes in the market's competition conditions. This way, a theoretical limitation (the existence of infinite market equilibriums) is transformed into a useful tool, granting the possibility to determine this new equilibrium in terms of modeling past behavior.

Currently, the so-called “indirect” effects consume a great deal of hours and resources spent in a company, so they cannot be neglected. The connection with data mining allows overcoming this challenge, explaining complex phenomena by obtaining the statistical patterns present in the large quantity of data that companies are storing. This way the reasons that drive a person to prefer a determined company can be studied in detail.

The main limitation of the presented model is the data that needs to be collected, in particular the data referred to competitors' product placement. A workaround to this limitation consists in collecting this data through surveying customers of the company that produces the study's database.

The use of data mining models to estimate aggregated demand is another interesting contribution of this paper. A simple methodology is introduced to aggregate the obtained atomic results that gives very good results. The main reason a researcher would like to utilize this type of demand model is that data mining allows an efficient handling of large quantity of variables, so it is useful when compared to classical demand estimation models that cannot do so.

The model gives useful and applicable results that can be utilized in day-to-day decisions. In particular, the work from this paper was used to design a campaign to acquire competitors' customers, which had a high positive response rate and allowed to increase the market share of company E, a fact that gives even more credibility to the application of such models in companies.

Considering all these elements, the combination of data mining with game theory provides an interesting research field that has received a lot of attention from the community in recent years, and from which a great number of new models are expected. Future studies will generate promising results in all aspects where both a large number of data and interaction between agents are present. An integrated vision that takes into account, at the same time, consumers and companies has been introduced in this paper. This integrated vision allows interpreting the relationships of all the participants and giving a full spectrum of the market.

8 Future Work

Two separate lines of work have been developed from this paper. The first consists of improving the presented model using analytical techniques to avoid the numerical estimations and to improve the model results. The second one, still under development, is to use the techniques here presented to improved credit scoring models, modeling the loan granting process as a game and then applying credit scoring techniques.

Acknowledgments

The first author would like to acknowledge CONICYT for the grants that finance this research, and to C. Mora for her aid in editing this paper. Also, support from the Millennium Science Institute on Complex Engineering Systems (ICM: P-05-004-F, CONICYT: FBO16) (www.sistemasdeingenieria.cl) and from the Ph.D. in Engineering Systems (for the first author) is greatly acknowledged.

References

1. Cristiannini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Methods. Cambridge University Press, Cambridge (2003)
2. Dubé, J., Chintagunta, P., Bronnenberg, B., Goettler, R., Petrin, A., Seetharaman, P., Thomadsen, R., Zhao, Y.: Structural applications of the discrete choice model. Marketing Letters 13(3), 207–220 (2002)

3. Dubé, J., Manchanda, P.: Differences in dynamic brand competition across markets: An empirical analysis. Marketing Science 24(1), 81–95 (2005)
4. Faires, J.D., Burden, R.: Numerical Analysis. Thomson Brooks/Cole (1986)
5. Friedman, J.: A non-cooperative equilibrium for supergames. Review of Economic Studies 38, 112 (1971)
6. Fudenberg, D., Tirole, J.: Game Theory. MIT Press, Cambridge (1991)
7. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2(5), 359–366 (1989)
8. Nash, J.: Non-cooperative games. Annals of Mathematics 54, 286–295 (1951)
9. Rifkin, R., Klautau, A.: In defense of one-vs-all classification. The Journal of Machine Learning Research 5, 101–141 (2004)
10. Rotemberg, I., Saloner, G.: A super-game theoretic model of business cycles and price wars during booms. American Economic Review 76(3), 390–407 (1986)
11. Ruck, D.W., Rogers, S.K., Kabrisky, M., Oxley, M.E., Suter, B.W.: The multilayer perceptron as an approximation to a bayes optimal discriminant function. IEEE Transactions on Neural Networks 1(4), 296–298 (1990)
12. Schlkopf, B.: Statistical learning and kernel methods. Tech. Rep. Msr-Tr-23-2000, Microsoft Research (2000)
13. Sudhir, K., Chintagunta, P., Kadiyali, V.: Time varying competition. Marketing Science 24(1), 96–110 (2005)
14. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons Inc., Chichester (1998)
15. Vapnik, V., Lerner, A.: Pattern recognition using generalized portrait method. Automatization and Remote Control 24, 774–780 (1963)
16. Vilcassim, N., Kadiyali, V., Chintagunta, P.: Investigating dynamic multifirm market interaction in price and advertising. Management Science 45(4), 499–518 (1999)

Determination of the Fault Quality Variables of a Multivariate Process Using Independent Component Analysis and Support Vector Machine

Yuehjen E. Shao[1], Chi-Jie Lu[2], and Yu-Chiun Wang[3]

[1] Department of Statistics and Information Science, Fu Jen Catholic University,
Hsinchuang, Taipei County 242, Taiwan, R.O.C
stat1003@mail.fju.edu.tw
[2] Department of Industrial Engineering and Management, Ching Yun University,
Jung-Li 320, Taoyuan, Taiwan, R.O.C
jerrylu@cyu.edu.tw
[3] Graduate Institute of Applied Statistics, Fu Jen Catholic University,
Hsinchuang, Taipei County 242, Taiwan, R.O.C
yuhchunwang001@yahoo.com.tw

Abstract. The multivariate statistical process control (MSPC) chart plays an important role in monitoring a multivariate process. Once a process disturbance has occurred, the MSPC out-of-control signal would be triggered. The process personnel then begin to search for the root causes of a disturbance in order to take remedial action to compensate for the effects of the disturbance. However, the use of MSPC chart encounters a difficulty in practice. This difficult issue involves which quality variable or which set of the quality variables is responsible for the generation of the out-of-control signal. This determination is not straightforward, and it usually confused the process personnel. This study proposes a hybrid approach which is composed of independent component analysis (ICA) and support vector machine (SVM) to determine the fault quality variables when a step-change disturbance existed in a process. The well-known Hotelling T^2 control chart is employed to monitor the multivariate process. The proposed hybrid ICA-SVM scheme first uses ICA to the Hotelling T^2 statistics generating independent components (ICs). The hidden useful information of the fault quality variables could be discovered in these ICs. The ICs are then used as the input variables of the SVM for building the classification model. The performance of various process designs is investigated and compared with the typical classification method.

Keywords: Multivariate statistical process control chart, Independent component analysis, Support vector machine, Fault quality variable.

1 Introduction

Multivariate statistical process control (MSPC) chart is one of the most important techniques to monitor a multivariate process. The generation of the out-of-control signal indicates that the disturbance has been introduced in the underlying process.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 338–349, 2010.

When the MSPC chart triggers a signal, the process personnel should remove the root causes of the disturbance and then bring the process back in a state of statistical control. The remove of the disturbance would mainly depend on the correct determination of the fault quality variables. Once the correct determination has been made, the corresponding remedial actions can be properly taken to compensate for the effects of the disturbance. As a consequence, the process improvement can be significantly achieved.

However, the use of MSPC chart often encounters a problem in which the interpretation of the signal is confusing. Although the generation of signal implies that the underlying process is out-of-control, the contributors of the fault quality variables to this signal are difficult to determine. Typically, there are 2^p-1 possible set of fault quality variable in an out-of-control process with P quality characteristics or variables. For example, there should be 31 possible set of fault quality variables in a process with 5 quality characteristics. When a signal is triggered, it is not straightforward to determine which one of the 31 possible combinations is responsible for the generation of the signal.

Runger, Alt, and Montgomery [1] (RAM) addressed a solution to overcome this problem. The RAM method computes an approximate chi-square statistic to determine which of the monitored quality variables causes the MSPC signal. However, the RAM method has some limitations in certain situations [2]. The RAM approach may not be able to offer correct identification rate (CIR) when the small magnitude of process disturbance existed in a process. Some classification techniques are therefore developed to overcome the drawback of the RAM method [2-3]. Shao [2] proposed the use of artificial neural networks (ANN) and support vector machine (SVM) approaches to determine the fault quality variables in the case of process mean shifts, and Cheng and Cheng [3] also used the ANN and SVM techniques to identify the fault quality variables in the case of process variance shifts.

Different from the typical “one-step” classifiers’ approach [2-3], this study is concerned with the development of a hybrid or a two-step approach. The essential concept of the proposed hybrid approach is that the distinguishability information of the fault quality variable may be embedded in the monitor statistics, for example, the Hotelling T^2 statistics in the Hotelling T^2 control chart. We may enhance the CIR if we decompose the monitor statistics and input the decomposed factors to the classifiers. Because of the common applications in practice [2, 4-7], this study considers the case of process mean shifts in a multivariate process with the use of a Hotelling T^2 control chart. In addition, since the independent component analysis (ICA) has been reported to have the capability of distinguishability [8-12], this study uses the ICA as the first step technique to extract the independent components (ICs) from Hotelling T^2 statistics. The hidden useful information of the fault quality variables could be embedded in these ICs. In the second step of classification, those ICs are then used as the input variables of the classifiers. This study considers SVM as classifiers, and the main reason is its great potential and superior performance in practical applications [13-17].

The structure of this study is organized as follows. Section 2 addresses the methodologies used. Section 3 constructs the appropriate models for determining the fault quality variables when the process mean shifts are introduced in a multivariate process. In this section, the experimental example is addressed and the simulation results are also discussed. The final section presents the research findings and draws the conclusion to complete this study.

2 Methodology

This study uses ICA to enhance the classification capability of SVM. There are some applications of using ICA in process monitoring. Kano *et al.* [18] have successfully demonstrated the idea of process monitoring based on the observation of ICs instead of the original measurements. In their work, a set of devised statistical process control charts have been developed effectively for each IC. Lee *et al.* [19-20] investigated the utilization of kernel density estimation to define the control limits of ICs that do not satisfy Gaussian distribution. In order to monitor the batch processes which combine independent component analysis and kernel estimation, Lee *et al.* [21] extended their original method to multi-way ICA. Xia and Howell [22] developed a spectral ICA approach to transform the process measurements from the time domain to the frequency domain and to identify major oscillations.

2.1 Independent Component Analysis

Let $\mathbf{X} = [\mathrm{x}_1, \mathrm{x}_2, \Lambda\ , \mathrm{x}_m]^T$ be an matrix of size $m \times n$, $m \le n$, consisting of observed mixture signals x_i of size $1 \times n$, $i = 1, 2, \Lambda\ , m$. In the basic ICA model, the matrix **X** can be modeled as

$$\mathbf{X} = \mathbf{AS} = \sum_{i=1}^{m} \mathbf{a}_i \mathbf{s}_i \, , \tag{1}$$

where $\mathbf{a}_i$ is the i^{th} column of the $m \times m$ unknown mixing matrix $\mathbf{A}$; $\mathbf{s}_i$ is the i^{th} row of the $m \times n$ source matrix $\mathbf{S}$. The vectors $\mathbf{s}_i$ are latent source signals that cannot be directly observed from the observed mixture signals x_i. The ICA model aims at finding an $m \times m$ de-mixing matrix $\mathbf{W}$ such that

$$\mathbf{Y} = [\mathbf{y}_i] = \mathbf{WX} = [\mathbf{w}_i \mathbf{X}], \tag{2}$$

where $\mathbf{y}_i$ is the i^{th} row of the matrix $\mathbf{Y}$, $i = 1, 2, \Lambda\ , m$. The vectors $\mathbf{y}_i$ must be as statistically independent as possible, and are called independent components (ICs). ICs are used to estimate the latent source signals $\mathbf{s}_i$. The vector $\mathbf{w}_i$ in equation (2) is the i^{th} row of the de-mixing matrix $\mathbf{W}$, $i = 1, 2, \Lambda\ , m$. It is used to filter the observed signals $\mathbf{X}$ to generate the corresponding independent component $\mathbf{y}_i$, *i.e.*, $\mathrm{y}_i = \mathbf{w}_i \mathbf{X}$, $i = 1, 2, \Lambda\ , m$.

The ICA modeling is formulated as an optimization problem by setting up the measure of the independence of ICs as an objective function and using some optimization techniques for solving the de-mixing matrix $\mathbf{W}$ [23-24] The ICs with non-Gaussian distributions imply the statistical independence [23], and the non-Gaussianity of the ICs can be measured by the negentropy [25]:

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y}), \tag{3}$$

where $\mathbf{y}_{gauss}$ is a Gaussian random vector having the same covariance matrix as $\mathbf{y}$. H is the entropy of a random vector $\mathbf{y}$ with density $p(\mathbf{y})$ defined as $H(\mathbf{y}) = -\int p(\mathbf{y}) \log p(\mathbf{y}) d\mathbf{y}$.

The negentropy is always non-negative and is zero if and only if $\mathbf{y}$ has a Gaussian distribution. Since the problem in using negentropy is computationally very difficult, an approximation of negentropy is proposed [25] as follows:

$$J(y) \approx [E\{G(y)\} - E\{G(v)\}]^2 \tag{4}$$

where v is a Gaussian variable of zero mean and unit variance, and y is a random variable with zero mean and unit variance. G is a nonquadratic function, and is given by $G(y) = \log(\cosh y)$ in this study. The *FastICA* algorithm proposed by [23] is adopted in this paper to solve for the de-mixing matrix $\mathbf{W}$. Two preprocessing steps are common in the ICA modeling, centering and whitening [23]. Firstly, the input matrix $\mathbf{X}$ is centered by subtracting the row means of the input matrix, *i.e.*, $\mathbf{x}_i \leftarrow (\mathbf{x}_i - E(\mathbf{x}_i))$. The matrix $\mathbf{X}$ with zero mean is then passed through the whitening matrix $\mathbf{V}$ to remove the second order statistic of the input matrix, *i.e.*, $\mathbf{Z} = \mathbf{VX}$. The whitening matrix $\mathbf{V}$ is twice the inverse square root of the covariance matrix of the input matrix, *i.e.*, $\mathbf{V} = 2(C_{\mathbf{X}}))^{-(1/2)}$, where $C_{\mathbf{X}} = E(\mathbf{xx}^T)$ is the covariance matrix of $\mathbf{X}$. The rows of the whitened input matrix $\mathbf{Z}$, denoted by $\mathbf{z}$, are uncorrelated and have unit variance, *i.e.*, $E(\mathbf{zz}^T) = \mathbf{I}$. In this study, it is assumed that the training and testing process datasets are centered and whitened.

2.2 Support Vector Machine

The use of SVM algorithm can be described as follows. Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, $\mathbf{x}_i \in R^d$, $y_i \in \{-1, 1\}$ be the training set with input vectors and labels. Here, *N* is the number of sample observations and *d* is the dimension of each observation, y_i is known target. The algorithm is to seek the hyperplane $\mathbf{w} \cdot \mathbf{x}_i + b = 0$, where **w** is the vector of hyperplane and *b* is a bias term, to separate the data from two classes with maximal margin width $2/\|\mathbf{w}\|^2$, and the all points under the boundary is named support vector. In order to obtain the optimal hyperplane, the SVM was used to solve the following optimization problem [26]:

$$\text{Min } \Phi(\mathbf{x}) = \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, i = 1, 2, \ldots, N \tag{5}$$

It is difficult to solve Eq(5), and we need to transform the optimization problem to be dual problem by Lagrange method. The value of α in the Lagrange method must be non-negative real coefficients. The Eq(5) is transformed into the following constrained form [26],

$$\text{Max } \Phi(\mathbf{w}, b, \xi, \alpha, \beta) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1, j=1}^{N} \alpha_i \alpha_j y_i y_j \mathbf{x}_i{}^T \mathbf{x}_j$$

$$\text{s.t. } \sum_{j=1}^{N} \alpha_j y_j = 0$$

$$0 \le \alpha_i \le C, i = 1, 2, ..., N \tag{6}$$

In Eq(6), C is the penalty factor and determines the degree of penalty assigned to an error. It can be viewed as a tuning parameter which can be used to control the trade-off between maximizing the margin and the classification error.

In general, it could not find the linear separate hyperplane in all application data. For problems that can not be linearly separated in the input space, the SVM uses the kernel method to transform the original input space into a high dimensional feature space where an optimal linear separating hyperplane can be found. The common kernel function are linear, polynomial, radial basis function (RBF) and sigmoid. In this study, we used multi-class SVM method proposed by [27].

3 The Proposed Approach and the Example

3.1 The ICA-SVM Scheme

This study integrates ICA and SVM for determining the fault quality variables of an out-of-control multivariate process. In the training phase, the aim of the proposed scheme is to obtain the proper parameter setting for the SVM model. Since the RBF kernel function is adopted in this study, the performance of SVM is primarily affected by the setting of parameters of the parameters, C and γ. There are no general rules for the choice of those two parameters. This study uses the grid search proposed by [28] for those two parameters setting. The trained SVM model with proper parameter setting is preserved and employed in the testing phase.

The proposed model first collect two sets of Hotelling T^2 statistics from the out-of-control process. The ICA model is used to generate the two estimated ICs from the observed Hotelling T^2 statistics. Then, the proposed scheme considers those two ICs and 3 averaged quality variables, 4 averaged quality variables, and 5 averaged quality variables as inputs for SVM in the case of processes with 3 quality characteristics, 4 quality characteristics, and 5 quality characteristics, respectively.

3.2 The Simulated Example

In order to demonstrate the use of our proposed approach, this study considers a simulated example. This study applies Hotelling T^2 control chart to monitoring a multivariate process with 3, 4, and 5 quality characteristics, respectively. For each type of processes, this study considers the types of correlation, ρ, between any two qualtiy variables as no correlation (i.e., $\rho = 0$), moderate correlation (i.e., $\rho = 0.6$), and high correlation (i.e., $\rho = 0.9$). Now, consider a case of out-of-control process with 3 quality characteristics. Since the process has 3 quality characteristics (i.e., P=3), the possible sets of fault quality variables would be 2^P-1=7. In our study, we use the following notations: (1,0,0), (0,1,0), (0,0,1), (1,1,0), (1,0,1), (0,1,1), and (1,1,1) to represent the 7 possible sets, in which "0" stands for the "in-control" state and "1" stands for the "out-of-control" state. The meaning of (1,1,0) stands for the first and second quality variables (i.e., X_1 and X_2) are fault while the third quality variable (i.e., X_3) is not fault. In our simulation, we assume that the in control process follows a normal distribution with mean of 0 and standard deviation of 1. The out-of-control process has a mean shift of 1 standard deivation, that is, the out-of-coantrol process follows a normal distribution with mean of 1 and standard deivation of 1. This study also considers a common used sample size of 5, and the sample averages ($\overline{X_i}$, i = 1, 2, and 3) are used to calculate the Hotelling T^2 statistics. The Hotelling T^2 statistics are computed as follows.

$$T^2 = n(\overline{X} - \overline{\overline{X}})' \; S^{-1}(\overline{X} - \overline{\overline{X}}), \tag{7}$$

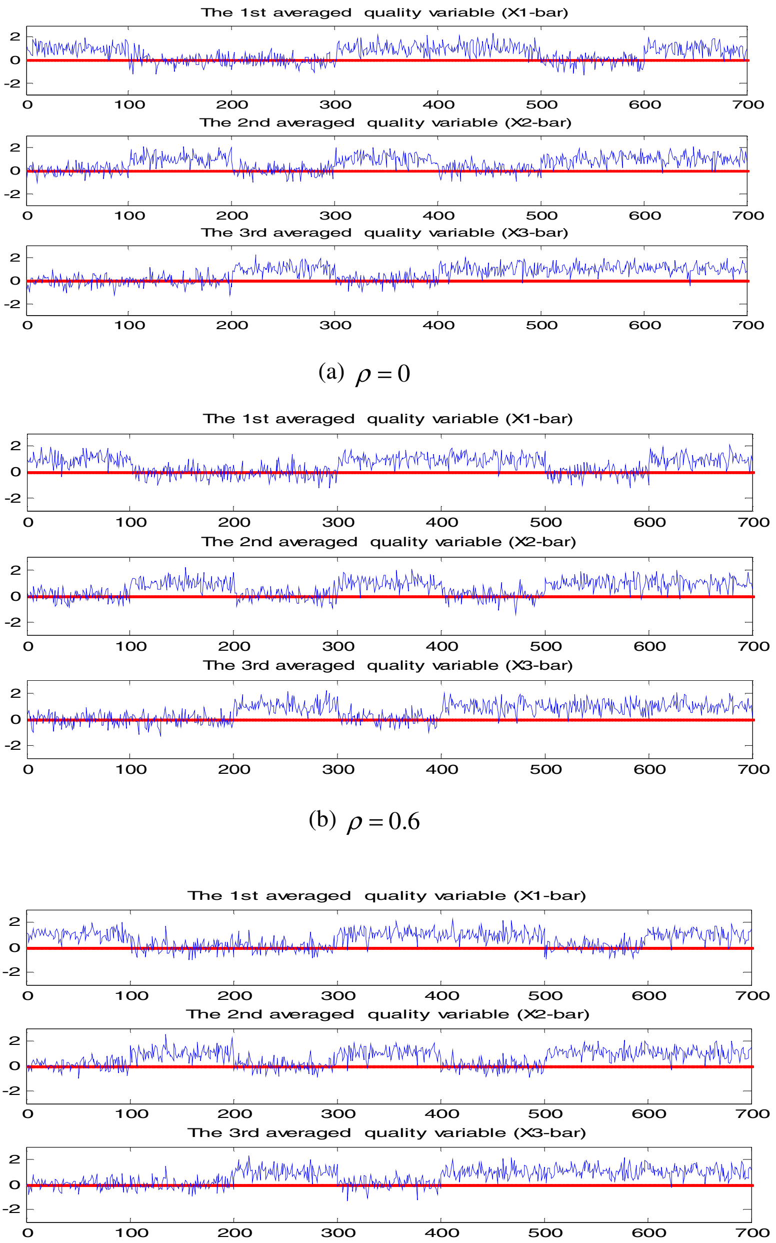
where

n:	the sample size,
$\overline{X}$:	the mean vector at the time t,
$\overline{\overline{X}}$:	the grand mean vector of the quality characteristics, and
S^{-1}:	the inverse of variance and covariance matrix.

Also, this sttudy generates 100 data sets of observations (each of sample size 5) for every possible sets. Since there are 7 possible sets of fault quality variables in the case of P=3, we have 700 data sets in a simulation run. Those 700 data sets are initially used to be the training data. This study generate another 700 data sets for the purpose of the testing. Figure 1 displays the 700 data sets of $\overline{X_1}$, $\overline{X_2}$, and $\overline{X_3}$ in the cases of $\rho = 0$, $\rho = 0.6$, and $\rho = 0.9$, respectively. In the first step of classification, we also use the data set of out-of-control Hotelling T^2 statistics which is shown in Figure 2. Figure 3 displays the two ICs which is generated by using ICA technique.

3.3 The Results

In the case of P=3 in a multivariate process, the typical approach uses four variables, $\overline{X_1}$, $\overline{X_2}$, $\overline{X_3}$, and the Hotelling T^2 statistics as inputs for SVM. When the proposed approach is employed, the five variables, $\overline{X_1}$, $\overline{X_2}$, $\overline{X_3}$, and the two ICs, are considered as the inputs for SVM. Table 1 shows the experimental results of the testing phase for the typical and the proposed appraoches. Considering the case of ρ =0 and the shift type of (1,0,0), the typical approach (i.e., T^2+SVM) shows that the CIR is 73.5% and the CIR is 81.1% for the proposed approach. We can apparently notice that the proposed approach outperforms the typical approach. In the case

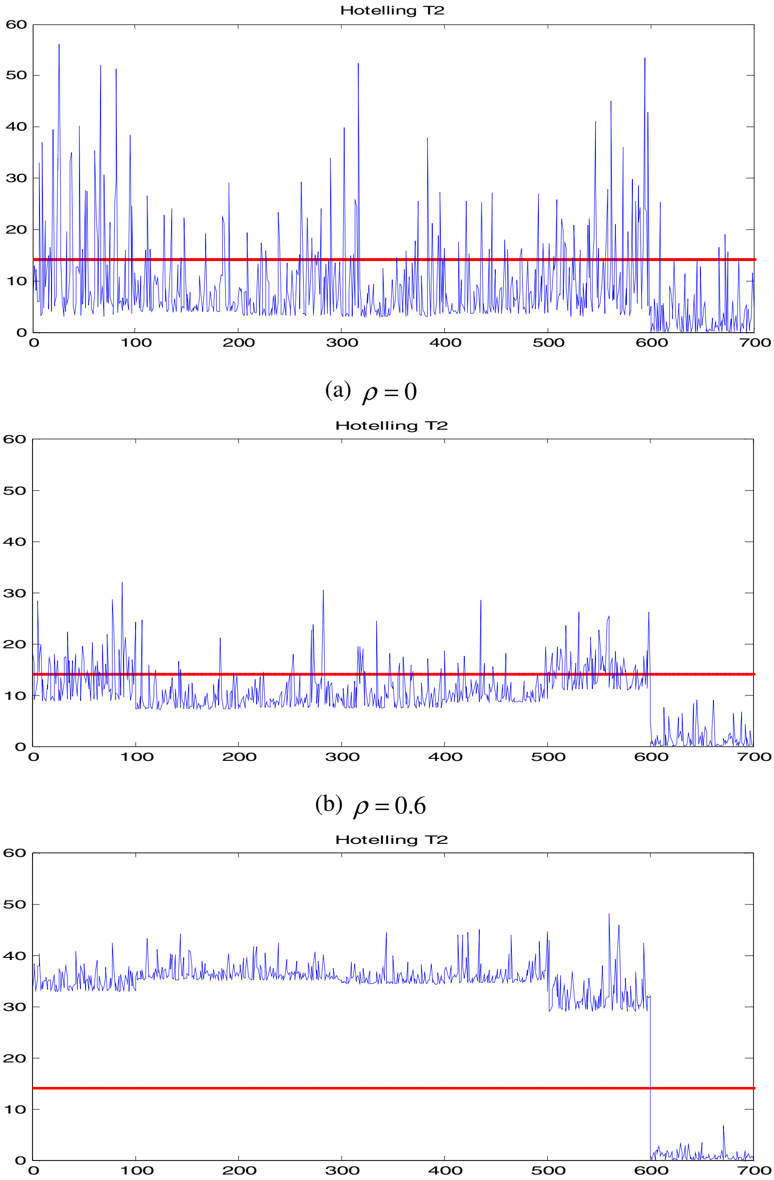


(a) $\rho = 0$

(b) $\rho = 0.6$

(c) $\rho = 0.9$

Fig. 1. 700 data sets of $\overline{X}_1$, $\overline{X}_2$, and $\overline{X}_3$ for the cases of $\rho = 0$, $\rho = 0.6$, and $\rho = 0.9$

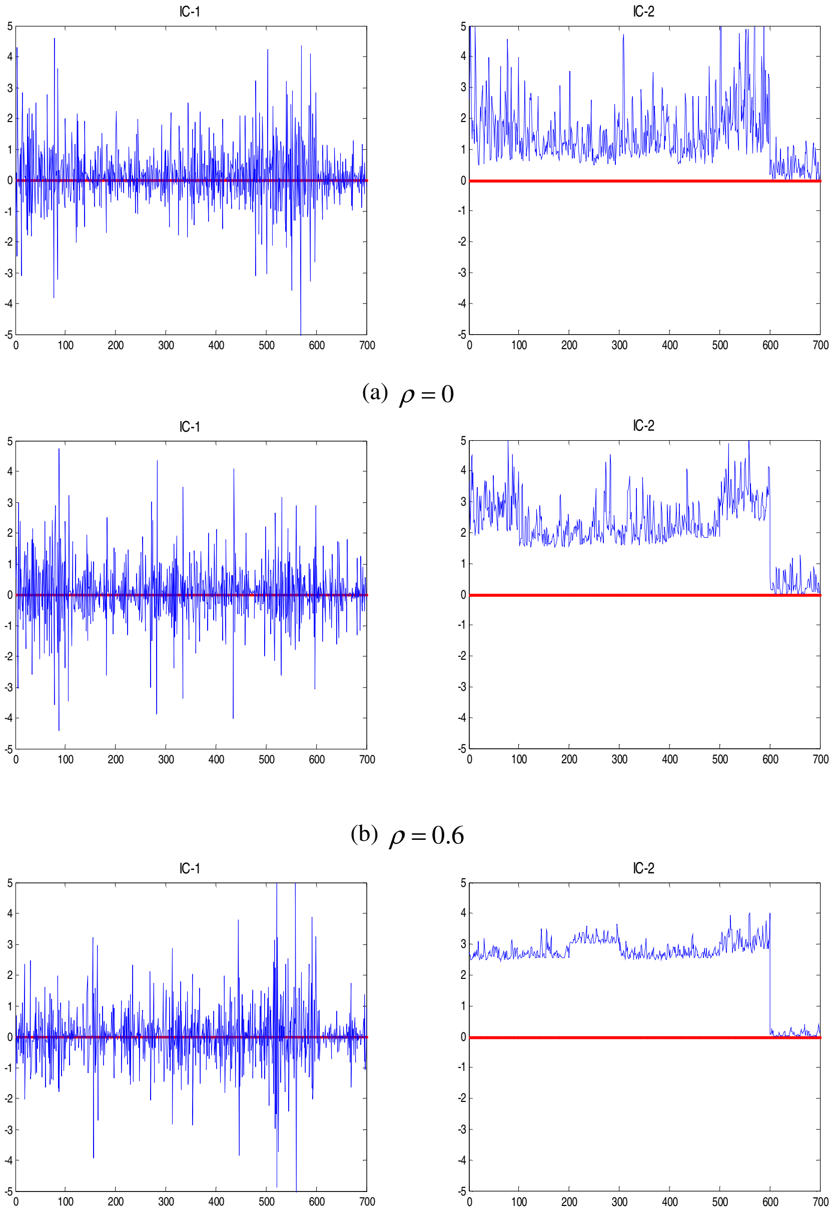


(a) $\rho = 0$

(b) $\rho = 0.6$

(c) $\rho = 0.9$

Fig. 2. The Hotelling T^2 statistics corresponding to the data sets in Figures 1



(a) $\rho = 0$

(b) $\rho = 0.6$

(c) $\rho = 0.9$

Fig. 3. The two ICs corresponding to the Hotelling T^2 statistics in Figures 2

Table 1. Results of CIR for typical and the proposed appraoches

	ρ =0		ρ =0.6		ρ =0.9	
Methods / Shift types	T^2+SVM	Proposed method	T^2+SVM	Proposed method	T^2+SVM	Proposed method
(1,0,0)	73.5	81.1	87.0	92.4	73.8	99.2
(0,1,0)	67.2	75.8	66.4	64.8	50.1	97.8
(0,0,1)	71.3	73.3	44.7	84.4	59.8	94.8
(1,1,0)	68.2	64.8	83.2	86.2	72.0	97.9
(1,0,1)	72.9	63.5	80.3	83.1	55.1	94.7
(0,1,1)	70.4	69.2	86.3	89.5	87.9	99.8
(1,1,1)	67.4	71.0	97.3	93.1	99.2	99.5

Table 2. Results of averaged CIR in the case of P=2, P=3, and P=5, respectively

	ρ =0		ρ =0.6		ρ =0.9	
Methods	T^2+SVM	Proposed method	T^2+SVM	Proposed method	T^2+SVM	Proposed method
P=2	87.40 (3.77)	85.10 (2.07)	96.16 (2.42)	97.40 (0.97)	93.03 (8.70)	99.43 (0.47)
P=3	70.12 (2.48)	71.24 (2.70)	77.88 (3.41)	84.78 (3.92)	71.12 (13.40)	97.67 (1.03)
P=5	43.62 (1.43)	49.49 (1.56)	70.56 (2.21)	78.67 (2.09)	41.05 (7.23)	87.34 (6.93)

of ρ =0.9 and the shift type of (1,0,0), the proposed approach is significantly superior than the typical approach. Observing Table 1, we can conclude that the proposed approach is better than the typical approach in most cases.

Table 2 demonstrates the experimental results of the testing phase in the case of P=2, P=3, and P=5, respectively. Considering the case of P=5 and ρ =0.9, the typical approach shows that the averaged CIR is 41.05% while the proposed approach is

87.34%. The number in parentheses stands for the standard error of the averaged CIR. The smaller standard error indicates that the identification mechanism is more consistent. Again, comparing the two standard errors of 7.23 and 6.93, we can conclude that the proposed approach is much better.

4 Conclusion

Determination of the fault quality variables for an out-of-control multivariate process is very important in practice. While most of the studies use the single step of classification, this study proposes the two-step approach, ICA-SVM, to overcome the difficulties. The proposed ICA-SVM scheme is able to enhance the correct identification rate for the determination of fault quality variables.

The proposed scheme initially uses ICA to the Hotelling T^2 statistics to generate two ICs. As a consequence, the SVM model uses the two ICs as inputs for the proposed classification. In this study, three types of quality variables and correlations are considered for evaluating the performance of the proposed approach. Experimental results strongly agreed that the proposed ICA-SVM scheme is able to produce the better correct identification rate for the testing datasets. Observing the experimental results, we can strongly conclude that the proposed approach is able to effectively enhance the correct identification rate.

Acknowledgment. This work is partially supported by the National Science Council of the Republic of China, Grant No. NSC 97-2221-E-030-012-MY2.

References

1. Runger, G.C., Alt, F.B., Montgomery, D.C.: Contributors to a Multivariate Statistical Process Control Chart Signals. Communications in Statistics-Theory and Methods 25, 2203–2213 (1996)
2. Shao, Y.E., Hsu, B.S.: Determining the Contributors for a Multivariate SPC Chart Signal Using Artificial Neural Networks and Support Vector Machine. International Journal of Innovative Computing, Information and Control 5, 4899–4906 (2009)
3. Cheng, C.S., Cheng, H.P.: Identifying the Source of Variance Shifts in the Multivariate Process Using Neural Networks and Support Vector Machines. Expert Systems with Applications 35, 198–206 (2008)
4. Shao, Y.E.: An Integrated Neural Networks and SPC Approach to Identify the Starting Time of a Process Disturbance. ICIC Express Letters – An International Journal of Research and Surveys 3, 319–324 (2009)
5. Chiu, C., Shao, Y.E., Lee, T., Lee, K.: Identification of Process Disturbance Using SPC/EPC and Neural Networks. Journal of Intelligent Manufacturing 14, 379–388 (2003)
6. Mason, R.L., Tracy, N.D., Young, J.C.: Decomposition of T^2 for Multivariate Control Chart Interpretation. Journal of Quality Technology 27, 99–108 (1995)
7. Mason, R.L., Young, J.C.: Improving the Sensitivity of the T^2 Statistic in Multivariate Process Control. Journal of Quality Technology 31, 55–165 (1999)
8. Wang, C.H., Dong, T.P., Kuo, W.: A Hybrid Approach for Identification of Concurrent Control Chart Patterns. Journal of Intelligent Manufacturing 20, 409–419 (2009)

9. Tay, F.E.H., Cao, L.J.: Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting. IEEE Transactions on Neural Networks 14, 1506–1518 (2003)
10. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons, New York (2001)
11. Lu, C.J., Lee, T.S., Chiu, C.C.: Financial Time Series Forecasting Using Independent Component Analysis and Support Vector Regression. Decision Support Systems 47(2), 115–125 (2009)
12. Lu, C.J., Wu, C.M., Keng, C.J., Chiu, C.C.: Integrated Application of SPC/EPC/ICA and Neural Networks. International Journal of Production Research 46(4), 873–893 (2008)
13. Tay, F.E.H., Cao, L.J.: Support Vector Machine with Adaptive Parameters in Financial Time Series Forecasting. IEEE Transactions on Neural Networks 14, 1506–1518 (2003)
14. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Berlin (2000)
15. Kim, K.I., Jung, K., Park, S.H., Kim, H.J.: Support Vector Machines for Texture Classification. IEEE Transactions on Pattern Analysis and Machine Intelligenc 24, 1542–1550 (2002)
16. Shin, K.S., Lee, T.S., Kim, H.J.: An Application of Support Vector Machines in Bankruptcy Prediction Model. Expert Systems with Applications 28, 127–135 (2005)
17. Wang, X.: Hybrid abnormal Patterns Recognition of Control Chart Using Support Vector Machining. In: International Conference on Computational Intelligence and Security, vol. 2, pp. 238–241 (2008)
18. Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I., Ohno, H.: Monitoring Independent Components for Fault Detection. AIChE Journal 49, 969–976 (2003)
19. Lee, J.M., Yoo, C., Lee, I.B.: On-line Batch Process Monitoring Using Different Unfolding Method and Independent Component Analysis. Journal of Chemical Engineering of Japan 36, 1384–1396 (2003)
20. Lee, J.M., Yoo, C., Lee, I.B.: New Monitoring Technique with an ICA Algorithm in the Wastewater Treatment Process. Water Science and Technology 47, 49–56 (2003)
21. Lee, J.M., Yoo, C., Lee, I.B.: Statistical Process Monitoring with Independent Component Analysis. Journal of Process Control 14(5), 467–485 (2004)
22. Xia, C., Howell, J.: Isolating Multiple Sources of Plant-Wide Oscillations Via Independent Component Analysis. Control Engineering Practice 13(8), 1027–1035 (2003)
23. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley and Sons, New York (2001)
24. David, V., Sanchez, A.: Frontiers of Research in BSS/ICA. Neurocomputing 49, 7–23 (2002)
25. Hyvärinen, A., Oja, E.: Independent Component Analysis: Algorithms and Applications. Neural Networks 13, 411–430 (2000)
26. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Berlin (2000)
27. Hsu, C.W., Lin, C.J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Network 13, 415–425 (2002)
28. Hsu, C.W., Chang, C.C., Lin, C.J.: A Practical Guide to Support Vector Classification. Department of Computer Science and Information Engineering, National Taiwan University, Taipai, Taiwan (2003)

Dynamic Pattern Extraction of Parameters in Laser Welding Process

Gissel Velarde and Christian Binroth

Hugo Miebach GmbH, Welding Machines Division, Dortmunder Feld 51,
Dortmund, Germany
info.ws@miebach.de

Abstract. Tuning parameters is essential for the results of the welding process. In order to optimize the tuning process of welding parameters, we propose a system based on historical data of laser welding machines. On a given combination of materials, the system extracts patterns dynamically and classifies new cases with a relative accuracy, which depends on the selected data set. The analysis of the generated patterns helps decision makers to visualize important features in large databases and therefore, achieve optimal results.

Keywords: Data mining, patterns, welding parameters, rough sets, automatic laser welding process, coil joining.

1 Introduction

Hugo Miebach GmbH builds laser welding machines (See Figure 1) for joining strip ends of different material compositions and combinations of thicknesses. In industrial lines of more than 200 meters length, the automatic laser welding machine is one of the most significant components in the entry section [1]. The capacity of the machines varies according to the application i.e. Pickling Lines, Rolling Mills, Coupled Pickling Lines and Tandem Mills. The usual materials to be joined are low carbon steel, higher strength steel, Si-steel, all newly developed advanced high strength steels (AHSS), austenitic and ferritic stainless steel, aluminum and titanium. The strip can vary: from 0.4 mm to 6.8 mm thickness and from 600 mm to 2100 mm width [2]. Strip ends are joined within different combinations of thicknesses in defined ranges i.e. thick to thin or vice versa, and a corresponding set of parameters according to the production plan of the plant. This leads to several thousands sets of parameters for each machine.

The parameters are tuned according to the characteristics of the coils (rolls of e.g. steel strip). It means, the characteristics of the entry and exit materials, according to the direction of the production line, and an optimal combination of welding speed, laser power, focal position, laser head pressure, pre-heater and post-heater power, mechanical settings, etc. In the case of known combinations, there is a set of optimal parameters that has to be tuned for every welding machine. As expected, the industry develops and produces new materials and demands new combinations and ranges and therefore, new optimal sets of parameters. With the knowledge of the experts, these

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 350–361, 2010.

recipes are built and tested. The seam is inspected through an optical system developed jointly with Falldorf [3], called Quality Control Data System (QCDS) based on camera sensors, which inspects and evaluates the geometry of the seam by image processing and analysis of analog signals. Moreover, the samples are also mechanically tested with the “Erichsen” bulge test, which is a rapid factory test that proofs the resistance (strength) of the welded joint by pushing a metal ball with hydraulic force in several positions of the seam length. Additionally, some weld seam samples are sent to laboratories in order to study their structure under the microscope and measure the hardness. All this tasks are fulfilled so that the welding parameters are optimized until the seam is acceptable with a very small potential of failure.

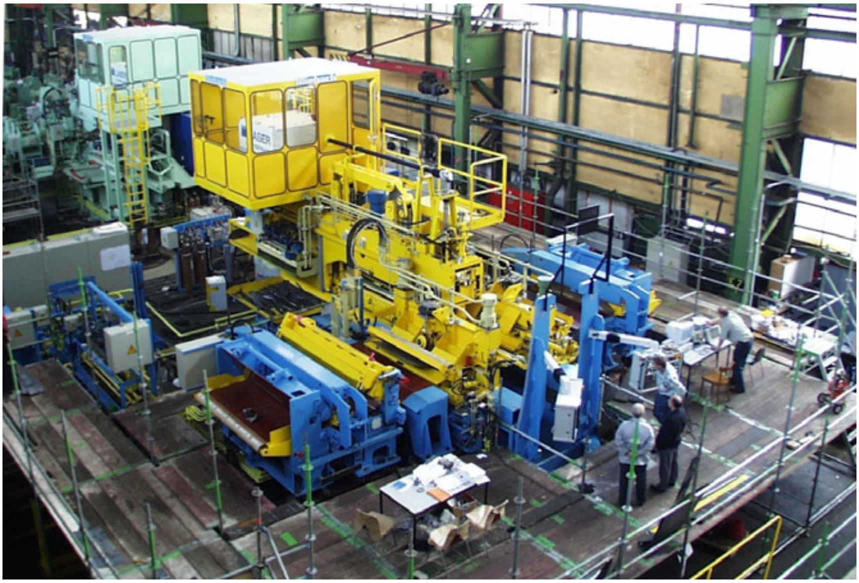


Fig. 1. Laser welding machine during testing and tuning process at Hugo Miebach GmbH., Dortmund, Germany

1.1 Related Work and Motivation

Previous studies focused mainly on the analysis of welding parameters [4],[5],[6]. Kim et al. used mathematical equations in order to find relationships between a reduced number of welding process parameters [4]. Chan and Na applied neural networks and numerical analysis of welding parameters to predict the bead shape in laser spot welding of thin stainless steel sheets as a possible prediction solution [5]. Olabi et al. found optimal ranges of parameters such as welding speed, laser power and focal position for CO_2 keyhole laser welding by means of the back propagation artificial neural network and the Taguchi approach for the experiment design [6].

We have a great collection of data, from machines in production and test that has not been mined. The data related to every weld can be inspected separately but not as a whole. Therefore, we propose a system that gathers the data of all welds in a relational database and analyses a great number of parameters from a statistical point of view as well as a data mining technique, in order to obtain patterns and classifiers, as support for the tuning process and finding of optimal welding parameters.

In chapter 2, we explain the technique we used in order to achieve our task. We give our conceptual approach, as well as the description of the patterns discovery technique and how we classify new cases. In chapter 3 we present the results. Finally, we state our conclusions and future work.

2 Technique

The technique to be described in this section, responds to the requirements addressed by the experts, who are concerned with the problem of recognizing dynamically the leading factors or patterns that affect the results of an optimal or pour weld in large collections of data in relational databases.

2.1 Approach

We assume that the quality of the seam is directly related to the weld settings for a selected combination of materials and thicknesses. Nevertheless, successful and unsuccessful welds share some information. The shared information can be seen as a boundary region between sets. Sets that cannot be precisely described can be approximated. This kind of sets are called Rough Sets [7], [8]. Our approach is based on the concept of roughness, discernibility and belonging.

Welds are objects described by their parameters and classified by the QCDS and saved in our relational database. Relevant parameters are gathered in a decisional database entity, which undergoes different stages of data manipulation until the generation of knowledge.

Conceptually, we want to detect patterns that reveal the roughness of sets and overlap new cases to a “cleaned” set of parameters. Moreover, when patterns are identified and visualized, experts can decide to remove patterns that lead to undesired results, and approximate new combination with suggested patterns.

In order to make the system adaptive, we built the model, in such a form that every time it creates new sets, patterns and classifiers depending on the characteristics of the materials.

2.2 Pattern Discovery

A relational database precise a defined structure in order to simplify the extraction of patterns. A *decision* table is a database entity where every object is described by attributes or parameters and is associated to a decision. Usually *decision* tables are large structures with a great population of objects. It is desired that its dimension be reduce preserving its complete information [9], [10], [11]. Moreover, dimension reduction is used for computational efficiency and classification performance [12], [13]. We propose a paradigm through a *mining* table, where every column corresponds to a parameter and is a nested table of all values of a set.

We used a typical machine learning approach, where the *decision* table splits into two tables, *training* and *test*. These tables are built picking sequentially one object for each table, being the intersection of both tables empty. When patterns are found, the objects of the *test* table are classified and evaluated.

Lets consider Table 1 as illustrative example of a *decision* table with *n* objects, four parameters and a binary result, and Table 2 as the corresponding *mining* table. The nested tables of the *mining* table are populated dynamically with the data of the *training* set and are classified as patterns according to its known results: *ok* or *nok*. Then, the *boundary* region is created through the multiset intersection between every nested table of the two previous patterns. This boundary region help us to find the *ok* pattern that does not contain boundary parameters, we call it the extreme pattern *xok*, as well as the extreme pattern *xnok* that does not contain boundary parameters.

Table 1. Illustrative example of a *decision* table with *n* objects, four parameters and a binary result

Object	Parameters1	Parameter2	Parameter3	Parameter4	Result
1	a	o	i	y	ok
2	e	p	j	y	nok
3	b	q	k	y	ok
4	c	m	l	y	ok
5	c	o	m	y	ok
6	a	o	n	y	nok
7	d	p	i	y	ok
8	e	q	k	y	ok
9	a	m	m	y	ok
…	…	…	…	…	…
n-1	b	q	k	y	ok
n	b	n	n	y	ok

Table 2. Illustrative example of a *mining* table containing nested tables for every parameter. The five patterns are built dynamically.

Pattern_id	Parameters1	Parameter2	Parameter3	Parameter4
ok	(a,b,c,d,e)	(o,p,q)	(i,j,k,l,m,n)	(y)
nok	(e,f,g)	(m,n)	(i,m,n)	(y)
boundary	(e)	( )	(i,m,n)	(y)
xok	(a,b,c,d)	(o,p,q)	(j,l)	( )
xnok	(f,g)	(m,n)	( )	( )

2.3 Dynamic Pattern Extraction

We built a relational ORACLE database, and populated it processing the data generated by QCDS. We selected the parameters related to the welding process, in our case more than 80 selected and representative parameters[1], and created a *decision* table, containing all objects associated to a binary result. The *mining* table is populated

[1] The selection criteria were performed based on expert knowledge and experience.

dynamically applying multiset operators, which combine the results of two nested tables into a single nested table [14]. At the end of the process, the *mining* table possesses five rows that correspond to the following patterns: *ok, nok, boundary, xok* and *xnok*. The cardinality of the patterns for every parameter is not greater than the number of the parameter's distinct set of values. The following pseudo code basically summarizes the essential multiset operations in order to obtain patterns from the *training* and *mining* tables:

```
--Class pattern extraction
For k in {ok, nok} loop
    For i in {1 to parameter_n } loop
        update mining table set parameter_i = (
            select cast collect values of parameter_i ∪ class_k as Nested_table_i
            from training table)
        where pattern_k

--Boundary region
For k in {boundary}
    For i in {1 to parameter_n } loop
        update mining table set parameter_i = (
            select (Nested_table_i ∪ pattern_ok) ∩ (Nested_table_i ∪ pattern_nok)
            from mining table)
        where pattern_k

--Extreme patterns
For j, k in {(ok, xok), ( nok, xnok)}
    For i in {1 to parameter_n } loop
        update mining table set parameter_i = (
            select (Nested_table_i ∪ pattern_j) - (Nested_table_i ∪ pattern_boundary)
            from mining table)
        where pattern_k
```

Consider that our system uses dynamic SQL in PL/SQL[2] [15]. However, instead of presenting the whole dynamic procedure, we want to present the function of multiset operators.

As seen previously in the pseudo code, consider the following statements within loops. The SQL statement used for class pattern extraction extracts patterns by creating a nested table from all *parameters* of the *training* table and updates the nested table for a *parameter* in a defined pattern:

[2] Dynamical SQL statements in PL/SQL update dynamically values using the EXECUTE IMMEDIATE statement and bind variables.

```
--Class pattern extraction
sql_stmt :=
'UPDATE mining SET '|| parameter ||' = (
      SELECT CAST(COLLECT('|| parameter ||') AS nested_table)
      FROM training
      WHERE result = :1)
WHERE pattern_id = :2';

EXECUTE IMMEDIATE sql_stmt USING class, pattern_id;
```

The value of the *parameter* will take the corresponding value of the parameter in the *training* table and *result* will take the values of the decision in the *training* table according to its cardinality. When all patterns related to a class are populated, we build the boundary region as the multiset intersection of distinct values as follows:

```
--Boundary region
sql_stmt :=
'UPDATE mining SET "'|| parameter ||'" = (
  SELECT '||parameterA||'
  MULTISET INTERSECT
  DISTINCT '|| parameterB ||' multiset_intersect
  FROM mining A, mining B
  WHERE A.pattern_id = :1 AND B.pattern_id = :2)
WHERE pattern_id = :3';

EXECUTE IMMEDIATE sql_stmt USING class1,class2,pattern_id;
```

The bind variables of the SQL statement that builds the boundary region will take values: *ok* for *class1*, *nok* for *class2* and *boundary* for *pattern_id*. Finally, we build the extreme sets with all values of the nested table for a determined pattern, except the values in the *boundary* region:

```
--Extreme Sets
sql_stmt :=
'update mining set "'|| parameter ||'" =(
  SELECT '||parameterA||'
  MULTISET EXCEPT DISTINCT '|| parameterB ||' multiset_except
  FROM mining A, mining B
  WHERE A.pattern_id = :1 AND B.pattern_id = ''boundary'')
where pattern_id = :2';

EXECUTE IMMEDIATE sql_stmt USING class,  pattern_id;
```

The above statement populates the extreme sets *xok* and x*nok*, where the bind variable *class* takes values: *ok* and *nok*, each time, as well as *pattern_id* takes values: x*ok* and x*nok*, respectively.

2.4 Classification

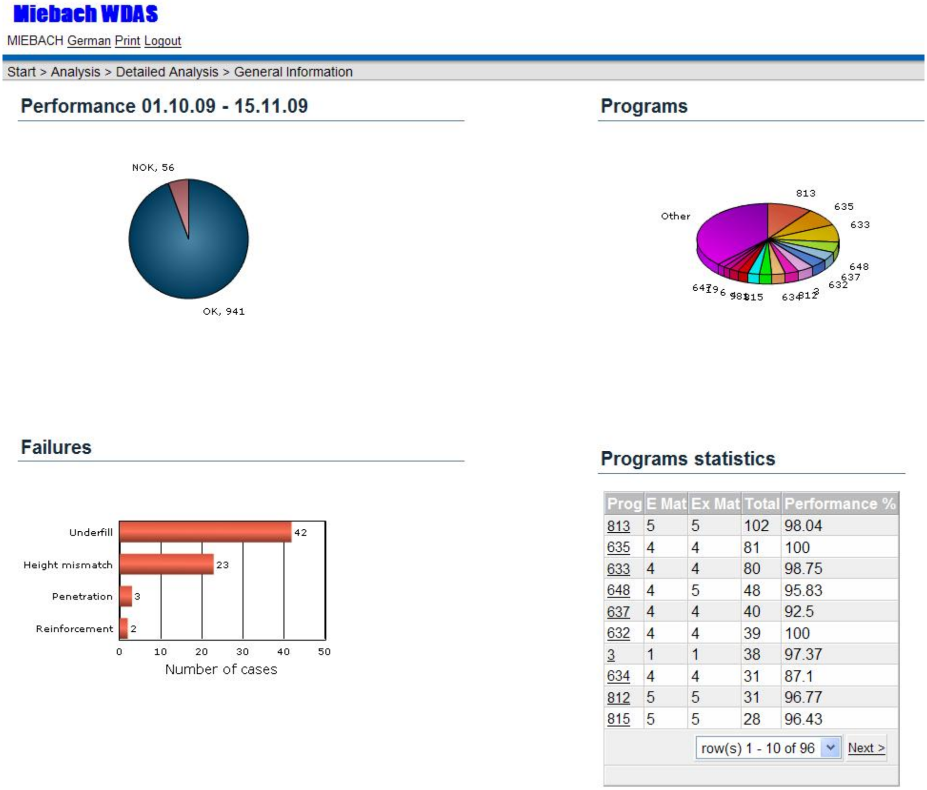
Once the patterns are extracted, we calculate an overlap coefficient μ between every case of the test set *Ct* and the extreme sets denoted by *X*. This overlap coefficient is the rate of hitting parameters.

$$\mu = \frac{|Ct \cap X|}{|X|} \ . \tag{1}$$

Every new case will be classified to one of the extreme sets according to its overlap coefficient. In case that this coefficient be equal to both sets or zero, then we assign the new case to one of the sets according to the probability distribution of the training set.

3 Results

The approach described in section 2 was tested using data of one machine in tuning stage. After mining the data, the experts analyzed the findings and used the patterns found in order to optimize the data sets. We present the status of the machine in a first tuning stage, followed by the evaluation of the patterns found and finally, the status of the machine in a second tuning stage.



Prog	E Mat	Ex Mat	Total	Performance %
813	5	5	102	98.04
635	4	4	81	100
633	4	4	80	98.75
648	4	5	48	95.83
637	4	4	40	92.5
632	4	4	39	100
3	1	1	38	97.37
634	4	4	31	87.1
812	5	5	31	96.77
815	5	5	28	96.43

Fig. 2. Statistical information about the performance of one Laser Welding Machine in a first tuning process. Number of welds 997, performance 94.38 %.

3.1 Nature of the Data

Welds are classified into weld programs according to the material characteristics and combinations. Weld programs contain empirical parameters that should be adjusted and optimized for every machine. Figure 2 shows the statistical information of one machine in a first tuning stage. The chart in the upper left corner shows the amount of welds, *ok*:941 and *nok*:56. The lower left corner shows the failures of the 56 *nok* welds. Consider that a weld can present more than one failure. In the right upper corner, the chart shows the percentage of weld programs used. Finally, the 10 most representative weld programs, in terms of number of welds, are shown in a statistics report with the materials welded, the total number of welds, and the performance of every program.

The description of the materials assigned to the machine can be seen on Table 3. The thickness ranges of every material group can be seen on Table 4.

3.2 Results and Evaluation

Table 5 describes the most representative combinations, in terms of number of welds, of materials and thicknesses tested. The corresponding results can be seen in Table 6,

Table 3. Material code and description

Material group	Steel grade
1	Interstitial free
2	Bake hardening
3	RePhos
4	Forming light
5	Forming heavy
6	HSLA light
7	HSLA heavy
8	Dual phase

Table 4. Thickness ranges defined for all material groups

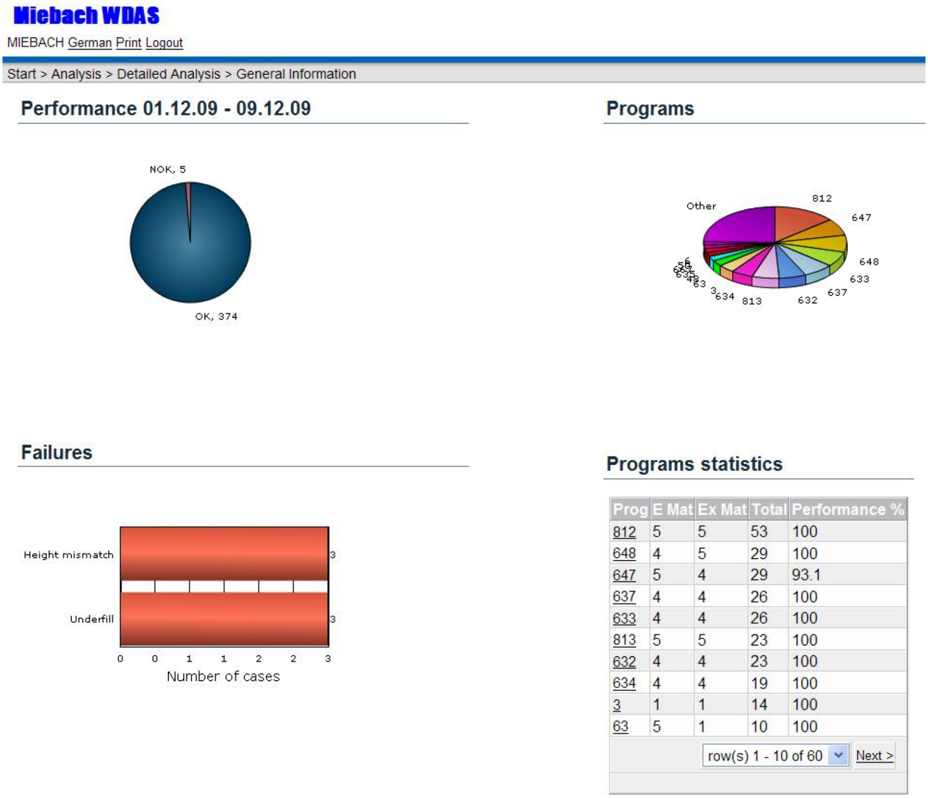
Range id	Lower limit mm	Upper limit mm
1	0.4	0.54
2	0.55	0.74
3	0.75	0.99
4	1	1.24
5	1.25	1.49
6	1.5	1.79
7	1.8	2.29
8	2.3	2.79
9	2.8	3.3
10	3.31	3.79
11	3.8	4.29
12	4.3	4.79
13	4.8	5.29
14	5.3	5.79
15	5.8	6.79

Table 5. Representative combinations of entry and exit material, and their respective thickness range as defined in Table 4

No.	Material Entry - Exit	Ranges mm Entry - Exit
1	4 – 4	2 – 2
2	4 – 4	3 – 3
3	5 – 5	2 – 2
4	4 – 4	5 – 5
5	5 – 5	3 – 3
6	4 – 4	7 – 7
7	5 – 5	5 – 5
8	1 – 1	7 – 7
9	6 – 6	6 – 6
10	4 – 5	3 – 3
11	4 – 4	4 – 4
12	5 – 4	2 – 2
13	5 – 4	3 – 3
14	5 – 4	2 – 3
15	4 – 4	2 – 3
16	1 – 1	3 – 3
17	1 – 1	5 – 5
18	1 – 5	3 – 3
19	4 – 5	3 – 2
20	1 – 1	3 – 4

Table 6. Number of patterns extracted and classification performance of the most representative combinations

No.	TN	FP	TP	FN	Total Tested	Xok Patterns	Xnok Patterns	Precision	Specificity	Sensitivity	Accuracy
20	1	0	6	0	7	22	17	1	1	1	1
16	7	0	23	0	30	16	8	1	1	1	1
15	18	0	13	1	32	30	8	1	1	0.93	0.9688
11	2	1	41	2	46	13	0	0.98	0.67	0.95	0.9348
3	0	7	112	2	121	17	1	0.94	0	0.98	0.9256
18	2	0	8	1	11	19	13	1	1	0.89	0.9091
12	0	4	39	0	43	16	0	0.91	0	1	0.907
8	43	2	4	3	52	9	16	0.67	0.96	0.57	0.9038
19	1	1	8	0	10	29	5	0.89	0.5	1	0.9
14	0	4	38	1	43	16	0	0.9	0	0.97	0.8837
5	0	10	79	3	92	13	6	0.89	0	0.96	0.8587
17	0	1	12	1	14	15	8	0.92	0	0.92	0.8571
6	9	8	47	1	65	9	15	0.85	0.53	0.98	0.8515
1	9	19	184	23	235	29	6	0.91	0.32	0.89	0.8213
7	9	11	34	0	54	15	4	0.76	0.45	1	0.7963
4	2	19	82	6	109	24	6	0.81	0.1	0.93	0.7706
9	1	10	37	4	52	17	2	0.79	0.09	0.9	0.7308
2	30	34	68	3	135	24	18	0.67	0.47	0.96	0.7259
10	12	9	21	4	46	14	8	0.7	0.57	0.84	0.7174
13	5	8	22	8	43	13	2	0.73	0.38	0.73	0.6279



Prog	E Mat	Ex Mat	Total	Performance %
812	5	5	53	100
648	4	5	29	100
647	5	4	29	93.1
637	4	4	26	100
633	4	4	26	100
813	5	5	23	100
632	4	4	23	100
634	4	4	19	100
3	1	1	14	100
63	5	1	10	100

Fig. 3. Statistical information about the performance of one Laser Welding Machine in a second tuning stage. Number of welds 379, performance 98.68 %[3].

which summarizes the number of tested objects, as well as the number of TN: True Negatives, FN: False Negatives, FP: False Positives, TP True Positives; the number of patterns found for both extreme sets, where this number represents the number of nested tables associated to the extreme sets and their Precision = TP/(TP + FP), Specificity= TN/(TN + FP), Sensitivity= TP/(TP + FN) and Accuracy=(TP + TN)/(TP+FN+TN+FP). Table 6 was reordered according to the accuracy of the patterns.

The results show that it is not necessary to retrieve a greater number of patterns in order to obtain the best performances, but perhaps extract the most significant ones. Nevertheless, when no patterns are retrieved for a class, it is clear that the system is less trustful classifying objects because the overlap coefficient is 0.

[3] Concerning the failure "Height mismatch" it is necessary to state that some times the real thickness difference of the strip ends provided to the welding machine is bigger than the expected thickness step because of thickness deviation. So the evaluation of this signal will show the deviation to the operator, who can still decide as good weld according to the evaluation result of other important signals by the QCDS.

A very important aspect for our analysis was focused not only in considering the accuracy as the most determining factor, but the specificity. We take special consideration of it, because misclassification of an object as successful implies very high costs in case of breakages of a weld in the production line. Nevertheless, it is important to remark that our special interest was extracting valuable information from the database, more than classifying new welds.

4 Discussion and Future Work

The accuracy of the system and the performance of the patterns are related to historical data. If the training data reveals at least a pattern or a set of patterns that appears again in the test set, the system has demonstrated to be able to classify new cases with a relative accuracy to the input data. On the other hand, when no patterns are found or recognized in the training set, or the patterns found in the training set are not reproduced in the test set, the system is less appropriated for the classification task. This occurs when cases are indiscernible, it means, successful and unsuccessful cases share almost all data and therefore, we consider it as a lack of information. In this case, we can assume that the results are influenced by mechanical factors, or by other variables that are not considered in the model. Therefore, the present system can be seen as a complementary tool for welding parameter analysis and as a tool for the decision making process of the welding machine operator. In this sense, once the patterns are identified, the users of the system are encouraged to analyze these findings.

We did not compare this approach with other algorithms in this study. However, we are interested in testing several models in order to contrast results and analyze other aspects that have not been considered until now.

Figure 3 shows the performance of a machine in a second tuning stage. In comparison with Figure 2, it can be seen that the most representative programs in terms of number of welds, improved considerably its performance. The parameters for this machine were not optimized automatically. Experts considered the patterns found as suggestion and adjusted the parameters, combining their experience and knowledge with the *mining* table. As improvement to the system, it is desired that the expert be able to edit the *mining* table when appropriate.

The visualization of patterns can be seen as a support for decision makers and analysts, and can also serve as a tool for a better understanding of the welding parameters during the introduction and further use of the new laser welding technology in continuous steel strip production.

References

1. Binroth, C.: Lasersysteme für Contilinien in der Stahlindustrie. In: Sepold, G., Seefeld, T. (eds.) Strahltecknik Laserstrahlfügen: Prozesse, Systeme, Anwendungen, Trends, vol. 19, pp. 87–96. BIAS Verlag (2002)
2. Hugo Miebach GmbH, `http://www.miebach.de`
3. Quality Control Data System, `http://www.falldorfsensor.de`

4. Kim, I.S., Basu, A., Siores, E.: Mathematical models for control of weld bead penetration in the GMAW process. The International Journal of Advanced Manufacturing Technology 12(6), 393–401 (1996)
5. Chang, W.S., Na, S.J.: Prediction of laser-spot-weld shape by numerical analysis and neural network. Journal Metallurgical and Materials Transactions B, 723–731 (2001)
6. Olabia, A.G., Casalinob, G., Benyounisa, K.Y., Hashmia, M.S.J.: An ANN and Taguchi algorithms integrated approach to the optimization of CO_2 laser welding. In: Advances in Engineering Software, vol. 37, pp. 643–648. Elsevier, Amsterdam (2006)
7. Pawlak, Z.: Rough Sets. In: Theoretical Aspects of Reasoning About Data, Kluwer Academic Publishers, Dordrecht (1991)
8. Cios, K., Pedrycz, W., Swiniarski, R., Kurgan, L.: Data mining: A knowledge discovery approach. Springer, Heidelberg (2007)
9. Komorowski, J., Pawlak, Z., Polkowski, L., Skowron, A.: Rough sets: A tutorial. In: Pal, S., Skowron, A. (eds.) Rough Fuzzy Hybridization: A New Trend in Decision-Making, pp. 3–98. Springer, Singapore (1999)
10. Vinterbo, S., Øhrn, A.: Minimal approximate hitting sets and rule templates. In: Predictive Models in Medicine: Some Methods for Construccion and Adaptation. Department of Computer and information Sciences, NTU report 1999, 130 (1999)
11. Swiniarski, R.: Rough Sets Methods in Feature Reduction and Classification. Int. J. Appl. Math. Comput. Sci. 11(3), 565–582 (2001)
12. Perner, P.: Prototype-based classification. Applied Intelligence 28(3), 238–246 (2008)
13. Lee, M., Park, C.H.: On applying Dimension Reduction for Multi-labeled Problems. In: Perner, P. (ed.) MLDM 2007. LNCS (LNAI), vol. 4571, pp. 131–143. Springer, Heidelberg (2007)
14. Oracle Database SQL Reference 10g Release 2 (10.2). Oracle (2005)
15. Moore, S., Belden, E.: Oracle Database PL/SQL Language Reference, 11g Release 1 (11.1) B28370-05. Oracle (2009)

Trajectory Clustering for Vibration Detection in Aircraft Engines

Aurélien Hazan[1], Michel Verleysen[1,2], Marie Cottrell[1], and Jérôme Lacaille[3]

[1] SAMM, Université Paris 1, 90 rue de Tolbiac, 75013 Paris, France
aurelien.hazan@univ-paris1.fr

[2] DICE, Université Catholique de Louvain, 3 place du Levant, B-1348
Louvain-la-Neuve, Belgium
michel.verleysen@uclouvain.be

[3] SNECMA, Groupe Safran, 77550 Moissy Cramayel, France
jerome.lacaille@snecma.fr

Abstract. The automatic detection of the vibration signature of rotating parts of an aircraft engine is considered. This paper introduces an algorithm that takes into account the variation over time of the level of detection of orders, i.e. vibrations ate multiples of the rotating speed. The detection level over time at a specific order are gathered in a so-called trajectory. It is shown that clustering the trajectories to classify them into detected and non-detected orders improves the robustness to noise and other external conditions, compared to a traditional statistical signal detection by an hypothesis test. The algorithms are illustrated in real aircraft engine data.

1 Introduction

We tackle the issue of monitoring the behavior of an aircraft engine from the point of view of measured vibrations. Indeed, abnormal level or odd pattern of vibrations may be the consequence of mechanical or sensor malfunction, both of dramatic importance for engine manufacturers and airline operators.

More precisely this work focuses on the detection of the signature of specific parts of a turbofan engine (the fans), whose vibration levels are modelled as vibration *trajectories*, i.e. as the amount of vibration of a given fan with respect to time. The trajectories are then clustered thanks to standard clustering algorithms. The obtained clusters gather all trajectories that correspond to fans whose vibratory signature is present in the data.

Section 2 gives a more thorough introduction to the problem. In Section 4 the algorithms are presented. Results are discussed in Section 5, while conclusion and perspectives are sketched in Section 6.

2 Problem Description

A turbofan whose structure is presented by Fig. 1 is considered. Air from the outside enters an intake, then is successively compressed by the low-pressure (LP)

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 362–375, 2010.

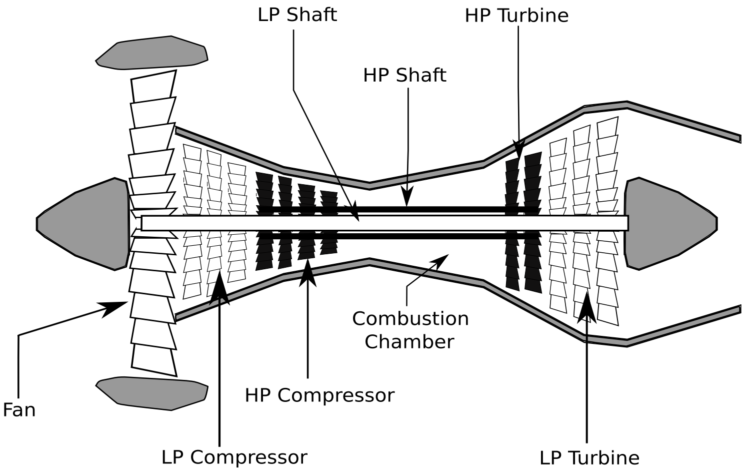


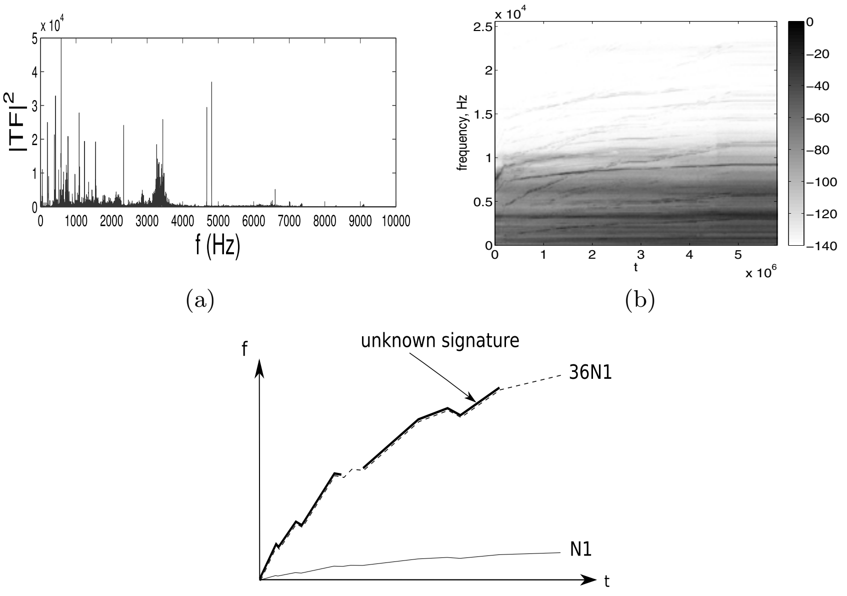
Fig. 1. Turbofan engine. Simplified diagram of fan, low-pressure and high-pressure compressors and turbines attached to their respective shafts.

and high-pressure (HP) compressors. Compressed air passes to a combustion chamber, where it is mixed with fuel and burnt. Both compressors are powered by turbines located at the rear of the engine, which transmit their energy to the compressors through two contra-rotating shafts, the low-pressure (LP) shaft and the high-pressure (HP) shaft.

Although turbofan condition monitoring can be achieved in various ways, we take the stand to focus on vibrations monitoring in this work. Two accelerometers provide vibration measurements at a constant 51 kHz frequency. Since compressors and turbines are fan-like components made of a varying number of blades mounted on the shafts, it is expected that their motion entails vibrations at frequencies which are multiples of shaft speeds. Although this is a strong simplification of the overall vibratory behavior of a bladed disk mounted on a rotor [1], it allows an efficient detection of malfunction and damages, given the low quantity of available information (two vibration sensors). Vibration patterns corresponding to multiples of shaft speed are known as “orders” in the engineering field.

Moreover, vibrations signals are usually processed not in the time-domain, but in the frequency [2] or in the time-frequency domain [3, 15.6]. When the rotation speed of the engine is constant, vibration signals can be considered as stationary, and the classical Fourier transform is sufficient. However when the rotation speed is varying, signals are non-stationary and the spectrogram or more advanced time-frequency distributions might be of some help. Examples of a Fourier transform and a spectrogram on a vibration signal are given in Fig. 2(a,b).

In addition to accelerometers, the sensors give an approximation of both shaft velocities. Since many mechanical parts of the engine rotate at a speed multiple of these shafts speeds (compressors and turbines for example), we can anticipate



(a)

(b)

(c)

Fig. 2. Vibration analysis of accelerometric data: (a) in the frequency domain with the Discrete Fourier Transform; (b) in the time-frequency domain with the spectrogram; (c) idealized signature of fan-like mechanical part in accelerometer data. An unknown signature in the spectogram and an expected signature at 36 times the frequency of LP shaft superpose.

the position of their signature in the frequency or time-frequency domain, as illustrated in Fig. 2(c) where an unknown signature in the spectrogram and an expected signature superpose at a particular order.

As we can notice from Fig. 2(b), the signature is embedded in noise. For specific orders, it remains clearly visible whereas for others it vanishes. Furthermore, there is a potentially large number of orders. For both reasons, order detection is most of the time the task of an expert. Automatic detection would hence accelerate the process, and relieve the expert of this daunting task.

Several difficulties arise when considering the issue of automatic detection of orders in vibration data. First, as already mentioned, the signal is noisy. Second, the magnitude of vibration of the orders is a function of the excitation. The latter is a direct function of the shafts speeds. Indeed, the fans behave as mechanical parts that are excited and resonate for specific frequencies only. Consequently, if we examine a sufficiently long recording, the orders will appear with varying magnitude along time. The frequency response of rotating machines is classically summarized in Campbell diagrams where the experimentally[1] measured

[1] Such diagrams can be drawn from analytical models as well, if the eigenfrequencies can be computed.

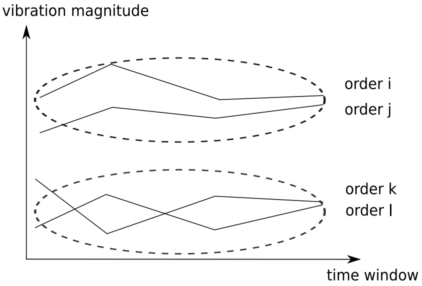


Fig. 3. Idealized diagram of trajectories clustering

vibration response spectrum appear as a function of the rotation speed of the shaft. If we suppose that the underlying shaft rotation speed follows a regular enough time pattern, then there should be some degree of continuity in the vibration magnitude of the orders.

Admitting that signals are split into a finite number of windows for computational reasons, we propose to consider vibration magnitude *trajectories*, where to each time window corresponds a robust measure of the vibration magnitude at a given order, for a selected shaft. This needs to be done systematically since we ignore whether a given order k of LP or HP shaft is visible for a specific regime. At the end, a large number of trajectories will be available, which we propose to cluster thanks to dedicated algorithms in order to segregate those who correspond to orders that are present in the data from those who do not. This is summarized by Fig.3. The expected outcome of this algorithm is thus a cluster of trajectories that correspond to what the expert would have singled-out as orders that are actually present in the recordings.

Related works are given a quick review in Section 3, while a detailed presentation of available data is given in Section 4.1. The proposed method is discussed in Section 4.2. Canonical signal processing detection procedures in the case of partially unknown sine wave are evoked in Section 4.3 for benchmarking purposes.

3 Related Work

This work addresses the problem of feature detection, where the features are the vibration signatures of bladed disks that compose compressors and turbines of an aircraft engine.

It is intended to be part of a Condition Based Monitoring (CBM) framework. CBM for industrial machines has been attracting increasing attention over the years in both academic and industrial areas. According to [4] it consists in four main steps: data acquisition, feature extraction, feature selection, and decision-making. The first two steps rely on mechanical modeling or rotor dynamics [5], noise and vibration phenomena in rotating machines [6,4] and data analysis [7]. The latter builds on the general tools and methods developped in signal processing [8,3], statistical signal estimation and detection [9,10,11], learning theory, change detection [12], fault detection and isolation [13].

Aircraft CBM deals with many problems such as structural health monitoring. It treats engine health monitoring (EHM) as a special case [14,15]. As a subtopic of EHM, vibrations monitoring in engines addresses the following issues: rotor/stator contact [16], rotor unbalance, blade defects [17], bearing [18] and gearings defects [19]. Another important topic is *order tracking*, i.e. the precise estimation of the frequency and amplitude of a periodic signal whose leading frequency is varying in time. This may involve data resampling in the case of rotating machines [20,21], non-parametric time frequency methods such as the Gabor transform [22] or parametric methods such as the Vold-Kalman filter [23,24], that models the vibrating machine and use estimation tools to track the frequencies and amplitudes of orders.

When features have been extracted, decisions can be made, such as change (or novelty) detection. The decision concerning the health of the engine is either taken from a statistical viewpoint [25,12], or can derive from learning techniques such as neural networks [26,27].

In this work the aim is not to detect an anomaly, but a normal bladed disk signal. Up to our knowledge, there is no example of such order detection in the litterature. The case of novelty or abrupt change detection in engine have been covered, as well as feature extraction for many particular mechanical parts, but not for bladed disks in turbines and compressors. The works dedicated to these parts aim at estimating their rotation speed, amplitude and phase, but not to detect their presence. In the following we show that classical tools such as sine wave detection and unsupervised clustering enable us to tackle this issue.

4 Methods and Data

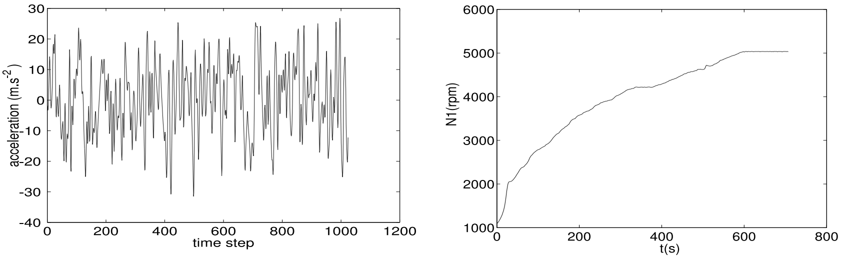
4.1 Data

The recordings under study were provided by the Health Monitoring Department of SNECMA[2] and correspond to a dual-shaft turbofan mounted on a testbench, that undergoes a continuous acceleration during several minutes. They include raw vibration outputs of an accelerometer sampled at 51kHz, as well as LP and HP shaft angular velocity computed from raw keyphasor data and sampled at 6.25 Hz. Sample time series are plotted in Fig. 4.

4.2 Clustering Contrast Trajectories for Order Detection

First we give a precise meaning to what was termed "vibration magnitude trajectory" in Section 2. Conforming to current practice in the field of vibration monitoring for rotating machine, we choose a classical time-frequency representation, the Discrete Gabor Transform (DGT) [22], which is a special kind of Discrete Short-Time Fourier Transform [8,28]. The representation of a given vibratory signal is a matrix, indexed in both time and frequency.

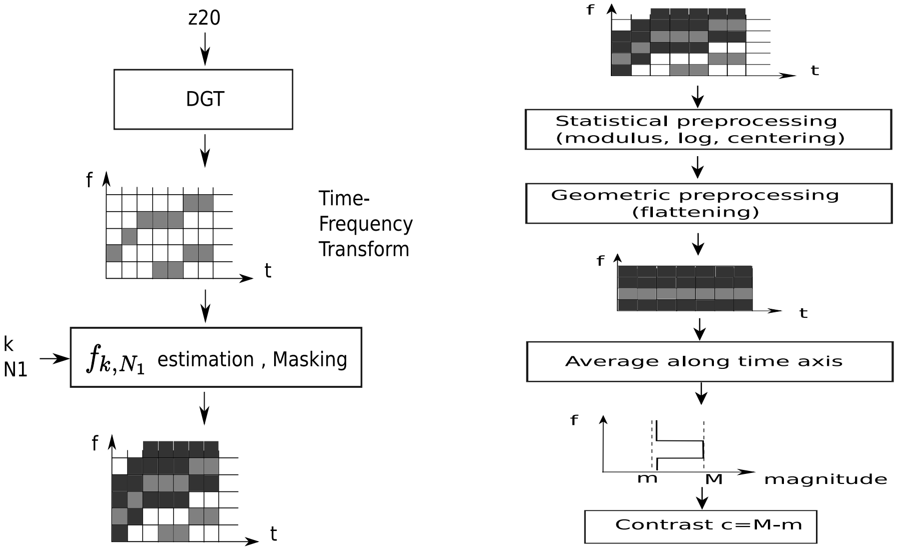
[2] http://www.snecma.fr



(a)

(b)

Fig. 4. (a) Raw accelerometric data; (b) LP shaft angular velocity in rpm



(a)

(b)

Fig. 5. (a) Order masking takes as its inputs the vibratory signal, a specific order integer k and the shaft rotation speed N_1 or N_2. The expected main frequency is f_{k,N_1}, and the region that surrounds ths central frequency has a fixed user-defined width ; (b) contrast.

As stated earlier, we focus on the signature of "orders", produced by the bladed disks that compose the compressors and turbines. Their simplified signature in the time-frequency domain has been illustrated by Fig. 2(c). We write $G(t, \nu)$ the complex-valued time-frequency transform at discrete time t and discrete frequency ν. Since we know approximately the frequency of an order from the choice of an integer number (the order) and the estimation of the shaft speed, we can build a region in the time-frequency plane in which we expect the order signature to appear, if it is present in vibratory data. This operation is called *masking* and is illustrated in Fig. 5(a).

Then, having focused on a specific order, we still need to compute a measure of magnitude for that order. Several statistical preprocessing steps (computation of the modulus, of the logarithm, then centering and scaling of the data) are followed by a geometric flattening of the signature as shown in Fig. 5(b). We call $G'(t,\nu)$ the resulting quantity. Finally the flattened signature is time-average, then the contrast function is computed.

$$\forall \nu,\ \tilde{G}(\nu) = \frac{1}{T}\sum_{t=t_1}^{t_2} G'(t,\nu) \tag{1}$$

$$c_k(n) = \max_{\nu}\ \tilde{G}(\nu) - \min_{\nu}\ \tilde{G}(\nu) \tag{2}$$

where k and n are respectively the order and the window indices, T is the length of the time window, and t_1 and t_2 are respectively the start and end of that time window.

Secondly, we describe the trajectory clustering algorithm announced in 2. The previous masking step returns a contrast value for each order k in a user-defined range $[k_1, k_2]$, and for each window $n \leq N$ that divides the full vibration recording. Hence for each order k we get an ordered set of scalar real-valued contrasts $T_k = \{c_k(1), \ldots, c_k(N)\}$ indexed by the order k. This set T is what we call a trajectory. We then propose to cluster it into a fixed number $C = 3$ of clusters, with the k-means algorithm. The whole procedure is illustrated in Fig. 6.

Clustering trajectories is a way to classify them into classes, corresponding respectively to orders that are detected, not detected, and a possible intermediate, ambiguous classes. One could argue that if an order k is effectively present in the recordings, any contrast $c_k(n)$ belonging to T_k should be high, allowing a detection on each time window n. However, one has to take into account the fact that the vibratory signals are extremely noisy. Splitting into windows is thus a way to expect an improved detection robustness in some windows. Averaging the detection hits over the windows could then improve the detection results, compared to a single detection (over one window or the whole signal).

Still, this average would completely lose the temporal correlation between orders. Indeed it is naturally expected that at some engine speeds (corresponding to some of the windows), all or most of the orders will be hardly detected (either because a large part of the engine will be less subject to vibrations, or because the latter will be polluted by higher noise). It is therefore of primary importance to take into account the longitudinal aspect of the detections at various orders, i.e. the temporal correlation. This is the justification for the original use of trajectories in this work.

In Section 4.3 we summarize classical results relevant to our problem in the field of statistical signal detection, in order to assess the performance of the proposed method.

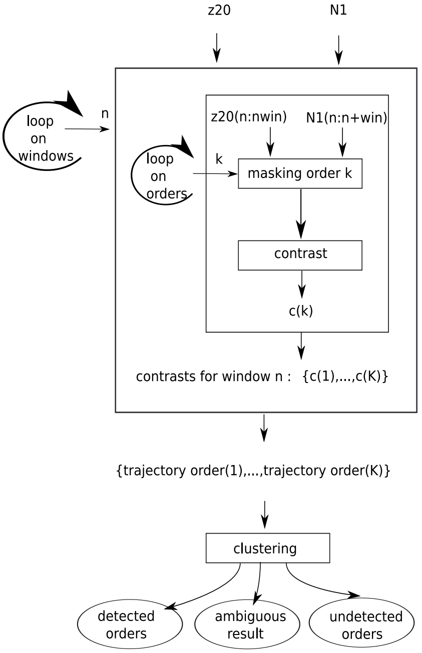


Fig. 6. Clustering algorithm

4.3 Statistical Signal Detection Method

From the point of view of classical statistical signal detection theory [9,10,11], the problem formulated in Section 2 is an instance of the detection of a deterministic sine wave in noise, with unknown parameters [9, 7.6.2]. Indeed, with the simplifying hypotheses stated in Section 2, the compressor and turbine disk blades whose signature is under study are seen as curves in the time-frequency plane, i.e. as sine waves with slowly varying leading frequency. On time windows where the frequency can be considered as constant, this sine wave can be parametrized as follows:

$$s(i) = A\cos(2\pi f i + \phi) \tag{3}$$

where A is the amplitude, i the time instant, f the frequency and ϕ the phase difference. Apart from this wave the signal contains a vast amount of other deterministic or random contributions which we call "noise" since they are irrelevant to the purpose of detecting a given order of a specific shaft. The detection of the sine wave in the noise is formalized as the following hypothesis test:

$$\begin{array}{ll} \mathcal{H}_0 : z20(i) = w(i) & \forall i \in [0, I-1] \\ \mathcal{H}_1 : z20(i) = w(i) + A\cos(2\pi f i + \phi) & \forall i \in [0, I-1] \end{array}$$

where $z20$ is the accelerometric signal, $w(i)$ is the noise term. It should be noted that, while the frequency f is approximately known, the amplitude A and phase difference ϕ are unknown to the observer. Under noise normality and independence hypotheses that diserve further discussion, this hypothesis test is amenable to analytic treatment under the Generalized Likelihood Ratio Test framework. Since this material is standard, we merely state the corresponding results, which are given proper development in classical textbooks (see for example [9, 7.6.2]):

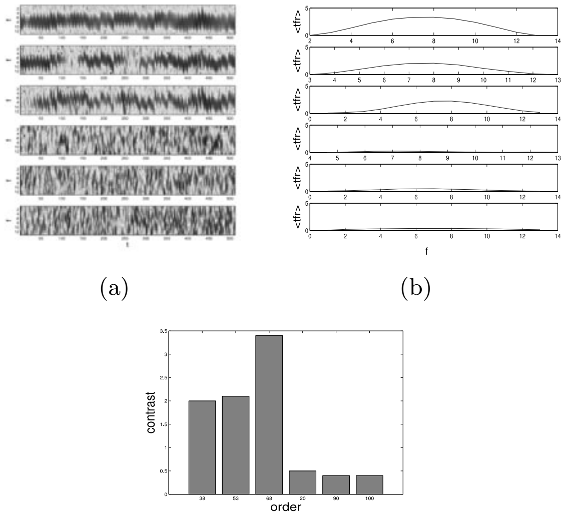
- the decision rule is written $I(f_0) \gtrless_{\mathcal{H}_0}^{\mathcal{H}_1} \gamma'$, where the decision statistics is $I(f_0)$, the value of the periodogram evaluated at target frequency f_0 which can be approximated by Discrete Fourier Transform. γ' is the detection threshold.
- the false alarm probability is $\alpha = \exp\left(-\frac{\gamma'}{\sigma^2}\right)$, where σ is the standard deviation of the noise, supposedly known. Usually α takes a fixed user-defined value such as $\alpha = 1\%$ so that the value of the detection threshold γ' can be easily derived.
- the performance of this test is quantified by the power π of the test, whose analytic expression can be computed.

With these results, we can build a sine wave detector. Given a vibratory time series and a few contextual parameters such as the noise variance, the detector will accept or reject hypothesis $\mathcal{H}_0$ depending if the signal is composed of noise or if an order is present in the signal. Note that several improvements over this standard procedure should be made, because the estimated noise does not respect exactly the white noise hypothesis. For example, the energy is not homogenous in the frequency spectrum. Consequently we divide the frequency spectrum into bins and perform the above detection separately in each bin. In Section 5 we compare the results with those elicited by the clustering algorithm.

5 Results

We first illustrate the computation of the contrast function defined in Section 4.2. The first steps depicted by Fig. 5 are the statistical and geometric preprocessing steps. Fig.7(a) shows the resulting preprocessed time-frequency transforms for six chosen orders of the HP shaft. Three orders (38,53,68) correspond to the number of blades of three bladed disks belonging to the compressor, therefore we expect the corresponding signatures to be present in vibration signals. Three ordrers (20,90,100) were selected because no significant vibratory activity is expected. In Fig.7(a) thick lines appear clearly in the first three cases, whereas no specific pattern except the background noise can be noticed in the last three cases. Time averaged values of the time-frequency transforms are plotted in Fig. 7(b), as well as their peak value in Fig. 7(c). The latter clearly shows that peak values are higher for orders 38, 53, 68 that correspond to actual mechanical rotating parts.

Secondly we comment on the contrast trajectory clusters that are found out by the clustering algorithm mentioned in Fig. 6. The number of clusters was set to $C = 3$. Results are shown in Fig. 8(a). The first cluster gathers trajectories with high mean value over the window range, that increases at the beginning,



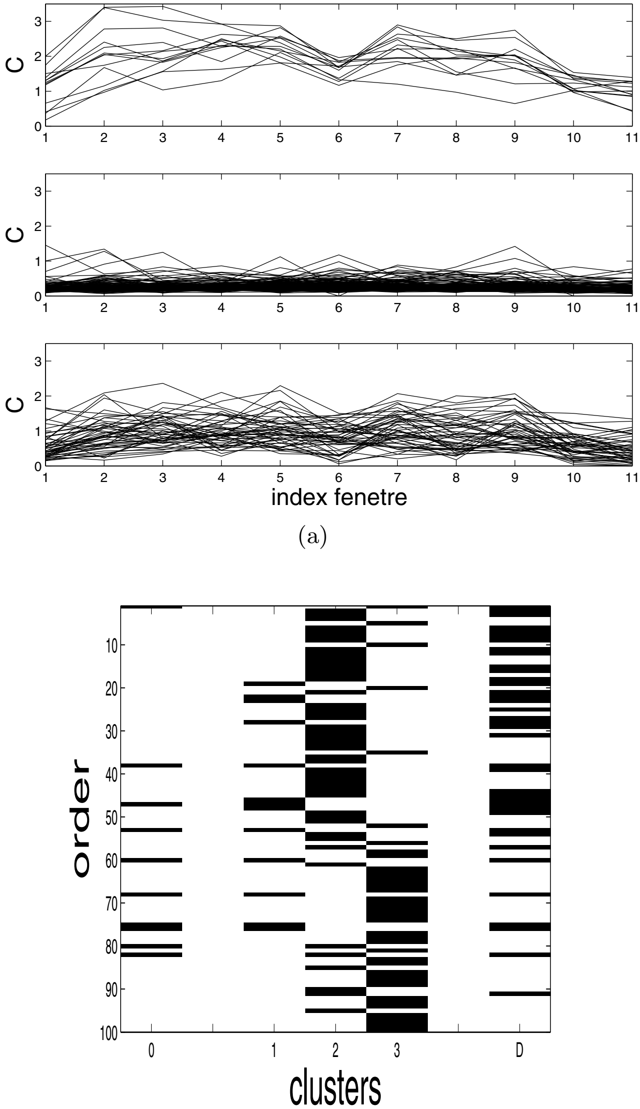
(a) (b)

(c)

Fig. 7. Contrasts of orders 38,53,68,20,90,100 of HP-shaft N_2: (a) DGT after preprocessing; (b) mean DGT along time axis; (c) contrasts

decreases at the end, and experiences a sudden decrease at window index 6. In the second one, trajectories are flat and have low mean value. The third cluster mixes different types of trajectories that have an average intermediate value. In first approximation, only the trajectories in cluster 1 are meaningful to the detection task.

In order to assess the significance of results, we use prior mechanical information. Indeed, the number of blades that compose the compressor and turbine mounted on HP shaft is known. Because of many mechanical factors it is not certain whether or not each bladed disk will have a noticeable vibratory activity, but this information can be helpful for comparison purposes. In Fig. 8(b), we plot the composition of the clusters obtained above (indices 1 and 3), and compare it with the cluster built with prior information which is given cluster label 0. Lastly, clusters found by the statistical detection algorithm from Section 4.3 are labelled as cluster D. We see that between orders 35 to 80 many orders are shared by clusters 0 and 1. This is true also for clusters 0 and D. However, clusters 2 and 3 show little similarity with cluster 0. This is coherent with our initial expectation, stated at the end of Section 2. In addition we remark that many low orders are detected in cluster D. This could be explained by the fact that low frequencies bear more energy that higher ones, as evidenced in Fig. 2(b). The clustering method is less prone to overweigthing such orders because of the higher energy content in the low orders area. Lastly the fact that orders not expected from mechanical knowledge appear both in clusters 1 and D suggest that interesting information not provided by naive mechanical data was actually discovered.



(a)

(b)

Fig. 8. Comparison of clustering and classical detection results for $z20$ vibratory data: (a) contrast vectors clustered by k-means with $C = 3$ clusters; (b) composition matrix for each cluster. Dark line with matrix coordinate i, j means that order i belongs to cluster j. Column 0 represents orders for which fan signature is expected from mechanical knowledge. Columns 1 to 3 are produced by the clustering algorithm. Column D is output by the statistical signal detection algorithm.

Lastly we confirm quantitatively these observations. Mutual information [29] is used here to measure the similarity between clusters. Setting cluster 0 as the default one, we compute the mutual information between the corresponding column of the composition matrix displayed in Fig. 8(b) and the other columns. What we expect is that self-information (i.e. the information between column 0 and itself) is high, while information between unrelated clusters such as 0 and 2 should be low. Moreover, mutual information between cluster pairs 0/1 and 0/D should be between the self-information value, and the unrelated clusters value. Indeed this is what we observe from Fig. 9, where the highest value corresponds to cluster pairs 0/0, while cluster pair 0/1 and 0/D have high mutual information. The negative value, which is theoretically meaningless, is a known limitation of the estimation algorithm. Lastly, the mutual information for cluster pair 0/D is lower than for cluster pair 0/1, which seems to indicate a higher performance of the proposed algoritm.

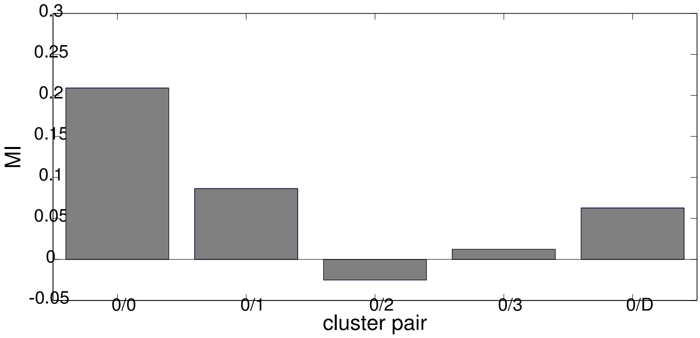


Fig. 9. Mutual information between pairs of clusters

The good results obtained by unsupervised clustering over signal-processing algorithms are suprising at first mainly because additional knowledge is embedded in the latter. Indeed, one needs to model the signal and the noise components before deriving a decision algorithm. We suggest that the results of the signal detection algorithm could be explained first by poor agreement between real data and noise hypotheses. This should be examined in further experiments.

Nevertheless, it remains that in the clustering approach we take into account the continuity of contrast values in time, which is a consequence of the continuity of shafts rotation speeds as a function of time. This continuity remains unexploited by the classical signal-detection algorithm, which iterates the decision process over successive windows without relating them.

6 Conclusion and Perspectives

In this work we have tackled the issue of order detection, i.e. the discovery of vibration patterns in noisy aircraft engine vibration signals. We proposed a contrast measure whose aim is to single out significant vibration patterns that correspond to compressor and turbines mounted on the engine shafts. Then a clustering algorithm is built, and compared to a statistical signal detection procedure. We show with real data that the clustering method performs well, and give quantitative measure of this performance. Future works will aim at:

– increasing the statistical significance by enlarging the database to several engines, in both acceleration and deceleration situations. Theoretical properties of the estimators of contrast, and of mutual information could be studied as well.
– improving the signal detection algorithm, for example by considering extension to colored noise situation. In addition the continuity from one window to the following could be used.
– merging the decisions from both methods.
– refining the clustering, mainly the interpretation of the intermediate cluster.
– assessing the continuity hypothesis and using it as prior for clustering, from the knowledge of theoretical Campbell diagrams.

Acknowledgements

We thank division YY of Snecma for providing us with vibrations data, more particularly S. Blanchard and J. Griffaton.

We also thank E. Côme for suggesting to compute similarity measure between clusters with mutual information.

Discrete Gabor Transform is computed thanks to the LTFAT library [30], available at http://ltfat.sourceforge.net/.

Mutual information is computed thanks to code made publicly available by Kraskov at http://www.klab.caltech.edu/~kraskov/MILCA/.

References

1. Bladh, R.: Efficient predictions of the vibratory response of mistuned bladed disks by reduced order modeling. PhD thesis, University of Michigan (July 2001)
2. Braun, S.: Mechanical Signature Analysis: theory and Applications. Academic Press, New York (1986)
3. Boashash, B.: Time-frequency signal analysis and processing - A comprehensive reference. Elsevier, Amsterdam (2003)
4. Randall: State of art in monitoring rotating machinery - Part I. Sound and Vibration 38(3), 14–21 (2004)
5. Muszynska, A.: Rotordynamics. Taylor & Francis, Abington (2005)
6. Lyon, R.: Machinery Noise and Diagnostics. Butterworths, Boston (1987)
7. Peng, Z.K., Chu, F.L.: Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography. Mechanical Systems and Signal Processing 18(2), 199–221 (2004)
8. Mallat, S.: Une exploration des signaux en ondelettes. Publications Ecole Polytechnique (2000)
9. Kay, S.: Fundamentals of statistical signal processing: detection theory. Prentice-Hall, Englewood Cliffs (1998)

10. Poor, H.: An introduction to signal detection and estimation, 2nd edn. Springer, Berlin (1994)
11. Van Trees, H.: Detection, estimation, and modulation theory-Part 1. John Wiley and Sons, Chichester (2001)
12. Basseville, M., Nikiforov, I.V.: Detection of abrupt changes: theory and application. Prentice-Hall, Englewood Cliffs (1993)
13. Gertler, J.: Fault detection and diagnosis in engineering systems. CRC Press, Boca Raton (1998)
14. Tumer, I., Bajwa, A.: A survey of aircraft engine health monitoring systems. In: 35th Joint Propulsion Conference. AIAA (June 1999)
15. Jaw, L.C., Mattingly, J.D.: Aircraft Engine Controls: Design, System Analysis, and Health Monitoring. AIAA Education Series (2009)
16. Peng, Z.K., Chu, F.L., Tse, P.W.: Detection of the rubbing-caused impacts for rotor-stator fault diagnosis using reassigned scalogram. Mechanical Systems and Signal Processing 19(2), 391–409 (2005)
17. Kharyton, V.: Fault detection of blades in blades ov an aviation engines in operation. PhD thesis, Ecole Centrale de Lyon (2009)
18. Orsagh, R., Sheldon, J., Klenke, C.: Prognostics/diagnostics for gas turbine engine bearings. In: Proceedings of IEEE Aerospace Conference (2003)
19. Wang, W., Ismail, F., Golnaraghi, M.: Assessment of gear damage monitoring techniques using vibration measurements. Mechanical Systems and Signal Processing 15(5), 905–922 (2001)
20. Potter, R., Gribler, M.: Computed order tracking obsoletes older methods. In: Proceedings of SAE Noise and Vibration Conference, pp. 63–67 (1989)
21. Fyfe, K.R., Munck, E.D.S.: Analysis of computed order tracking. Mechanical Systems and Signal Processing 11(2), 187–205 (1997)
22. Qian, S.: Gabor expansion for order tracking. Sound and Vibration 37(6), 18–22 (2003)
23. Vold, H., Leuridan, J.: Resolution order tracking at extreme slow rates, using Kalman tracking filters. In: Proc. SAE Noise and Vibration Conference, Traverse City, MI (1993)
24. Pan, M.C., Lin, Y.F.: Further exploration of Vold-Kalman-filtering order tracking with shaft-speed information-i: Theoretical part, numerical implementation and parameter investigations. Mechanical Systems and Signal Processing 20, 1134–1154 (2006)
25. Basseville, M., Le Vey, G.: Analyse et surveillance vibratoire d'une machine en rotation. In: Bensoussan, A., Lions, J., Thoma, M., Wyner, A. (eds.) Analysis and Optimization of Systems. LNCIS, vol. 111. Springer, Heidelberg (1988)
26. Ypma, A.: Learning methods for machine vibration analysis and health monitoring. PhD thesis, Technische Universiteit Delft (2001)
27. Staszewski, W., Worden, K.: Signal processing for damage detection. In: Staszewski, W., Boller, C., Tomlinson, G.R. (eds.) Health Monitoring of Aerospace Structures: Smart Sensor Technologies and Signal Processing. Wiley, Chichester (2004)
28. Feichtinger, H., Strohmer, T.: Gabor analysis and algorithms: theory and applications. Birkhäuser, Boston (1998)
29. Kraskov, A., Stögbauer, H., Grassberger, P.: Estimating mutual information. Phys. Rev. E 69(6) (2004)
30. Søndergaard, P.: Finite Discrete Gabor Analysis. PhD thesis, Institut for Matematik - DTU (2007)

Episode Rule-Based Prognosis Applied to Complex Vacuum Pumping Systems Using Vibratory Data

Florent Martin[1,2], Nicolas Méger[1], Sylvie Galichet[1], and Nicolas Becourt[2]

[1] University of Savoie, Polytech'Savoie, LISTIC laboratory,
Domaine Universitaire BP80439, 74944 Annecy-le-Vieux, France
{florent.martin,nicolas.meger,sylvie.galichet}@univ-savoie.fr
[2] Alcatel Vacuum technology
98 Avenue de Brogny, 74009 Annecy, France
{florent.martin,nicolas.becourt}@adixen.fr

Abstract. This paper presents a local pattern-based method that addresses system prognosis. It also details a successful application to complex vacuum pumping systems. More precisely, using historical vibratory data, we first model the behavior of systems by extracting a given type of episode rules, namely First Local Maximum episode rules (FLM-rules). A subset of the extracted FLM-rules is then selected in order to further predict pumping system failures in a vibratory datastream context. The results that we got for production data are very encouraging as we predict failures with a good time scale precision. We are now deploying our solution for a customer of the semi-conductor market.

Keywords: episode rules, FLM-rules, predictive maintenance, prognosis, vibratory signals.

1 Introduction

In the current economic environment, industries have to minimize production costs and optimize the profitability of equipments. Fault prognosis is a promising way to meet these objectives. Early detection of system behavior deviation indeed permits a better management of production means by anticipating rather than undergoing failures. Most of the prognosis applications come from medicine [1] or aerospace [9, 17]. In medicine, prognosis is defined as "the prediction of the future course and outcome of disease processes, which may either concern their natural course or their outcome after treatment" [1]. In aerospace domain, prognosis is defined as detecting the precursor signs of a system misfunction and predicting how much time is left before a major failure [17]. Research works in aerospace are very close to our application [9, 6]. We thus adopt the same point of view about prognosis objectives.

In the kind of pumping systems we want to monitor, common sensing technology (power levels, temperature, pressure and flow rates) is not sufficient for

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 376–389, 2010.

dealing efficiently with failure prognosis. We thus use vibratory data, more informative about system status, but also more difficult to handle. Indeed, because of their complex kinenatic, vacuum pumping systems may generate high vibration levels even if they are in good running conditions. It is thus difficult to prognose failures using traditional data analysis techniques (crest level, kurtosis, ...) and to develop monitoring methods based on expert knowledge or physical models. In this challenging context, a data-driven approach is preferred. In industrial applications, most of data-driven approaches are neural network based [17]. Unfortunately, neural networks are limited by their inability to explain their conclusions [12]. Thus we propose to extract local patterns, namely episode rules [14], from a large sequence of historical vibratory data, to describe the behavior of vacuum pumping systems. More precisely, we propose to extract *First Local Maximum-rules (FLM-rules)* as defined in [13]. FLM-rules are episode rules having an optimal temporal window width, which means that these rules are likely to appear within a specific temporal window width. This temporal window width can vary from a rule to another. In opposition to neural networks, such kind of rules is more interpretable. A subset of these rules is then used as a rule-based system to proceed to prognosis, i.e. to prognose seizings of pumping systems when considering our industrial application. This subset is generated by selecting the most predictive FLM-rules.

The prognosis of vacuum pumping system failures remains a challenge for process tool owners. Furthermore, explaining unexpected failures from typical vibratory behaviors and understanding early warning signs is of major interest for vacuum pump manufacturers. In this context, our contribution can be summarized as:

- running a recent data mining algorithm extracting FLM-rules from industrial vibratory data,
- providing a method for selecting the most predictive FLM-rules,
- providing a method for merging FLM-rules temporal information in order to prognose failures in a precise temporal window.

This paper is structured as follows: Section 2 reviews existing works in data mining applied to prognosis. Section 3 introduces our industrial application and describes the way data are selected and preprocessed. Section 4 presents the notion of FLM-rules while Section 5 details the process that is used to select the most predictive FLM-rules. In Section 6, we explain how to proceed to real time prediction, which includes matching the selected FLM-rules in a datastream and merging their respective temporal informations so as to provide a precise forecast window. Finally, experimental results are presented in Section 7 while Section 8 concludes and draws perspectives.

2 Related Work

Although maintenance is a manufacturing area that could benefit a lot from data mining solutions, few applications have been identified [8, 17] so far. Most of them are diagnosis applications (i.e. identifying problems without predicting them) [17] and mainly relate to aerospace [9] or to network analysis applications [7]. Prognosis applications often meet difficulties in predicting how much time is left before failure. End-users indeed have to set the width of the temporal window that is used to learn a model and predict failures [3].

In [9], Letourneau and al. present an approach that aims at predicting aircraft component faults. They rely on data mining techniques to build models from historical data. These models are then used to predict failures. More precisely, for each component, heterogeneous data (numerical, text) originating from measurements and maintenance reports are collected. Learning datasets are then defined according to the component replacement occurrences found in maintenance reports by selecting the data that have been recorded before and after replacements. Failure periods are defined by considering temporal windows (about 10% of the time scale of the dataset) that start just before replacement dates. All temporal aspects are user-defined. Classifiers are finally extracted using decision tree, nearest-neighbor or naive-bayes techniques. It is to note that the choice of the width of the failure periods significantly impacts results as it is used both to model and predict failures.

As further expounded in Section 3, the dataset we deal with is a large sequence of events, i.e. a long sequence of time-stamped symbols. Such a context has been identified in [7]. More precisely, in [7], a data mining application, known as the *TASA project*, is presented. It aims at extracting episode rules that describe a network alarm flow. These rules syntactically take the temporal aspect into account. They are known as *episode rules*. They are selected according to a frequency (or support) measure and a confidence measure (for more formal definitions, the reader is referred to [7]). In practice, users have to set the maximum temporal window width of the episode rule occurrences so as to make extractions tractable. Episode rule occurrences whose window width is greater than this maximum window width are indeed not considered. This approach can still generate a large amount of rules and experts are required to browse interesting rules according to their knowledge. Though this approach has not been designed to perform prognosis, it inspired some works that actually aim at predicting failures. For example, in [3], the authors propose a method that searches for previously extracted episode rules in datastreams. As soon as an episode rule is recognized, it is used for predicting future events. More precisely, they propose to build and to continuously maintain a queue of events which are likely to contribute to the occurrences of episode rules. Once the premiss of a rule is matched, they proceed to prediction by adding the maximum time span to the occurrence date of the first symbol of the episode rule. This method is interesting as it avoids scanning back the entire data stream but still, a crucial temporal information, i.e. the maximum window width, has to be set by users. Moreover, it has only been tested on synthetic data.

Beside asking for temporal parameters, the methods that are reviewed in this section all impose a same temporal window width for all possible models, for all possible rules. However, it is quite difficult to justify the use of a same maximum time span for each rule. It has thus been proposed in [13] to extract *FLM-rules*. These episode rules indeed come along with their respective *optimal temporal window widths*. In [11], we proposed a first approach for using those optimal window widths so as to establish prediction dates. An experiment on synthetic supply chain datasets was also presented. Though the results of that experiment were encouraging, and as explained in Section 6.1, the proposed approach was not consistent with respect to the definition of FLM-rules. Furthermore, dealing with vibratory data is more demanding than dealing with inventory levels. In this paper, we thus detail an improved version of this technique. We also describe a successful prognosis experiment that has been run on real production datasets acquired from vacuum pumping systems.

3 Application and Data Preprocessing

The aim of the approach described in this paper is to generate a valid model to identify abnormal behaviors of complex vacuum pumping systems (i.e. with a complex kinetic) and to predict a major default mode. Those systems are running under really severe and unpredictable conditions. They basically transfer gas from inlet to outlet. One major default mode that can alterate this function is the seizing of the pump axis. Seizings can be provoked by many causes such as heat expansion or gas condensation. Preventive maintenance plannings are not efficient. Therefore, Alcatel Vacuum Technology initiated a predictive maintenance project. With this aim in view, the quadratic mean (Root Mean Square, denoted RMS) of the vibration speed over 20 frequency bands has been collected over time at a frequency of 1/80 Hz. Real vibratory environnement is random by nature. Random vibrations can be measured on all mechanical systems such as vehicles or planes [10]. It also holds for vacuum pumping systems. Most of random vibration signals found in real applications follow a Gaussian distribution and are stationary time-function (i.e. they follow a zero-mean gaussian distribution). Thus, RMS is the standard deviation of the vibration speed [18]. It is measured by accelerometers and it is equal to the power of the vibration speed. Available data cover more than 2 years for 64 identical pumping systems. In order to build the learning dataset, we selected data that directly relate to some serious and fairly common failures we want to anticipate, i.e. first seizings. More precisely, we decided to build our learning dataset by only considering the data that are acquired before the first occurrence of such a failure. Indeed, after the first seizing, systems are seriously damaged and learning from potentially degraded systems would bias models. Moreover, if we succeed in predicting seizing, we will not observe such conditions. So we build learning sequences that start at the system startup date and end at the first failure occurrence. We got 13 doubtless sequences that end with the first occurrence of a seizing.

In order to detect evolutions of systems, for each system, we first determine the signal signature corresponding to good running conditions. The measured signal is then compared to this reference. Measurements represent the power the equipment is subjected to. It can be decomposed as follows: $P = P_0 + P_d$, with

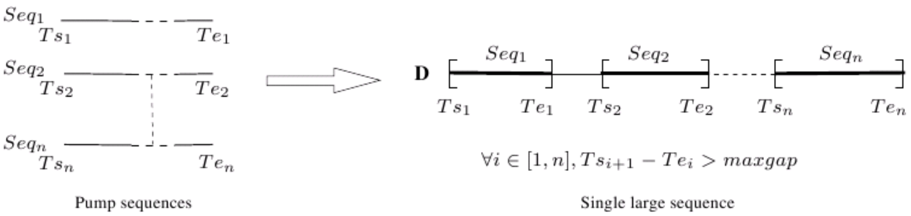


Fig. 1. Sequence building

P the measured power, P_0 the power in good running conditions and P_d the power of the default. According to experts, P_0 is defined for each frequency as the standard deviation calculated 24 hours after system power-on (so as to get a stable signal). It is calculated using sliding windows. Each time a new measure arises, P_0 is updated if the computed value is lower than the previous one. It generally takes up to 2 weeks to get a stable P_0.

The kind of default we are looking for generates a high level of power (P_d is high). Thus, for each frequency band we decide to focus on the values of the maximal envelope of measured power P. In order to perform FLM-rule extraction, the maximal envelope has to be encoded into symbols. First, to qualify the default severity, the ratio P/P_0 is computed and discretized using three levels. Then, we define a dictionary of 240 symbols, each symbol being associated with three pieces of information: the frequency band, the default severity and the duration at that severity level. We also introduce a specific symbol to represent seizing occurrences. Finally, we got 13 sequences containing 2000 symbols on average along with their occurrence dates. It is thus not recommended to use standard algorithms that extract patterns from a *collection of short sequences* (i.e. *base of sequences* [2]). Indeed short sequences generally do not contain more than about 500 events. Pattern extraction from a single *large sequence* [7, 13, 14] is then focused on. 13 sequences were thus concatenated into a single large sequence D. In order to avoid the extraction of FLM-rule occurrences spreading over various subsequences Seq_i, a large time gap between each initial sequence is imposed (see Fig. 1) and the *WinMiner* algorithm [13] is used for extracting FLM-rule occurrences. WinMiner can indeed handle a maximum time gap constraint (denoted as $maxgap$) between occurrences of symbols.

4 FLM-Rules

This section aims at defining the main concepts that are necessary to understand the notion of FLM-rules as originally proposed in [13]. First, we define the dataset itself. As previously explained in Section 3, our application context generates a large sequence.

Definition 1 (event, event sequence). *Let E be a set of symbols, namely event types. An* event *is defined by the pair (e, t) where $e \in E$ and t is an integer*

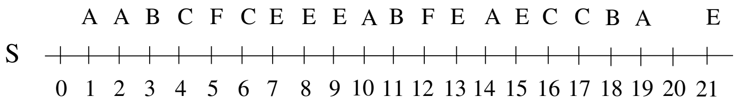


Fig. 2. An example of such an event sequence

giving the occurrence date of e. An event sequence is a triple $S = (s, T_s, T_e)$ where s is an ordered sequence of events $\langle(e_1, t_1), (e_2, t_2), ..., (e_n, t_n)\rangle$ *such that* $\forall i \in \{1, ..., n\}$, $e_i \in E \wedge \forall i \in \{1, ..., n-1\}$, $t_i \leq t_{i+1}$. T_s, T_e *are integers that denote the starting and ending time of the event sequence.*

Figure 2 depicts a toy example of such a sequence. FLM-rules are built upon a given kind of episodes, namely serial episodes.

Definition 2 (serial episode, prefix, suffix). *A serial episode is a tuple* $\alpha = \langle e_1, e_2, ...e_k\rangle$ *such that* $\forall i \in \{1, ..., k\}, e_i \in E$ *and there exists a total order relation between event types. The prefix of* α*, denoted by* $prefix(\alpha)$ *is the tuple* $\langle e_1, e_2, ...e_{k-1}\rangle$*. The suffix of* α*, denoted by* $suffix(\alpha)$ *is the singleton* $\{e_k\}$*.*

For the sake of simplicity, a serial episode $\alpha = \langle e_1, e_2, ...e_k\rangle$ is also denoted by $e_1 \rightarrow e_2 \rightarrow ... \rightarrow e_k$. As we only consider serial episodes, we will now refer to *episodes* instead of referring to *serial episodes*. For example, $A \rightarrow B \rightarrow C$ is an episode stating that B occurs after A and is followed by C. The prefix of $A \rightarrow B \rightarrow C$ is $A \rightarrow B$ and its suffix is C. Let us now define how an episode is said to occur within an event sequence:

Definition 3 (occurrence). *An episode* $\alpha = \langle e_1, e_2, ..., e_k\rangle$ *occurs in a sequence* $S = (s, T_s, T_e)$ *if there exists at least one* ordered sequence of events *s' =* $\langle(e_1, t_1), (e_2, t_2), ..., (e_k, t_k)\rangle$ *such that* s' *can be obtained by removing some elements of s or* $s' = s$ *(which will be denoted by* $s' \sqsubseteq s$ *in this paper) and* $\forall i \in \{1, ..., k-1\}$, $0 < t_{i+1} - t_i \leq$ *maxgap with maxgap a user-defined constraint that represents the maximum time gap allowed between two consecutive events.* $[t_1, t_k]$ *is an occurrence of* α*. The set of all occurrences of* α *in S is denoted by* $occ(\alpha, S)$*.*

The *maxgap* constraint is set both to reduce the search space and match recurrent application requirements. It has been introduced in [13]. It linearly constraints the window width of episode occurrences in function of the number of symbols that form the episode (instead of having a same maximum window width for all episodes). According to this definition, and by setting *maxgap* to 4 all along this section, $occ(A \rightarrow B, S) = \{[1, 3], [2, 3], [10, 11], [14, 18]\}$. Intervals $[1, 11], [2, 11], [1, 18], [2, 18], [10, 18]$ do not match the *maxgap* constraint. In order to reduce the size of such sets and to consider occurrences that do not already contain another occurrence, *minimal occurrences* are considered, as proposed in [14] and [13]:

Definition 4 (minimal occurrence). *A minimal occurrence of an episode α in a sequence S is a time interval $[t_s, t_e]$ containing α and such that there is no other occurrence $[t'_s, t'_e]$ verifying $[t'_s, t'_e] \subset [t_s, t_e]$. The set of all minimal occurrences of α in S is denoted by $mo(\alpha, S)$.*

Back to our example, the minimal occurrences of episode $A \rightarrow B$ in S are $mo(A \rightarrow B, S) = \{[2,3],[10,11],[14,18]\}$. The occurrence $[1,3]$ does not belong to $mo(A \rightarrow B, S)$ because it spreads over the occurrence $[2,3]$. We here recall that this definition relies on the occurrence definition that includes a *maxgap* constraint. *Minepi* [15] can not handle this constraint as it causes incompleteness. More precisely, if minimal occurrences of episodes of size k (i.e. having k event types) are considered, then it is not possible to compute episodes of size $k+1$. Let us consider sequence S, episode $A \rightarrow B \rightarrow C$ and episode A. With $maxgap = 4$ time units, $mo(A \rightarrow B \rightarrow C, S) = \{[2,4]\}$ and $mo(A, S) = \{[1,1],[2,2],[10,10],[14,14],[19,19]\}$ can not be used to generate the *minimal occurrence*$\{[2,10]\}$ of episode $A \rightarrow B \rightarrow C \rightarrow A$. Indeed, the minimal occurrence of $(A \rightarrow B \rightarrow C)$ in S occurs too early with respect to the ending date of $A \rightarrow B \rightarrow C \rightarrow A$. Therefore, in [13], the *WinMiner* algorithm has been proposed to extract all minimal occurrences of the episodes satisfying a *maxgap* constraint. For more details, the reader is referred to [13].

Episode rules are derived from episodes. Let α be an episode. An *episode rule* is the expression $prefix(\alpha) \Rightarrow suffix(\alpha)$. For example if $\alpha = A \rightarrow B \rightarrow C$, the episode rule built on α is $A \rightarrow B \Rightarrow C$. A more generic definition of episode rules can be found in [15]. Episode rules are characterized with two measures :

- the *support*: the number of occurrences of an episode rule over the whole sequence. The support of $A \rightarrow B \Rightarrow C$, denoted by *support*$(A \rightarrow B \Rightarrow C)$, is equal to the number of occurrences of episode $A \rightarrow B \rightarrow C$, denoted by *support*$(A \rightarrow B \rightarrow C)$.
- the *confidence*: the observed conditional probability of observing the conclusion of an episode rule knowing that the premiss already occurred. Confidence of $A \rightarrow B \Rightarrow C$ is thus defined as follows: $confidence(A \rightarrow B \Rightarrow C) = \frac{support(A \rightarrow B \rightarrow C)}{support(A \rightarrow B)}$.

Those measures are used for selecting episode rules according to a minimum support threshold σ and a minimum confidence threshold γ. As proposed in [13], support and confidence can be defined for each *window width*, i.e. the maximum time span of episode occurrences. In the example of Fig. 2, $mo(A \rightarrow B, S) = \{[2,3],[10,11],[14,18]\}$ and $mo(A \rightarrow B \rightarrow F, S) = \{[2,5],[10,12]\}$. If we consider a window width of 2 time units, then we have $Support(A \rightarrow B \Rightarrow F, S, 2) = 1$ and $Confidence(A \rightarrow B \Rightarrow F, S, 2) = \frac{1}{2}$. This means that $A \rightarrow B \Rightarrow F$ occurs once and has a confidence of 50% for a 2 time units window width. If, for a given episode rule λ, and for the shortest possible window width w,

- the support of λ is greater or equal to σ,
- the confidence c_w of λ is greater or equal to γ,

– there exists a window width $w'|w' > w$ such that such confidence of λ for w' is $decreaseRate\%$ lower than c_w,
– there is no window width between w and w' for which confidence is higher than c_w,

then, episode rule λ is said to be a *First Local Maximum-rule* or *FLM-rule*. Parameter *decreaseRate* is user-defined and allows selecting more or less pronounced local maxima of confidence with respect to window widths. The window width w corresponding to a first local maximum is termed as the *optimal window width* of FLM-rule λ. If we set *decreaseRate* to 30%, σ to 2, γ to 100% and *maxgap* to 4, then rule $r = A \rightarrow B \Rightarrow F$ (Fig. 3) is a FLM-rule which has a first local maximum of confidence for a 3 time units window. This can be interpreted as: "if I observe the premiss of r at t_0, then its conclusion must appear within t_0 and $t_s + w$". Parameter γ can be of course set to lower values. In this case, the probability of observing the conclusion between t_0 and $t_s + w$ is greater or equal to γ. For more formal definitions, reader is referred to [13].

As further explained in Sections 5 and 6, for predicting failures, most predictive FLM-rules ending on the symbol seizing are retained.

window width	1	2	3	4	5
$mo(A \rightarrow B \rightarrow F, S)$	$\varnothing$	$\{[10,12]\}$	$\{[2,5],[10,12]\}$	$\{[2,5],[10,12]\}$	$\{[2,5],[10,12]\}$
$mo(A \rightarrow B, S)$	$\varnothing$	$\{[2,3],[10,11]\}$	$\{[2,3],[10,11]\}$	$\{[2,3],[10,11],$ $[14,18]\}$	$\{[2,3],[10,11],$ $[14,18]\}$
$support(A \rightarrow B \rightarrow F, S)$	0	1	2	2	2
$support(A \rightarrow B, S)$	0	2	2	3	3
$confidence(A \rightarrow B \Rightarrow F, S)$	0	$1/2 = 50\%$	$2/2 = 100\%$	$2/3 = 66\%$	$2/3 = 66\%$

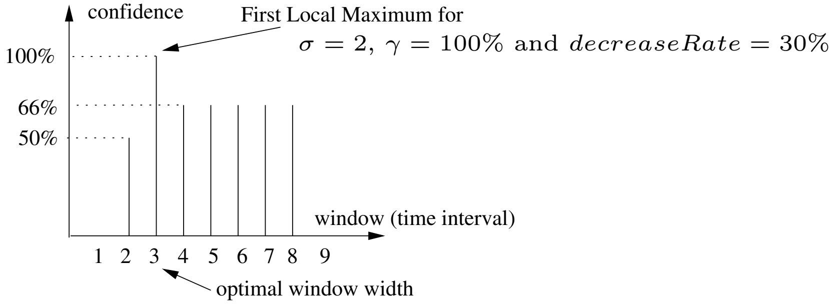


Fig. 3. Confidence and support for rule $A \rightarrow B \Rightarrow F$ in sequence S (Fig. 2), for $maxgap = 4$

5 Rule Selection

FLM-rules can be extracted using the *WinMiner* algorithm that is proposed in [13]. Extracted FLM-rules give us a description of the pump behavior. However, as we aim at predicting seizings, we only retain FLM-rules concluding on the symbol "seizing". Furthermore, we select a subset of these rules, the most predictive ones, in order to perform prognosis. Our selection process is inspired from the well known *leave-one-out cross validation* technique. It involves considering alternatively each subset Seq_i (see Fig. 1), ending with a failure, as a validation set while other subsets form the learning set. We thus alternate FLM-rules extractions (with the same parameter values) to get descriptions from the training set and validations of these descriptions by matching extracted FLM-rules. A FLM-rule is matched in the validation set if it occurs and if its occurrence time span is lower or equal to its optimal window width. Once our selection process ends, we get a set of FLM-rules that do not trigger any false alarm on validation sets. A same FLM-rule can be extracted at several iterations, each iteration providing a different optimal window width. In this case, its optimal window width is set to the most observed value. The set of selected FLM-rules is termed as the *FLM-base*. Back to our application, as seizings can originate from very different causes, and as we only have 13 subsets relating to a seizing, the minimum support threshold σ is set to 2. In order to extract the most confident rules, the minimum confidence threshold γ is set to 100%. Parameter *decreaseRate* is set to 30% to select pronounced/singular optimal window widths and the *maxgap* constraint is set to 1 week to consider very large optimal window widths. Indeed, when searching for the optimal window width of an episode rule, confidence and support measures are computed for window widths that are lower or equal to the number of events of the rule multiplied by the *maxgap* constraint. Finally, we asked for FLM-rules containing 4 event types as a maximum so as to consider generic rules and to make extractions tractable. For each extraction, we got about 3000000 FLM-rules. Among them 486 FLM-rules end on symbol seizing. At the end of the rule selection process, we got a FLM-base containing 29 FLM-rules with their respective optimal window widths. Optimal window widths distribution is presented in Figure 4. It clearly shows that setting a unique window width does not make sense when extracting episode rules. Running such a process remains tractable: execution times does not exceed 3 hours on a standard PC (proc. Intel Xeon CPU 5160 @ 3.00GHz, 3.9 Go ram, linux kernel 2.26.22.5).

6 Real Time Prognosis

Real time prognosis first relies on matching premisses of the 29 most predictive FLM-rules issued from the rule-selection step (Section 5). Then, for each matched premiss, a time interval within conclusions/seizings should occur is computed. Those aspects are detailed in Section 6.1. As many different premisses can be matched, we may get different time intervals for a single failure occurrence. We thus propose in Section 6.2 a method for merging these informations and for providing a single time interval, namely the *forecast window*.

6.1 Matching FLM-Rules Premisses in Datastreams

For matching premisses of the FLM-rules belonging to the FLM-base, we build a queue of event occurrences whose time span is lower than W, the largest optimal window width of the FLM-base. This queue is maintained by removing events

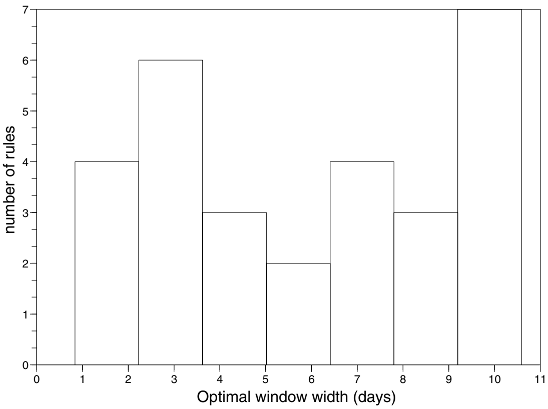


Fig. 4. Distribution of optimal window widths of the FLM-rule base

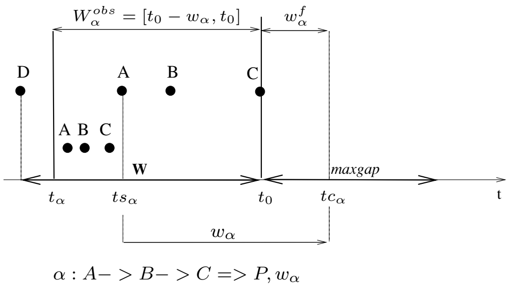
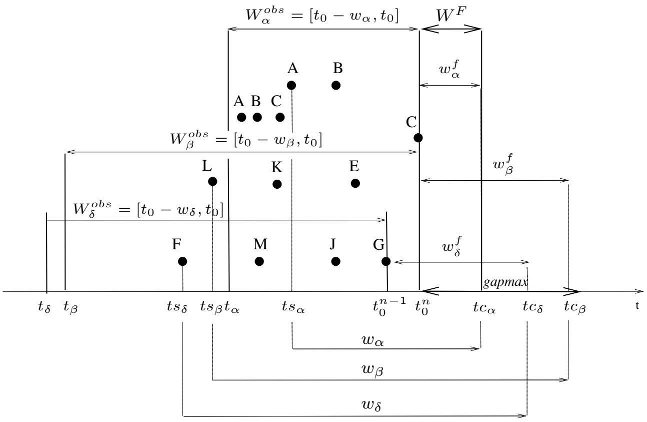


Fig. 5. Matching rule $A \rightarrow B \rightarrow C \Rightarrow P$

whose occurrence date t' is lower than $t - W$, with t the current system time. We thus make sure that enough data are kept for being able to identify the premisses of all the FLM-rules that form the FLM-base. Each time a new event occurs, it is added to the queue. If that one corresponds to the suffix (last event) of the premiss of a FLM-rule r belonging the the FLM-base, a premiss matching process is launched. We scan the queue through an observation window $W_r^o =]t_r, t_0[$ with t_0 the date at which rule matching process is launched and $t_r = t_0 - w_r$, with w_r the optimal window width of rule r. Indeed, in the worst case scenario, the conclusion of rule r is about to occur at $t_0 + 1$ and the earliest date of occurrence of the first symbol of its premiss is $t_0 + 1 - w_r = t_r + 1$. As proposed in [3], in this observation window, we search for the latest minimal occurrence of the premiss of r which amounts in finding the occurrence date of its first event, denoted ts_r and defined as follows: let Ts_r be the set of the occurrence dates of the first event of the premiss occurrences of rule r that occurs in $]t_r, t_0]$. The date ts_r is the single element in Ts_r such that $\nexists\ t \in Ts_r$ with $ts_r \neq t$ and $t > ts_r$ and such that the conclusion of rule r does not appear in $]ts_r, t_0]$. Then, by definition of



$\alpha : A- > B- > C => P, w_\alpha$

$\beta : L- > K- > E- > C => P, w_\beta$

$\delta : F- > M- > J- > G => P, w_\delta$

Fig. 6. Merging prediction information of FLM-rules

FLM-rules, we forecast the conclusion of rule r in $w_r^f =]t_0, tc_r]$ with $tc_r = ts_r + w_r$. By construction $t_0 < tc_r < t_0 + maxgap$. Figure 5 illustrates this process for rule $A \rightarrow B \rightarrow C \Rightarrow P$. In [11], for each matched premiss of rule r, its conclusion is forecasted at tc_r though it may appear in $]t_0, tc_r]$ by definition of FLM-rules. The prognosis approach proposed in [11] is thus not consistent with respect to the definition of FLM-rules.

6.2 Merging FLM-Rules Predictions

Let t_0 be the date at which a FLM-rule premiss is matched. For each matched premiss of rule r, its conclusion should occur in $w_r^f =]t_0, tc_r]$ with 100% confidence (if the minimum confidence is set to 100%). Let Tc be the set of all prediction dates tc_r that are active, i.e. that are greater than t_0 (those prediction dates can be computed before and at t_0). The associated failure prediction time interval $]t_s^f, t_e^f]$, also termed as the forecast window W^F, is such that $t_s^f = t_0 \wedge t_e^f = min(Tc)$. By choosing the min operator to aggregate prediction dates, this forecast window is defined to be the earliest one. Figure 6 provides a forecast window established using rules α, β, δ that have been recognized at t_0^{n-1} and t_0^n.

7 Experimental Evaluation of Prognosis

In order to evaluate the accuracy of our forecast, we consider two cases:

- each time our forecast method foresees a seizing, we check if seizing really occurs in the given forecast window $W^F =]t_s^f, t_e^f]$.

- each time t_0 a new event arises, and if no warning is triggered, we check if a seizing occurs in $]t_0, t_0 + maxgap]$. We extracted rules under $maxgap$, a maximum time gap between events. We thus can not predict any occurrence of conclusions of FLM-rules after $t_0 + maxgap$.

We applied the real time prediction method on 2 datasets: the 13 sequences used to build our FLM-base (dataset 1) and 21 new sequences of production data (dataset 2). Using these datasets, we simulated 2 data streams and made respectively 24125 and 32525 forecasts. Results of evaluations are given in Table 1 and in Table 2, using confusion matrices. We denote $\widehat{failure}$ the number of forecasts stating that pump will seize and $\widehat{healthy}$ the number of forecasts that do not foresee anything. Though we could access few data relating to failures so far, results are really encouraging as we predicted 10 seizings out of 13 with $99,97\%$ of accuracy on dataset 1 and as we foresaw 2 upcoming seizings with $98,75\%$ accuracy on dataset 2. The 262 misforecasts on dataset 1 all relate to the 3 failures that have not been detected. Furthermore, the 20 and 404 false alarms (tables 1 and 2) stating that pump will seize, were generated on pump that really seized few days later. Though the forecast window was not precise enough, diagnostic was right. Earliest failure predictions provided by our software prototype arise at the latest 3 hours before the seizing really occurs and, most of the time, more than 2 days before. This is enough to plan an intervention. In Section 5, we outlined that optimal window widths distribution shows different modes. They are all involved in forecast windows.Those results are good. The proposed approach is thus patent-pending and we are now deploying it for a client of the semi-conductor market.

Table 1. Results for dataset 1

	$\widehat{failure}$	$\widehat{healthy}$
failure	492	262
healthy	20	23351

Table 2. Results for dataset 2

	$\widehat{failure}$	$\widehat{healthy}$
failure	300	0
healthy	404	31821

8 Conclusion and Perspectives

In this paper, we present a local pattern-based approach for modeling pumping systems by means of FLM-rules and for forecasting failures using a subset of these rules, namely the FLM-base. We applied this approach in an industrial context in which vacuum pumping systems are running under severe and unpredictable conditions. Identified FLM-base is based on vibratory data which contain rich information but are difficult to handle. Indeed, as complex pumping systems may have high vibration levels even in good running conditions, the extraction of rules for failure predictions without false alarms has been a critical point. Results are encouraging as we forecast failures with a good accuracy, i.e. more than 98% on both learning data and new data. Moreover, using our predictions, enough time is left to technical teams for planning an intervention. The presented forecast method is patent-pending. Others applications to telecommunication networks, constant frequency rotating machines or supply chain management can be considered. Future work directions include introducing fuzzy logic to merge prediction dates so as to provide end-users with gradual warnings.

References

[1] Abu-Hanna, A., Lucas, P.J.F.: Prognostic models in medicine: AI and statistical approaches. Jr. of Methods of Information in Medicine 40, 1–5 (2001)

[2] Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of the 11th Intl. Conf. on Data Engineering, pp. 3–14 (1995)

[3] Cho, C., Zheng, Y., Chen, A.L.P.: Continuously Matching Episode Rules for Predicting Future Events over Event Streams. In: Dong, G., Lin, X., Wang, W., Yang, Y., Yu, J.X. (eds.) APWeb/WAIM 2007. LNCS, vol. 4505, pp. 884–891. Springer, Heidelberg (2007)

[4] Fiot, C., Masseglia, F., Laurent, A., Teisseire, M.: Gradual trends in fuzzy sequential patterns. In: Proc. of the 12th Intl. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU), pp. 456–463 (2008)

[5] Giannella, C., Han, J., Pei, J., Yan, X., Yu, P.S.: Mining frequent patterns in data streams at multiple time granularities. Next Generation Data Mining 212, 191–212 (2003)

[6] Grabill, P., Brotherton, T., Berry, J., Grant, L.: The us army and national guard vibration management enhancement program: data analysis and statistical results. In: Annual proceeding of American helicopter society, vol. 58, pp. 105–119 (2002)

[7] Hatonen, K., Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H.: TASA: Telecomunications Alarm Sequence Analyzer or: How to enjoy faults in your network. In: IEEE Network Operations and Management Symposium, pp. 520–529 (1996)

[8] Harding, J.A., Shahbaz, M., Kusiak, A.: Data mining in manufacturing: a review. Jr. of Manufacturing Science and Engineering 128(4), 969–976 (2006)

[9] Letourneau, S., Famili, F., Matwin, S.: Data mining for prediction of aircraft component replacement. IEEE Intelligent Systems and their Applications 14(6), 59–66 (1999)

[10] Lalanne, C.: Vibrations aléatoires. Hermes Science (1999)

[11] Le Normand, N., Boissiere, J., Meger, N., Valet, L.: Supply chain management by means of FLM-rules. In: 12th European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 29–36 (2008)

[12] Magoulas, G.D., Prentza, A.: Machine learning in medical applications. Jr. Machine Learning and Its Applications 2049, 300–307 (2001)

[13] Meger, N., Rigotti, C.: Constraint-based mining of episode rules and optimal window sizes. In: Chin, W.-N. (ed.) APLAS 2004. LNCS, vol. 3302, pp. 313–324. Springer, Heidelberg (2004)

[14] Mannila, H., Toivonen, H.: Discovering generalized episodes using minimal occurrences. In: Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining (KDD), pp. 146–151 (1996)

[15] Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. Jr. of Data Mining and Knowledge Discovery 1(3), 259–298 (1997)

[16] Raghavan, V., Hafez, A.: Dynamic data mining. In: Logananthara, R., Palm, G., Ali, M. (eds.) IEA/AIE 2000. LNCS (LNAI), vol. 1821, pp. 220–229. Springer, Heidelberg (2000)

[17] Schwabacher, M., Goebel, K.: A Survey of Artificial Intelligence for Prognostics. In: Working Notes of 2007 American Institute in Aeronautics and Astronautics Fall Symposium: AI for Prognostics (2007)

[18] Jens, T.B.: Mechanical Vibration and Shock Measurements. Bruel & Kjaer (1973)

Predicting Disk Failures with HMM- and HSMM-Based Approaches

Ying Zhao[1], Xiang Liu[2], Siqing Gan[2], and Weimin Zheng[1]

[1] Department of Computer Science and Technology
Tsinghua University
Beijing, China 100084
[2] School of Mathematical Sciences and Computing Technology
Central South University
Changsha, China 410075

Abstract. Understanding and predicting disk failures are essential for both disk vendors and users to manufacture more reliable disk drives and build more reliable storage systems, in order to avoid service downtime and possible data loss. Predicting disk failure from observable disk attributes, such as those provided by the Self-Monitoring and Reporting Technology (SMART) system, has been shown to be effective. In the paper, we treat SMART data as time series, and explore the prediction power by using HMM- and HSMM-based approaches. Our experimental results show that our prediction models outperform other models that do not capture the temporal relationship among attribute values over time. Using the best single attribute, our approach can achieve a detection rate of 46% at 0% false alarm. Combining the two best attributes, our approach can achieve a detection rate of 52% at 0% false alarm.

Keywords: Disk Failure, SMART data, Hidden Markov Model, Hidden Semi-Markov Model.

1 Introduction

Reliable storage systems serve as one of the fundamental key components for providing reliable performance in large enterprise systems. Although disk failfure is a rare event (lower than 1% annualized failure rate (AFR) reported by vendors [1] and as high as 6% AFR reported by users [2]), it could be very costly in terms of service downtime once it fails. As storage systems become more complex and can easily reach 1000 disks per node (e.g., NetApp FAS6000 series [3]), understanding and predicting failures have been a challenging task for both disk manufacturers (e.g., [1]) and users (e.g., [2]). An accurate failure prediction mechanism is desired to raise warnings to users and to reduce the cost caused by such failures.

The focus of disk failure studies has been on the rate it happens and the relationship between disk failures and observable factors. They can be categorized in two groups. The first group utilizes a broad range of information such as system logs, disk model, room temperature and so on ([2,4,3]), whereas the second

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 390–404, 2010.

group focuses specifically on information collected through the Self-Monitoring and Reporting Technology (SMART) system ([5,6,2]), which provides close monitoring on a number of attributes indicating the health and performance of disks, such as Read Error Rate, Throughput Performance, Seek Error Rate, Reallocated Sectors Count, and so on. The analytical methods on SMART data vary from theshold-based methods (provided by most manufacturers), Bayesian approaches [5], machine learning approaches [6], to correlation study over time [2]. Nevertheless, they all fail to fully consider the characteristics of the observed attributes over time as a time series and tend to make their predictions based on individual or a set of attribute values.

In this paper, we consider observed SMART attributes as time series and employ Hidden Markov Models (HMMs) and Hidden Semi-Markov Models (HSMMs) to build models for "good" and "failed" drives to make predictions. The motivation is that people have observed for some attributes, "a pattern of increasing attribute values (or their rates of change) over time is indicative of impending failure" [6]. It is reasonable to believe that attribute values observed over time are not independent, and a sequence of observed values with certain patterns may be a good indicator on whether or not a drive may fail soon. Hence we would like to fully explore the prediction power of SMART data when treated as time series. The reason we choose HMMs and HSMMs is that they provide flexible, general-purpose models for univariate time series, especially for discrete-valued series, and HSMM is more flexible and offers a large variety of possible temporal dependence structures. In addition to the proposed HMMs- and HSMMs-based methods, we also conduct experiments with the best methods reported in [6], namely the rank-sum test and SVMs. The experimental results show that our approaches improve the prediction accuracy over other approaches for both single- and multiple-attribute settings. Our proposed approaches do not require expensive parameter searching, and are able to reach a detection rate of 52% while keeping the false alarm rate as low as 0%.

The rest of the paper is organized as follows. We first formulate the problem and discuss basic data preparation techniques in Section 2. Section 3 discusses various prediction models for single attributes, including Hidden Markov Models, Hidden Semi-Markov Models, and Rank-sum, whereas Section 4 presents prediction models for multiple attributes, including for combining the results from multiple classifiers and Support Vector Machines. The description of the dataset used in our experiments is in Section 5, as well as experimental methodology and metrics. Section 6 presents the detailed experimental results. Related work is discussed in Section 7, and Section 8 concludes the entire paper.

2 Problem Formulation and Data Preprocessing

2.1 Problem Formulation

Given an attribute matrix $A_{T\times M} = \{a_{ij}\}$ of a disk drive, where a_{ij} is the value of attribute j measured at time i (T is the number of time intervals, M is the number of attributes), the problem of disk failure prediction is to predict whether the disk is going to fail based on $A_{T\times M}$. The attributes could come from a wide range of sources, including room temperature, hours of operating, system logs, and other physical features measured by the Self-Monitoring and Reporting Technology (SMART) system etc.

If the information of a set of failed disk drives and a set of good disk drives is available, the problem of disk failure prediction can be formulated as a classification problem. In particular, let failed disks be positive samples, good disks be negative samples, we can build classifiers based on attribute matrices from both positive and negative samples. When an attribute matrix of a new disk is available, we can use the trained classifiers to classify whether the disk belongs to the positive class (i.e., is going to fail), or the negative class (i.e., is good).

There are three ways to use attribute matrices for obtaining classifiers. First, the entire or part of each row of an attribute matrix can be extracted out as a vector (e.g., at time t, $V_t = \{a_{ti_1}...a_{ti_n}\}$, where $\{i_1, ..., i_n\}$ is a set of attributes of size n), and be used individually as a training sample. In this way, the relationship between attributes is taken into consideration, and this is how SVMs are trained. Second, the entire or part of each column of an attribute matrix can be extracted out as a sequence (e.g., for attribute i, $S_i = \{a_{t_1i}...a_{t_2i}\}$ is the attribute sequence from time t_1 to time t_2), and be used individually as a training sample. In this way, the sequence for an attribute is considered as a time series, and the temporal relationship within an attribute is taken into consideration. Our proposed methods in this paper fall into this category. Finally, time series of multiple attributes can be considered at the same time for training, however, it requires complicated learning and may only suitable for a small set of attributes and short period of time.

2.2 Data Preprocessing

Previous research has reported that using binning in data preprocessing is effective in disk failure prediction [5,6]. We employ the equal-width binning procedure is our study, where each attribute's range is divided into a fixed number of bins with equal-width, and attribute values become bin numbers.

3 Predition Models for Single Attributes

In this section, we discuss the two prediction models designed specifically for time series, and the rank sum test, which was reported as the best prediction model for single attributes by [6]. We first briefly describe Hidden Markov Models (HMMs) and Hidden Semi-Markov Models (HSMMs), and how they are used to build prediction models. Then, we briefly review the rank-sum test, and discuss how it is different in terms of using SMART data as time series.

The study of HMMs and HSMMs has a long history. The first HMM work was introduced by the paper of Baum and Petrie [7] in the mid-sixties, and the first application of HSMMs was analyzed in 1980 by Ferguson [8]. They both are built for two stochastic processes: an observed process and an underlying 'hidden' process. Since they provide flexible, general-purpose models for time series, people have been using them on discrete-valued series, categorical series, and many other types of data. The state duration probability density of HMMs is implicitly modeled and follows exponential distribution, which may not be appropriate for some applications. Whereas HSMMs explicitly set a state duration probability density, hence offer a large variety of possible temporal dependence structures.

3.1 Hidden Markov Models

The basic idea of HMM is that the system has an underlying 'hidden' process involving different states and transitions between them at each time t. Each state has probabilities to emit observable symbols, which correspond to the physical output of the system. Given N states $S = \{S_1, S_2, ..., S_N\}$ (the state at time t is denoted as q_t), M distinct observation symbols $V = \{v_1, v_2, ..., V_M\}$, and an observation sequence $O = O_1 O_2 ... O_T$ over time T, the transition probability distribution A, the observation probability distribution B, and the initial state distribution π form the complete parameter set of a model $\lambda = (A, B, \pi)$. In particular, the transition probability distribution $A = \{a_{ij}\}$ is defined as

$$a_{ij} = P[q_{t+1} = S_j | q_t = S_i], 1 \leq i, j \leq N. \tag{1}$$

The observation probability distribution $B = \{b_{jk}\}$ is defined as

$$b_{jk} = P[O_t = v_k | q_t = S_j], 1 \leq j \leq N, 1 \leq k \leq M. \tag{2}$$

Finally, the initial state distribution $\pi = \{\pi_i\}$ is defined as

$$\pi_i = P[q_1 = S_i], 1 \leq i \leq N. \tag{3}$$

Once λ is determined, we can use the Forward-Backward procedure to calculate the likelihood of observing sequence O from the model. Also, when a set of observation sequences are available, λ can be estimated by the Baum-Welch method or other methods [9].

To apply HMMs to the disk failure prediction problem, N is determined through an automatic model selection procedure, which will be discussed in Section 5. M is the number of bins used in data preparation. We build HMMs from positive training sequences (failed disks) and negative training sequences (good disks), respectively. Then, the testing data (disks without labels) are evaluated using the two models and sequence log likelihood values are calculated accordingly. Finally, we take the difference between the two log likelihood values for each testing sample, and a threshold value is selected to make the prediction (i.e., if the difference is greater than the threshold, the testing disk is predicted as failed, otherwise, it is predicted as good). The threshold value is adjusted to vary the tradeoff between true positive rates and false positive rates.

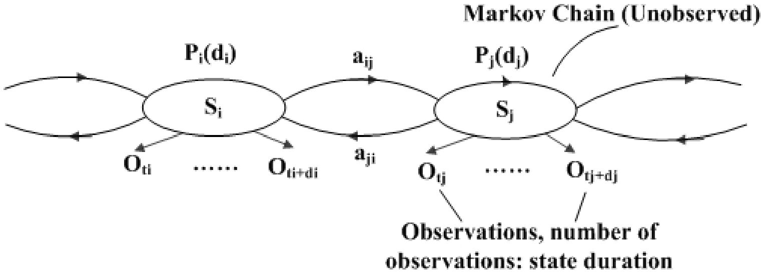

Markov Chain (Unobserved)
$P_i(d_i)$
a_{ij}
$P_j(d_j)$
S_i
S_j
a_{ji}
O_{ti} ...... O_{ti+di}
O_{tj} ...... O_{tj+dj}
Observations, number of observations: state duration

Fig. 1. Illustration of Hidden Semi-Markov Model with specified state duration distributions $p_i(d)$, where multiple observations can associate with one state

3.2 Hidden Semi-markov Models

Letting $a_{ii} = 0, 1 \leq i \leq N$ and setting an state duration probability density, Hidden Semi-Markov Models (HSMMs) allow the underlying 'hidden' process stay in a state for a while, which is a reasonable modification for problems like failure detection, where failures may happen in stages, and the duration of each stage may not follow exponential distributions as implicitly assumed in HMMs.

An illustration of the basic structure of a HSMM is shown in Fig. 1. The complete parameter set for HSMMs now becomes $\lambda = (A, B, D, \pi)$, where the state duration distribution $D = \{p_j(u)\}$ is defined as

$$p_j(u) = P[q_{t+u+1} \neq j, q_{t+u-v} = j, v = 0, ..., u-2 | q_{t+1} = j, q_t \neq j], 1 \leq j \leq N. \tag{4}$$

Sequence likelihood calculation and model parameter estimation can be solved similarly by modifying the Forward-Backward procedure and the Baum-Welch method, respectively [9].

Similarly, we apply HSMMs to the disk failure prediction problem, where N is determined through an automatic model selection procedure, which will be discussed in Section 5. M is the number of bins used in data preparation. We build HSMMs from positive training sequences (failed disks) and negative training sequences (good disks), respectively. Then, the testing data (disks without labels) are evaluated using the two models and sequence log likelihood values are calculated accordingly. Finally, we take the difference between the two log likelihood values for each testing sample, and a threshold value is selected to make the prediction (i.e., if the difference is greater than the threshold, the testing disk is predicted as failed, otherwise, it is predicted as good). The threshold value is adjusted to vary the tradeoff between true positive rates and false positive rates.

3.3 Rank-Sum Test

The Wilcoxon-Mann-Whitney rank-sum is one of the models for comparison against our prediction models, as it was reported as the best model when using single attributes [6]. The rank-sum test [10] is used for testing whether two datasets come from the same distribution. In particular, it is assumed that attribute values measured from good disks follow a "good" distribution, attribute values measured from failed disks follow a "about-to-fail" distribution, and the two distributions are different. Hence, given a reference set R (of size n) consisting of attribute values from good disks and a testing set T (of size m) consisting of attribute values from disks without labels, the rank-sum test statistic W_S can be calculated and used to test against the null hypothesis, i.e., R and T are from the same distribution. We follow the same calculations in performing this test as in [6], where more details about this test can be found.

Note that although a sequence of consecutive attribute values of the test data is used as T for the rank-sum test, it is not used as a sequence in the sense that the testing attribute values are sorted together with reference points with respect to the value, and the temporal dependence among the testing attribute values is ignored.

4 Prediction Models for Multiple Attributes

In this section, we discuss the strategies of combining individual prediction models trained from single attributes. We also briefly discuss Support Vector Machines at the end, which was reported as the best model for multiple attributes in [6].

4.1 Combining Multiple Classifiers

There is a rich literature on the subject of combining multiple classifiers (refer to [11] for a detailed review). Classifiers trained from different training data, feature sets, and classification methods are used together to provide better classification. In this paper, we adopt a simple fixed rule for this purpose, namely, the maximum rule. More complicated combining strategies involving training the combiner can also be considered, however, we do not discuss them here and leave them to future work.

Given M binary classifiers, for one testing sample x, each classifier i returns some sort of confidence value $c_i^+(x)$ for assigning x to the positive class and $c_i^-(x)$ for assigning x to the negative class. In our case, $c_i^+(x)$ and $c_i^-(x)$ are the sequence log likelihoods observed from the positive and negative model, respectively. In general, $c_i^+(x) - c_i^-(x) > 0$ ($c_i^+(x) - c_i^-(x) < 0$) means the testing sample is more likely from the positive (negative) class. Now we have the simple strategy to combine these values.

the maximum rule: The combined value $C(x)$ is defined as

$$C(x) = \max_{i=1}^{M}\{c_i^+(x) - c_i^-(x)\}. \tag{5}$$

The intuitive behind the maximum rule is that we would like to assign x to the positive class (i.e., predicted as failed in our case) if one attribute shows great confidence in doing so. Once we have the combined values, a threshold is used again to trade off between detections and false alarms.

4.2 Support Vector Machines

Support vector machine (SVM) [12] is a state-of-the-art classification technique based on pioneering work done by Vapnik et al. This algorithm is introduced to solve two-class pattern recognition problems using the Structural Risk Minimization principle. Given a training set in a vector space, this method finds the best decision hyperplane that separates two classes. The quality of a decision hyperplance is determined by the distance (referred as margin) between two hyperplances that are parallel to the decision hyperplance and touch the closest data points of each class. The best decision hyperplane is the one with the maximum margin. By defining the hyperplane in this fashion, SVM is able to generalize to unseen instances quite effectively. The SVM problem can be solved using quadratic programming techniques. SVM extends its applicability on the linearly non-separable data sets by either using soft margin hyperplanes, or by mapping the original data vectors into a higher dimensional space in which the data points are linearly separable through appropriate kernel functions. A new example is classified by representing the point the feature space and computing its distance from the hyperplane.

SVM has been applied to a wide range of classification problems because of its many attractive features, including effective avoidance of overfitting, and the ability to handle large feature spaces. The success of SVM has been showed in documents classification and secondary structure predictions. It was also reported as the best classification method on disk failure data when using 25 features from SMART data [6].

5 Datasets and Experimental Methodology

5.1 Dataset

We use the SMART dataset provided by Center for Magnetic Recording Research, University of California, San Diego [6]. The original dataset contains 178 “good” drives and 191 “failed” drives. For each drive, an entry of 60 SMART attributes was recorded every 2 hours for a total of 600 hours. However, failed drives may not survive the entire recording period, thus may have fewer than 300 entries (fewer than 10 entries for some failed drives are observed). In our experiments, since both HMMs and HMMs require non-trivial sequence lengths to be effective, we selected failed drives with more than 50 entries resulting in a total of 99 failed drives. The number 50 is a tradeoff between providing more sequential information to HMMs/HSMMs and keeping enough number of failed drives to fairly compare our results with others. We do not think a minimum length requirement for detecting disk failure is a major limitation, as in practice, the prediction analysis is invoked along with a continuous disk monitoring. It is a safe assumption that before a disk fails, we have already met the length requirement.

In short, our SMART dataset contains 178 good drives and 99 failed drives. We use this dataset for our prediction models and other models for comparison, i.e., the rank-sum test and SVMs. Comparing this dataset with the one used in [6], there are fewer failed disks, making the dataset smaller and more unbalanced (i.e., the ratio between positive and negative class samples is more skewed).

5.2 Metrics

We use receiver operating characteristic (ROC) curves as the metric for evaluating various prediction approaches in our experiments. As mentioned in Section 2, we would like to predict or classify disks with failures accurately through our models. An ROC curve displays the tradeoff between true positive rate and false positive rate. The true positive rate, in our case termed as **detection rate**, is defined as the fraction of positive examples predicted correctly by the model, i.e., the fraction of disks with failures predicted correctly. The false positive rate, in our case termed as **false alarm rate**, is defined as the fraction of negative examples predicted as a positive class, i.e., the fraction of "good" disks wrongly predicted as disks with failures. A good classification model should be as close as possible to the upper left coner of the diagram. Since high false alarm rates are prohibited as the consequent overhead is not acceptable by users (manufacturers' recommendation is 0.1%-0.3%), we only plot ROC curves up to a false alarm rate of 5%, and focus our discussions at the low end (i.e., a false alarm rate less than 1%).

5.3 Experimental Methodology

Now we describe how various models are applied to predict disk failures and how our experiments are set up.

HMMs and HSMMs: The number of distinct observation symbols is the number of bins, which is set to be 10 in our experiments, as suggested by [5,6]. The number of states is determined by an automatic model selection process, in which five models are trained with the number of states varying from 10 to 30, and the one that maximizes the sequence log likelihoods of training sequences is selected. For an attribute, the positive models are trained with sequence segments of the last 50 values of the failed disks in the training set, whereas the negative models are trained with sequence segments of the last 50 values of the good disks in the training set. A disk is predicted as failed if any of its sequence segments of consecutive 50 values over time is predicted as failed. Using a sequence segment of length 100 does not seem to improve performance. In addition, to simply training, we use a parametric state duration distribution instead of non-parametric state duration distribution for HSMMs, i.e., a kernel including a normal distribution is assumed.

We evaluate HMMs and HSMMs in the following way. We randomly selected one fifth of the total failed and good disks as the training sets for positive and negative models, respectively, and use the remaining disks as the testing set. By varying the threshold of the difference of sequence log likelihoods calculated over positive and negative models, we can plot an ROC curve for the trained models.

We repeated this process 5 times, and the average ROC curve was calculated and presented in the following section.

Rank-sum test: We followed the same implementation as in [6]. In particular, for single attribute testing, we randomly choose 50 values from good disks as the reference set (i.e., $n = 50$). Sequence segments consisting of consecutive 15 values of a disk are used as the testing set (i.e., $m = 15$). A disk is predicted as failed if any of its sequence segments over time is predicted as failed by the rank-sum test.

SVMs: We used a 5-fold cross validation for evaluating SVMs. The training samples are attribute vectors from failed disks (positive samples) and good disks (negative samples). A disk is predicted to be failed if any of its attribute vectors over time is predicted as positive. We used mySVM [13] to perform the experiments and followed the same parameter selection as in [6].

6 Experimental Results

6.1 Single Attributes

In the first set of experiments, we focus on the prediction performance of various models when using single SMART attributes. In particular, we started from the 25 attributes identified by [6] with reverse arrangements and z-score test that appear to be promising for distinguishing good and failed disks. We then run our HMM and HSMM predictors and found four attributes provided good failure detection, namely, ReadError18, Servo2, Servo10, and FlyHeight7. We show the ROC curves of HMM, HSMM, and the rank-sum test on these four attributes in Fig. 2 to Fig. 5. Note that ReadError18 and Servo10 were also among the best attributes that appeared to provide good failure detection when using the rank-sum test [6].

The detection rates obtained by the rank-sum test on attribute FlyHeight7 and Servo2 are either very low or too low to be measured, which is consistent with Murray et al.'s observation [6]. On the other two attributes, ReadError18 and Servo10, the HMM and HSMM outperformed the rank sum test significantly. Especially for Servo10, there are no measurable detections at a false alarm less than 1% for the rank-sum test, while HMM and HSMM can achieve a detection rate of 46% and 30% at 0% false alarm, respectively. Except for ReadError18, HMM outperformed HSMM at low false alarm rates. Possible reasons for this observation are two-fold: 1) HSMMs require more parameters to be estimated and our training samples are too limited to produce good training results; 2) to simplify training, we use a parametric state duration distribution, i.e., a kernel including a normal distribution is assumed. Using other distribution families might improve the performance of HSMMs, and we will explore more options in the future.

Note that the performance of the rank-sum test for attribute ReadError18 and Servo10 is slightly different from that reported in [6]. For example, for Servo10,

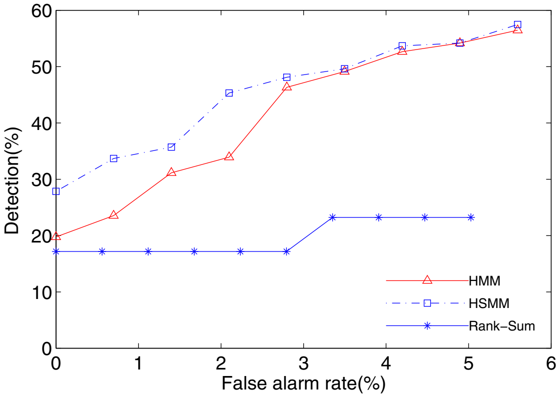


Fig. 2. Performance of HMM, HSMM, and the rank-sum test on disk failure prediction using attribute ReadError18. Results of HMM and HSMM are average over 5 trails.

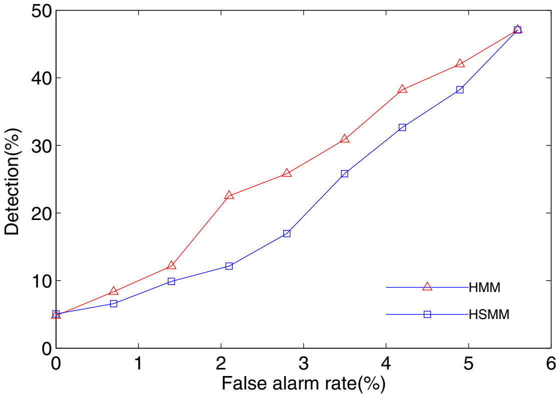


Fig. 3. Performance of HMM, HSMM, and the rank-sum test on disk failure prediction using attribute FlyHeight7. Results of HMM and HSMM are average over 5 trails. No measured detection rates for the rank-sum test with false alarms $\leq 5\%$.

we report a detection rate of 34% and 36% at a false alarm rate of 1% and 2.5%, respectively, while Murray et al. [6] reported a detection rate of 30% and 45%, respectively. There are two reasons behind this observation. Firstly, the reference set of 50 data points is randomly selected from a large population of data points, hence, it is inherently prohibited to reproduce the exact results because of the possible different choice of the reference set. Secondly, our dataset contains fewer failed disks than the one used in [6] as mentioned in Section 5. Nevertheless, we consider the comparison with the rank-sum test has been made carefully, and the conclusion drawn from the comparison is a fair conclusion.

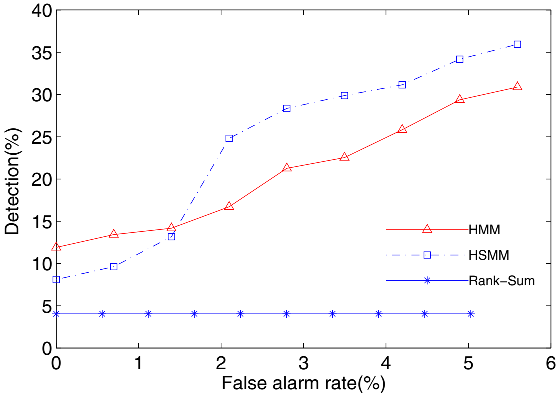


Fig. 4. Performance of HMM, HSMM, and the rank-sum test on disk failure prediction using attribute Servo2. Results of HMM and HSMM are average over 5 trails.

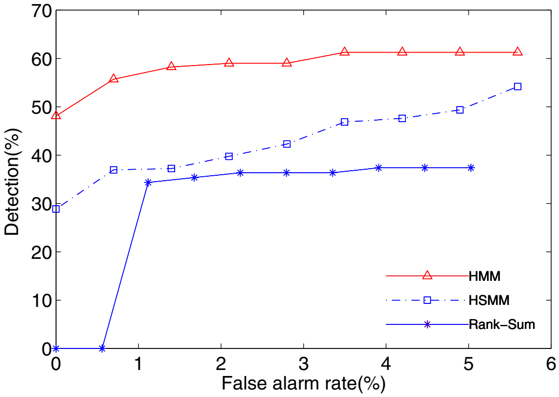


Fig. 5. Performance of HMM, HSMM, and the rank-sum test on disk failure prediction using attribute Servo10. Results of HMM and HSMM are average over 5 trails.

6.2 Multiple Attributes

In the second set of experiments, we focus on the prediction performance of various models when using multiple SMART attributes. From the best single attributes, we choose to combine the best two, namely, ReadError18 and Servo10, which can achieve high detection rates at 0% false alarm. We also compare our model with a SVM trained on 25 attributes. As reported by [6] training SVM on 25 features gave the best multiple-attribute prediction model. We show the ROC curves of the combined HMM using the two attributes and SVM trained on 25 features in Fig. 6.

From Fig. 6, we can see that our combined model with two attributes can achieve a detection rate of 52% at 0% false alarm, and this result is better than the SVM result on 25 attributes. Again, since the dataset used in our experiments is smaller and more unbalanced than the one used in [6], the results of SVM are lower than those reported in [6]. Also there is a huge parameter space to search for finding the best SVM model, and varying parameters like $C, L+, L-$ to trade off between detections and false alarms is not intuitive and hard to interpret. In contrast, our HMM- and HSMM-base approaches employ an automatic model selection process with a much smaller parameter searching space, and varying the threshold of log likelihoods provides an intuitive way to trade off between detections and false alarms.

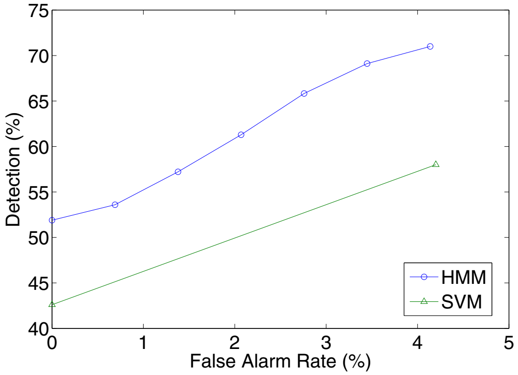


Fig. 6. Performance of the combined HMM and SVM on disk failure prediction. Results of HMM are average over 5 trails. Results of SVM are average over 5-fold cross validation.

6.3 Run Time

All experiments were conducted on a desktop with Pentium 4 dual 2.4GHz CPUs and 1GB memory. We summarize the training and testing time for each prediction model in Table 1. For HMM and HSMM, the single model training time shown in the table is the average for obtaining one model of the positive class or negative class when using one attribute, while the testing time is for calculating log likelihoods of all sequence segments for a testing disk for one model averaged over all testing data. As we mentioned before, HMM and HSMM prediction models have a self model selection procedure, i.e., the number of states are varied for 5 times and the best model is chosen based on the sequence likelihoods of the training data. Hence, obtaining the combined predictor with two attributes involves training 2 × 2 × 5 models (a positive model and a negative model per attribute, 2 attributes, repeated 5 times for model selection) and the combining time is trivial, which results in roughly 1 hour in total. Once we have

Table 1. Summary of run times (in seconds) for training and testing

Prediction Model	Training	Testing (per testing disk)
HMM (singel)	192.4	0.04
HSMM (single)	191.8	0.47
HMM (combined)	3848.4	0.16
Rank-sum	-	2.82
SVM (25 attributes)	1279.1	1.64

the combined predictor, testing one disk needs to be done against 4 individual models, which results in 0.16 seconds for HMM. The testing time of HSMM is much longer than HMM, since we use a normal distribution kernel that demands an integral calculation.

Similarly, the training time for SVM (roughly 20 mins) is for obtaining a SVM model using 25 attributes given one set of parameters. However, to find the optimal set of parameters, an extensive search in the parameter space (i.e., training more than 100 models with different sets of parameters) needs to be done and can be very costly. Note that the rank-sum has the advantage of no training needed; however, testing is a non-trivial process and needs to be repeated for many times with different sequence segments for a testing disk.

7 Related Work

People have been studying disk failures for decades and have been focusing on the relationship between disk failures and observable factors with the hope to predict disk failures accurately and manufacture more reliable disks. These work can be categorized in two groups. The first group utilizes a broad range of information such as system logs, disk model, room temperature and so on ([2,4,3]), whereas the second group focuses specifically on information collected through the Self-Monitoring and Reporting Technology (SMART) system ([5,6,2]), which are more related to our work. Hence we review them in more detail in this section.

Hamerly and Elkan [5] studied SMART data collected from 1934 "good" drives and 9 "failed" drives. They employed supervised naive Bayes learning and mixtures of naive Bayes models to predict "failed" drives, and were able to achieve a detection rate of 30% at a false alarm rate of 0.2%.

Murray et al. [6] collected data from 369 drives (roughly half of which are "good" drives), and studied the performance of multiple-instance naive Bayes, Support Vector Machines (SVMs), unsupervised clustering, the reverse arrangements test, and the rank-sum test on these data. They found that the rank-sum test was the best for single and small sets of SMART attributes (52.8% detection with 0.7% false alarm) and SVMs performed the best for all features (50.6% detection with 0% false alarm).

The work of Pinheiro et al. [2] was conducted on data collected from a system health infrastructure on a large population of hard drives under deployment within Google's computing infrastructure. Their work involved more than one hundred thousand disk drives of different types and periodic SMART data were extracted and cleaned up for their analysis. They found strong correlations between disk failures and some SMART parameters, such as first erros in reallocation, offline reallocations, and probational counts etc. However, they believed that the prediction accuracy based on SMART parameters alone may be quite limited.

HMMs and HSMMs first showed their great success in the context of automatic speech recognition ([9],[8]). Since then for more than two decades, people have been using HMMs and HSMMs on more and more fields involving signal processing applications, such as biological sequences, financial time series and so on. Recently, Salfner and Malek [14] used HSMMs to analysis system logs for accurate error-based online failure prediction on a commercial telecommunication system, where they treat system logs as time series. Along with sequence clustering and noise filtering, their HSMM-based approach was able to achieve a failure prediction F-measure of 0.66.

8 Conclusion

In this paper, we tackle the problem of disk failure prediction from a different angle. We consider various attributes measured at consecutive time intervals for a disk drive as time series, and use HMMs and HSMMs to model such time series to classify "failed" disks and "good" disks. Our proposed prediction models can achieve a detection rate of 46% and 52% at 0% false alarm for single- and multiple-attributes, respectively, which confirms that using time series of attributes is indeed effective in predicting disk failures. Moreover, our combined model does not require expensive parameter searching, and provides an intuitive way of trading off between detections and false alarms.

Acknowledgments. This work was funded by the National Science Fundation China (Project No. 60703058) and the National High-tech R&D Program (863 Program) No. 2008AA01A204 and 2008AA01A201.

References

1. Cole, G.: Estimating Drive reliability in Desktop Computers and consumer electronics systems. Tech. Rep., Seagate Technology Paper TP-338.1 (2000)
2. Pinheiro, E., Weber, W., Barroso, L.A.: Failure Trends in a Large Disk Drive Population. In: 5th USENIX Conference on File and Storage Technologies (FAST 2007), Berkeley, CA (2007)
3. Jiang, W., Hu, C., Zhou, Y., Kanevsky, A.: Are Disks the Dominant Contributor for Storage Failures? A Comprehensive Study of Storage Subsystem Failure Characteristics. ACM Transactions on Storage 4(3), Article 7 (2008)
4. Bairavasundaram, L.N., Goodson, G.R., Pasupathy, S., Schindler, J.: An Analysis of Latent Sector Errors in Disk Drives. SIGMETRICS Perform. Eval. Rev. 35(1), 289–300 (2007)

5. Hamerly, G., Elkan, C.: Bayesian Approaches to Failure Prediction for Disk Drives. In: 18th International Conference on Machine Learning (ICML 2001), Williamstown, MA (2001)
6. Murray, J.F., Hughes, G.F., Kreutz-Delgado, K.: Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. Journal of Machine Learning Research 6, 783–816 (2005)
7. Baum, L.E., Petrie, T.: Statistical Inference for Probabilistic Functions for Finite State Markov Chains. Annals of Mathematical Statistics 37, 1554–1563 (1966)
8. Ferguson, J.D.: Variable Duration Models for Speech. In: Symposium on the Application of Hidden Markov Models to Text and Speech, Princeton, New Jersey, pp. 143–179 (1980)
9. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. IEEE Transactions on Information Theory 77(2), 257–284 (1989)
10. Lehmann, E.L., D'Abrera, H.J.M.: Nonparametrics: Statistical Methods Based on Ranks. Prentice Hall, Upper Saddle River (1998)
11. Duin, R.P.W.: The Combining Classifier: to Train or not to Train? In: 16th international conference on Pattern Recognition, Quebec City, Canada, pp. 765–770 (2002)
12. Vapnik, V.: Statistical Learning Theory. John Wiley, New York (1998)
13. http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM
14. Salfner, F., Malek, M.: Using Hidden Semi-Markov Models for Effective Online Failure Prediction. In: 26th IEEE Symposium on Reliable Distributed Systems (SRDS 2007), Beijing, China, pp. 161–174 (2007)

Aircraft Engine Health Monitoring Using Self-Organizing Maps

Etienne Côme[1], Marie Cottrell[1], Michel Verleysen[2], and Jérôme Lacaille[3]

[1] SAMM - Universit Paris 1 Panthon-Sorbonne
90, rue de Tolbiac, 75013 Paris, France
{etienne.come,marie.cottrell}@univ-paris1.fr

[2] Université Catholique de Louvain, Machine Learning Group
Place du levant 3, 1348 Louvain-La-Neuve, Belgium
michel.verleysen@uclouvain.be

[3] Snecma, Rond-Point Ren Ravaud-Rau,
77550 Moissy-Cramayel CEDEX, France
jerome.lacaille@snecma.fr

Abstract. Aircraft engines are designed to be used during several tens of years. Ensuring a proper operation of engines over their lifetime is therefore an important and difficult task. The maintenance can be improved if efficients procedures for the understanding of data flows produced by sensors for monitoring purposes are implemented. This paper details such a procedure aiming at visualizing in a meaningful way successive data measured on aircraft engines. The core of the procedure is based on Self-Organizing Maps (SOM) which are used to visualize the evolution of the data measured on the engines. Rough measurements can not be directly used as inputs, because they are influenced by external conditions. A preprocessing procedure is set up to extract meaningful information and remove uninteresting variations due to change of environmental conditions. The proposed procedure contains three main modules to tackle these difficulties: environmental conditions normalization (ECN), change detection and adaptive signal modeling (CD) and finally visualization with Self-Organizing Maps (SOM). The architecture of the procedure and of modules are described in details in this paper and results on real data are also supplied.

Keywords: Health monitoring, Self-Organizing Maps, Changes detection.

1 Introduction

During the flights, some on-board sensors measure many parameters related to the behavior (and therefore the health) of aircraft engines. These parameters are recorded and used at short and long terms for immediate action and alarm generation, respectively. In this work, we are interested in the long-term monitoring of aircraft engines and we want to use these measurements to detect any deviations from a "normal" behavior, to anticipate possible faults and to facilitate the maintenance of aircraft engines.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 405–417, 2010.

This work presents a tool that can help experts, in addition to their traditional tools based on quantitative inspection of some relevant variables, to easily visualize the evolution of the engine health. This evolution will be characterized by a trajectory on a two-dimensional Self-Organizing Map. Abnormal aging and fault will result in deviations with respect to normal conditions.

The choice of Self-Organizing Maps is motivated by several points:

- SOMs are useful tools for visualizing high-dimensional data onto a low-dimensional grid;
- SOMs have already been applied with success for fault detection and prediction in plants and machines (see [1] for example).

The article is organized as follow. First, in Section 2, the data and the notations used throughout the paper are presented. The methodology and the global architecture of the proposed procedure are described in Section 3. Each step is defined and results on real data are given in Section 4. Finally, in Section 5, perspectives on trajectory analysis are supplied.

2 Data

Measurements are collected on a set of I engines. On each engine $i \in \{1, \ldots, I\}$, n_i measurements are performed successively flight after flight; there is thus no guarantee that the time intervals between two measures are approximately equal. Each observation is denoted by Z_{ij}, where $i \in \{1, \ldots, I\}$ is the engine number and $j \in \{1, \ldots, n_i\}$ is the flight number.

Each vector Z_{ij} contains two kinds of variables: those which are strictly related to the behavior of the engine (fuel consumption, static pressure, ...), and those which are related to the environment (temperature, altitude, ...). Let the p engine variables be denoted by $Y_{ij}^1, \ldots, Y_{ij}^p$ and the q environmental variables by $X_{ij}^1, \ldots, X_{ij}^q$. Each observation is therefore a $(p+q)$-vector Z_{ij}, where $Z_{ij} = [Y_{ij}, X_{ij}] = \left[Y_{ij}^1, \ldots, Y_{ij}^p, X_{ij}^1, \ldots, X_{ij}^q\right]$. The variables at disposal are listed in Table 1. There are $p = 5$ engine variables and $q = 15$ environmental variables. The dataset contains measurements for approximately one year of flights and $I = 91$ engines, that leads to a global dataset with $\sum_{i=1}^{91} n_i = 59407$ $(p+q)$-dimensional observations.

3 Methodology

The goal is to build the trajectories of all the engines, that is to project the successive observations of each engine on a Self-Organizing Map, in order to follow the evolution and to eventually detect some "abnormal" deviation.

It is not valuable to use the rough engine measurements: they are inappropriate for direct analysis by Self-Organizing Maps, because they are strongly dependent on environment conditions and also on the characteristics of the engine (its past, its age, ...).

Table 1. Variables names, descriptions and type

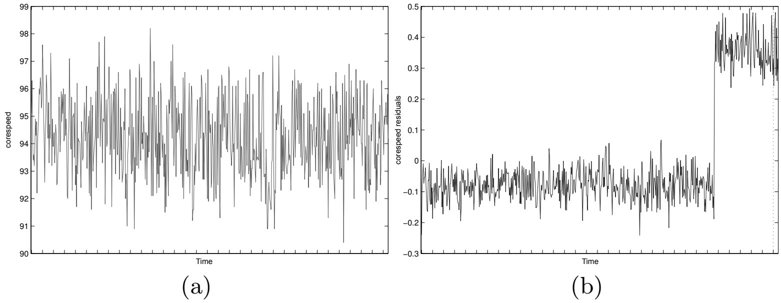
	Name	**Description**	**Type**	**Binary**
	aid	aircraft id		
	eid	engine id		
	fdt	flight date		
X_{ij}^1	temp	temperature	environment	
X_{ij}^2	nacelletemp	nacelle temperature	environment	
X_{ij}^3	altitude	aircraft altitude	environment	
X_{ij}^4	wingaice	wings anti-ice	environment	✓
X_{ij}^5	nacelleaice	nacelle anti-ice	environment	✓
X_{ij}^6	bleedvalve	bleed valve position	environment	✓
X_{ij}^7	isolationleft	valve position	environment	✓
X_{ij}^8	vbv	variable bleed valve position	environment	
X_{ij}^9	vsv	variable stator valve position	environment	
X_{ij}^{10}	hptclear	high pressure turbine setpoint	environment	
X_{ij}^{11}	lptclear	low pressure turbine setpoint	environment	
X_{ij}^{12}	rotorclear	rotor setpoint	environment	
X_{ij}^{13}	ecs	air cooling system	environment	
X_{ij}^{14}	fanspeedi	N1	environment	
X_{ij}^{15}	mach	aircraft speed	environment	
Y_{ij}^1	corespeed	N2	engine	
Y_{ij}^2	fuelflow	fuel consumption	engine	
Y_{ij}^3	ps3	static pressure	engine	
Y_{ij}^4	t3	temperature plan 3	engine	
Y_{ij}^5	egt	exhaust gas temperature	engine	

The first idea is to use a linear regression for each engine variable: the environmental variables (real-valued variables) and the number of the engine (categorical variable) are the predictors and the residuals of these regressions can be used as standardized variables (see [2] for details). For each engine variable $r = 1, \ldots, p$, the regression model can be written as:

$$Y_{ij}^r = \mu^r + \alpha_i^r + \lambda_1^r X_{ij}^1 + \ldots + \lambda_q^r X_{ij}^q + \epsilon_{ij}^r \tag{1}$$

where α_i^r is the engine effect on the r^{th} variable, $\lambda_1^r, \ldots, \lambda_q^r$ are the regression coefficients for the r^{th} variable, μ^r is the intercept and the error term ϵ_{ij}^r is the residual.

Figure 1 presents for example the rough measurements of the *corespeed* feature as a function of time (for engine 6) and the residuals computed by model (1). The rough measurements seem almost time independent on this figure, whereas the residuals exhibit an abrupt change which is linked to a specific event in the life of this engine. This simple model is therefore sufficient to bring to light interesting aspects of the evolution of this engine. However, the signals may contain ruptures, making the use of a single regression model hazardous.



(a) (b)

Fig. 1. (a) Rough measurements of the **corespeed** variable as a function of time for engine 6, (b) residuals of the same variable and for the same engine using a simple linear model with the environmental variables and the engine indicator as predictors (see Table 1)

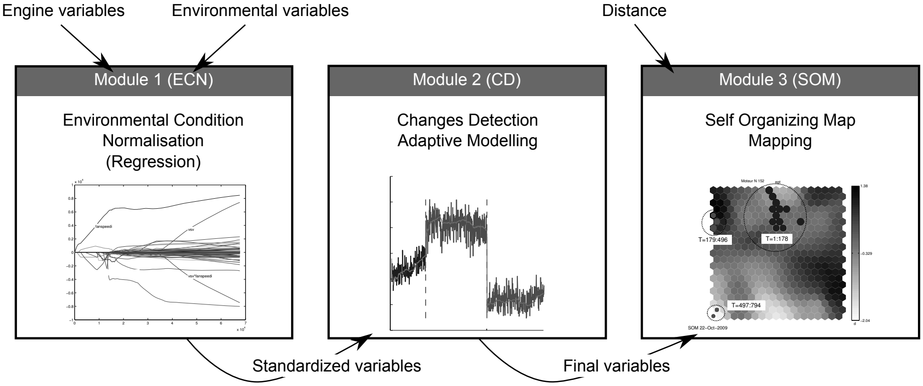


Fig. 2. Global architecture of the health monitoring tools

The main idea of this work is to replace model (1) by a new procedure which deals with the temporal behavior of the signals. The goal is therefore to detect the ruptures and to use different models after each rupture.

This new procedure is composed of two modules. The first module (Environmental Conditions Normalization, ECN) aims at removing the effects of the environmental variables to provide standardized variables, independent of the flight conditions. It is described in section 4.1.

The second module uses an on-line change detection algorithm to find the above mentioned abrupt changes, and introduces a piecewise regression model. The detection of the change points is done in a multi-dimensional setting taking as input all the normalized engine variables supplied by the ECN module. The Change Detection (CD) module is presented in Section 4.2.

Finally, as a result of these first two steps, the “cleaned” database can be used as input to a Self-Organizing Map with a “proper” distance for trajectories visualization. The third module (SOM) provides the "map" on which the trajectories will be drawn.

This three-steps procedure is summarized in Figure 2.

4 Description of the Three Modules

4.1 Environmental Conditions Normalization - ECN

The first module aims at removing the effects of the environmental variables.

For that purpose, one regression model has to be fitted for each of the p engine variables. As the relationship between environmental and engine variables is complex and definitively not linear, the environmental variables can be supplemented by some non-linear transformations of the latter, increasing the number of explanatory variables. Interactions (all the possible products between two environmental variables), squares, cubes and fourth powers of the non binary environmental variables are considered. The number q of predictors in the model is therefore a priori equal to $(11 + 4) * (11 + 4 - 1)/2 = 105$ for the interactions variables and $11 * 4 + 4 = 48$ for the power of the continuous variable and the binary variables leading to a total of $q = 153$ predictors.

This number is certainly too large and some of them are clearly irrelevant due to the systematic procedure used to build the non-linear transforms of environmental variables.

A LASSO criterion [3] is therefore used to estimate the regression parameters and to select a subset of significant predictors. This criterion can be written using the notations from Section 2 for one engine variable Y^r, $r \in \{1, \dots, p\}$ as:

$$\boldsymbol{\beta}^r = \arg \min_{\boldsymbol{\beta}^r \in \mathbb{R}^q} \sum_{i,j=1}^{I,n_i} \left(Y_{ij}^r - \sum_{l=1}^{q} \beta_l^r \, X_{ij}^l \right)^2 , \sum_{l=1}^{q} |\beta_l^r| < C^r \qquad (2)$$

The regression coefficients are penalized by a L_1 penalty which forces some of them to be null for a well chosen value of C^r. The LARS algorithm [3] is used to estimate all the solutions of the optimization problem (2) for all possible values of C^r. The optimal value of C^r with respect to the prediction error estimated by cross-validation (with 20 blocs) is finally selected. The number of selected predictors and the coefficient of determination R^2 are listed in Table 2 for all engine variables. Engine variables are well explained by the proposed models as attested by the high value of the coefficients of determination.

Table 2. Number of selected predictors and coefficients of determination for all engine variables

	corespeed	*fuelflow*	*ps3*	*t3*	*egt*
nb vars	25	43	31	30	41
R^2_{obs}	0.9875	0.9881	0.9773	0.9636	0.8755

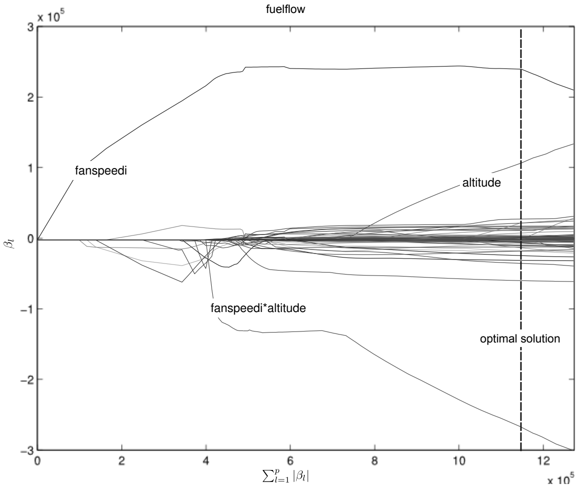


Fig. 3. Regularization path for the *fuelflow* variable: regression coefficients evolution with respect to C^r. The more significant explanatory variables are given and the best solution with respect to cross-validation is depicted by a vertical line.

A qualitative inspection of the model results was also carried out with the help of engine experts. The regularization path plot (as shown in Figure 3) is very interesting from the point of view of the experts, because it can be compared with their previous knowledge. Such a curve clearly highlights which are the more relevant predictors and they appear to be in very good adequateness with the physical knowledge on the system.

In summary, the first preprocessing module (ECN) provides 5 standardized engine variables, which are independent of environmental conditions. These new variables still contain some significant aspects such as linear trends and abrupt changes at specific dates. We therefore propose to use an on-line Change Detection algorithm (CD) together with an adaptive linear model to fit the data.

4.2 Change Detection - CD

To take into account the two types of variation (linear trend and abrupt changes), we implement an algorithm based on the ideas from [4] and [5]. The solution is based on the joint use of an on-line change detection algorithm to detect abrupt changes and of a bank of recursive least squares (RLS) algorithms to estimate the slow variations of the signals. The algorithm works on-line in order to allows projecting new measurements on the map as soon as new data are available.

The method can be described as follows :

1) One RLS algorithm is used for each one of the p engine variable to recursively fit a linear model. At each date l, for each standardized engine variable r, for each engine i, one has to minimize the following criterion:

$$(\alpha_l^r, \beta_l^r) = \arg \min_{\alpha\in\mathbb{R},\beta\in\mathbb{R}} J(\{Y_{i1}^r, \dots Y_{il}^r\}, \alpha, \beta) \tag{3}$$

$$= \arg \min_{\alpha\in\mathbb{R},\beta\in\mathbb{R}} \sum_{j=1}^{l} \lambda^{(l-i)} (Y_{ij}^r - (\beta.j + \alpha))^2 \tag{4}$$

The estimates α_l^r and β_l^r are respectively the intercept and the slope of the linear relationship. These estimates are then used to remove the slow variations of the signals by defining the quantity:

$$\varepsilon_l^r = Y_{il}^r - (\beta_l^r.l + \alpha_l^r) \tag{5}$$

2) These values are then computed for each standardized engine variable and concatenated in a vector $\boldsymbol{\varepsilon}_l = [\varepsilon_l^1, \dots, \varepsilon_l^p]$, which is then used in a multi-dimensional Generalized Likelihood Ratio (GLR) algorithm [6] to detect the abrupt changes of the signals. The GLR algorithm is a sequential test procedure based on the following model :

$$\boldsymbol{\varepsilon}_k \sim \mathcal{N}_p(\theta(k), \Sigma), \forall k > 0,$$

with :

$$\theta(k) = \begin{cases} \theta \in \Theta_0 & \text{si } k < t_0, \\ \theta \in \Theta_1 & \text{si } k \geq t_0, \end{cases} \tag{6}$$

where t_0 is the unknown change time and Θ_0 and Θ_1 are two non-overlapping subsets. They are defined by two hyper-spheres centered on θ_0 as shown in Figure 4. In such a case, Θ_0 and Θ_1 are defined by:

$$\Theta_0 = \left\{\theta : (\theta - \theta_0)^t \Sigma^{-1} (\theta - \theta_0) \leq a^2\right\} \tag{7}$$

$$\Theta_1 = \left\{\theta : (\theta - \theta_0)^t \Sigma^{-1} (\theta - \theta_0) \geq b^2\right\}, \tag{8}$$

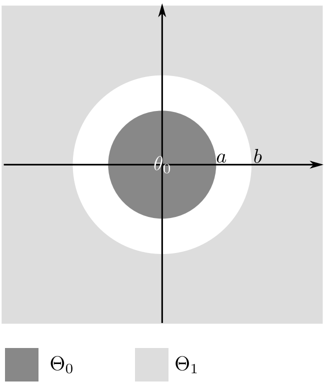
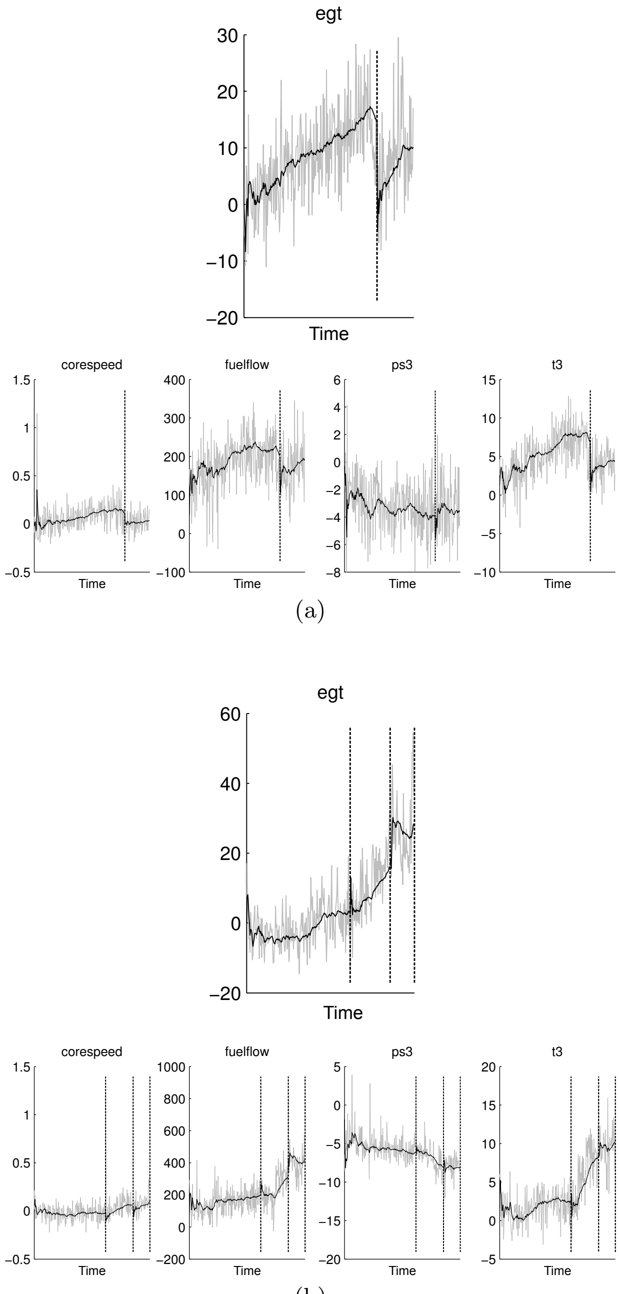


Fig. 4. Subsets for θ_1 and θ_0



(a)

(b)

Fig. 5. Change detection results for engines 2 and 41. Alarms are depicted by vertical lines, input signals are shown in light gray and signal estimates using RLS are depicted by a black line. One variable *egt* is bigger than the other to present more clearly the RLS estimate of the signal.

with $a \leq b$. The sequential test problem is then solved by defining the alarm date t_a to be the first time where the test statistic g_k is above a specified threshold h:

$$t_a = \min\{k \geq 1 : g_k \geq h\}, \tag{9}$$

with the following definition of the test statistic:

$$g_k = \max_{1 \leq j \leq k} \ln \frac{\sup_{\theta:||\theta-\theta_0||_\Sigma \geq b} \prod_{i=j}^{k} p_\theta(\varepsilon_i)}{\sup_{\theta:||\theta-\theta_0||_\Sigma \leq a} \prod_{i=j}^{k} p_\theta(\varepsilon_i)}. \tag{10}$$

where $p_\theta(.)$ denotes a density of a multivariate Gaussian of mean θ and variance Σ.

With S_j^k given by:

$$S_j^k = \ln \frac{\sup_{\theta:||\theta-\theta_0||_\Sigma \geq b} \prod_{i=j}^{k} p_\theta(\varepsilon_i)}{\sup_{\theta:||\theta-\theta_0||_\Sigma \leq a} \prod_{i=j}^{k} p_\theta(\varepsilon_i)}, \tag{11}$$

the maximization problem (10) has an analytical solution in the Gaussian case and S_j^k takes the following value :

$$S_j^k = \begin{cases} -\frac{(k-j+1)}{2}(\chi_j^k - b)^2, \text{ if } \chi_j^k < a, \\ \frac{(k-j+1)}{2}\left(-(\chi_j^k - b)^2 + (\chi_j^k - a)^2\right), \text{ if } a \leq \chi_j^k \leq b, \\ +\frac{(k-j+1)}{2}(\chi_j^k - a)^2, \text{ if } \chi_j^k \geq b, \end{cases} \tag{12}$$

with $\chi_j^k = \sqrt{(\bar{\varepsilon}_j^k - \theta_0)^t \Sigma^{-1} (\bar{\varepsilon}_j^k - \theta_0)}$ and $\bar{\varepsilon}_j^k = \sum_{i=j}^{k} \frac{1}{(k-j)} \varepsilon_i$.

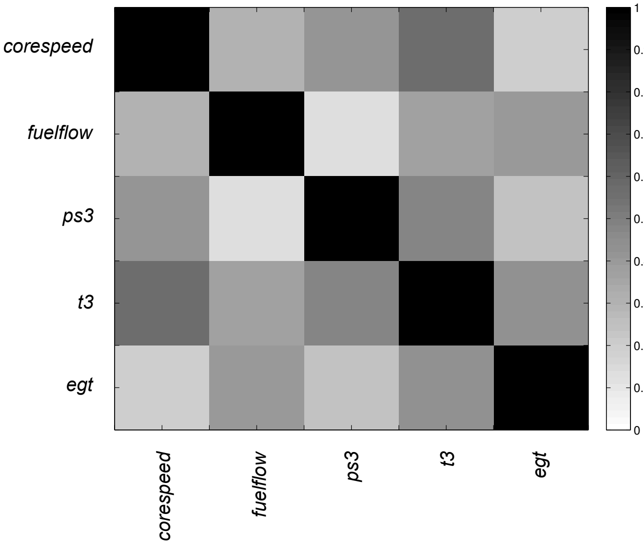


Fig. 6. Correlations between input variables

3) Finally, when an alarm is sent by the GLR algorithm, all the RLS algorithms are re-initialized.

The results supplied by this algorithm are the following :

- the alarm dates supplied by the multi-dimensional GLR algorithm;
- cleaned signals estimated by the RLS algorithm;
- slopes and intercepts estimated by the RLS algorithm.

Figure 5 presents the obtained results for two engines. One abrupt change was found for the first engine and 3 for the second; all of them seem to be reasonable and a comparison between estimated alarm dates and recorded real events of the engine life have confirmed this fact. The estimated signals are also shown on these two figures.

4.3 Self-Organizing-Maps - SOM

The cleaned signals provided by the previous two modules are then used as input to a SOM for visualization purpose. A [20 × 20] square SOM is defined to

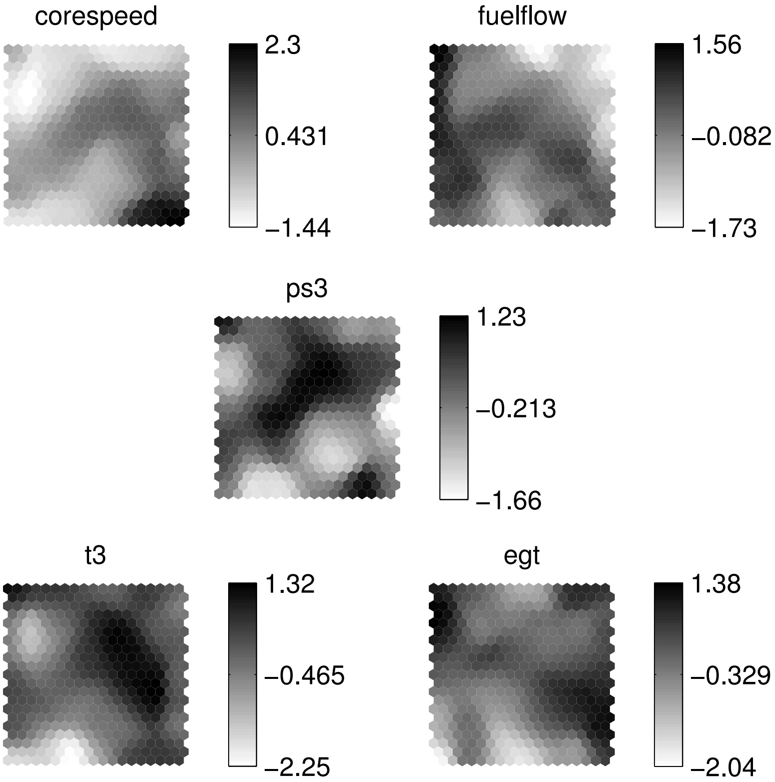
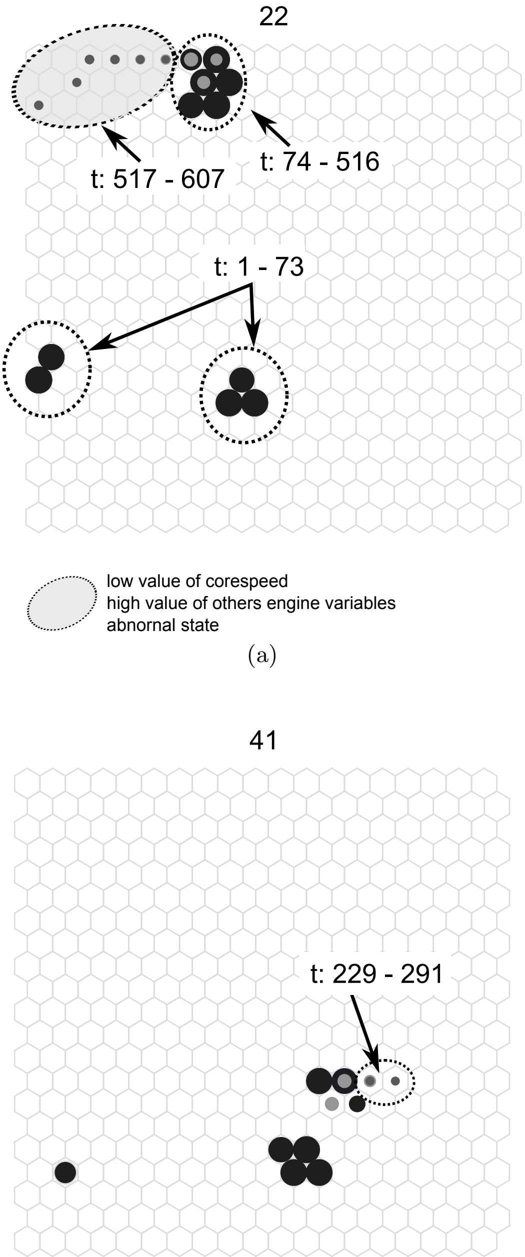


Fig. 7. Visualization of engine variable as map background; each cell of the map is colored according to the value of the selected variable



(a)

(b)

Fig. 8. Trajectories of engines 22 (a) and 41 (b) on the map. The sizes of the dots are proportional to the measurement date : smallest dots correspond to recent measurements, larger dots to older measurements. The colors correspond to the segments found by the change detection algorithm ($t_a = 394$ and 522 for engine 22; $t_a = 180$, 249 and 291 for engine 41).

project the observations. The Matlab toolbox [7] was used to implement it and the distance was carefully chosen since the standardized engine variables are very correlated as shown by the correlation matrix in Figure 6: several correlation coefficients have an absolute value greater than 0.6. A Mahalanobis distance is therefore used to whiten the data. The observations are normalized and a classical learning scheme is used to train the map.

Figure 7 shows the map colored according to the values of the five engine variables. It is clearly visible that the organization of the map is successful (all variables are smoothly varying on the map).

Figure 8 presents two examples of engine trajectories on the map, which clearly have different shapes. For the first engine, available maintenance reports inform us that this engine suffers from an deterioration of its high pressure core. This fault is visible on the map at the end of the trajectory: the engine which was projected on the middle north of the map during a large part of its trajectory, suddenly moves towards the nord-east corner of the map. This area of the map furthermore correspond to anomalous values of the engine variables as shown by the component plane representation of the map (see Figure 7). The second engine which is affected by another fault has also a trajectory which is interesting even if the interpretation is less obvious. It moves to an area characterized by a high value of the *exhaust gas temperature* which is known to be an indicator of possible trouble.

The visual representations on the Kohonen map provide a synthetic and meaningful representation for the temporal evolution of the engines. The next step is then to characterize the different shapes of trajectories, to define a suitable distance measure between these trajectories, and to define typical behaviors related to typical faults. Further work will consist in defining a proper distance between two trajectories (or parts of trajectories) in order to propose a request-type access to the database. Taking a piece of trajectory as input, the system will finally be able to recover the most similar trajectories of the database. Such similar trajectories can then be used to predict the possible evolution of the monitored engine. *Edit-type* distance widely used in biostatistics could be of interest for this task.

5 Conclusion and Perspectives

The method proposed in this paper is a useful tool to summarize and represent the temporal evolution of an aircraft engine health flight after flight. The regression approach taken to deal with the problem of environmental condition normalization seem to be effective. The joint use of an adaptive algorithm to estimate signal evolution (RLS) and of a change points detection method (GLR) is also an interesting solution to deal with the non-stationarity of the signals. Finally, Self-Organizing Maps can be used to show the engine health evolution in a synthetic manner.

References

1. Svensson, M., Byttner, S., Rgnvaldsson, T.: Self-organizing maps for automatic fault detection in a vehicle cooling system. In: 4th International IEEE Conference on Intelligent Systems, vol. 3, pp. 8–12 (2008)
2. Cottrell, M., Gaubert, G., Eloy, C., François, D., Hallaux, G., Lacaille, J., Verleysen, M.: Fault prediction in aircraft engines using self-organizing maps. In: Advances in Self-Organizing Maps, vol. 5629, pp. 37–44. Springer, Heidelberg (2009)
3. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.J.: Least angle regression. Annals of Statistics 32(2), 407–499 (2004)
4. Gustafsson, F.: Adaptative filtering and change detetction. John Wiley & Sons, Chichester (2000)
5. Ross, G., Tasoulis, D., Adams, N.: Online annotation and prediction for regime switching data streams. In: Proceedings of ACM Symposium on Applied Computing, March 2009, pp. 1501–1505 (2009)
6. Basseville, M., Nikiforov, I.: Detection of Abrupt Changes: Theory and Application. Prentice-Hall, Englewood Cliffs (1993)
7. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: Som toolbox for matlab 5. Technical Report A57, Helsinki University of Technology (April 2000)

Finding Temporal Patterns in Noisy Longitudinal Data: A Study in Diabetic Retinopathy

Vassiliki Somaraki[1,2], Deborah Broadbent[2,3],
Frans Coenen[1], and Simon Harding[2,3]

[1] Dept. of Computer Science, The University of Liverpool, Liverpool L69 3BX, UK
[2] Ophthamology Research Unit, School of Clinical Science,
The University of Liverpool, Liverpool L69 3GA, UK
[3] St. Pauls Eye Unit, Royal Liverpool University Hospital, L7 8XP, UK
{V.Somaraki,D.M.Broadbent,coenen,sharding}@liverpool.ac.uk

Abstract. This paper describes an approach to temporal pattern mining using the concept of user defined temporal prototypes to define the nature of the trends of interests. The temporal patterns are defined in terms of sequences of support values associated with identified frequent patterns. The prototypes are defined mathematically so that they can be mapped onto the temporal patterns. The focus for the advocated temporal pattern mining process is a large longitudinal patient database collected as part of a diabetic retinopathy screening programme, The data set is, in itself, also of interest as it is very noisy (in common with other similar medical datasets) and does not feature a clear association between specific time stamps and subsets of the data. The diabetic retinopathy application, the data warehousing and cleaning process, and the frequent pattern mining procedure (together with the application of the prototype concept) are all described in the paper. An evaluation of the frequent pattern mining process is also presented.

Keywords: Temporal Pattern Mining, Trend Mining.

1 Introduction

This paper describes an approach to finding temporal patterns in noisy longitudinal patient data. The identification of patterns in such data has many applications. One common example is the analysis of questionnaire returns collated over a number of years, for example Kimm et al. studied the nature of physical activity in groups of adolescents ([9]) and Skinner et al. studied children's food eating habits ([15]). Another example of the application of longitudinal studies is in the analysis of statistical trends; an early reported example is that of Wagner et al. [17], who performed an extensive longitudinal study of children with "special educational needs". Longitudinal studies particularly lend themselves to the analysis of patient data in medical environments where records of a series of "consultations" are available. For example Yamaguchi et al. ([18]) studied

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 418–431, 2010.

the effect of treatments for shoulder injuries, and Levy et al. [10] studied the long term effects of Alzheimer's disease. The application domain, with respect to this paper, is the longitudinal diabetic retinopathy screening data collected by The Royal Liverpool University Hospital (RLUH), a major centre for retinopathy research. The nature of the longitudinal data is of interest because it does not fit into any standard categorisation of such data, in that the "time stamp" used is the sequential patient consultation event number. The duration between consultations is also variable.

The temporal patterns of interest, in the context of this paper are frequent patterns (collections of attributes that appear together regularly) that feature some prescribed change in their frequency between two or more time stamps (i.e. a trend). For example patterns whose frequency increases/decreases overtime, patterns whose frequency remains constant with time, or patterns that display some other kind of trend. The patterns themselves are identified using a modified frequent pattern mining algorithm: the TFP algorithm [2,3] is used in this study, however alternative frequent pattern miners could be suitably modified. The proposed temporal pattern mining process is described in detail.

A further challenge of the work described is that the data collection is extremely large and complex; 150,000 records, comprising some 450 fields (of various types: categorical, quantitative, text, etc.), distributed over two databases each composed of a number of tables. The main challenge represented by the data was that, unlike more standard longitudinal data sets, there was no clear association between specific time stamps and subsets of the data. The data warehousing process established to prepare the data for mining is therefore also described. A further complication was that the data, in common with similar patient data sets, was very "noisy" in that it contained many missing and anomalous data. This issue was addressed by defining a set of *logic rules*. In the context of missing data the logic rules were used to derive appropriate values. In the case of anomalous data the logic rules were used to derive additional fields to formulate "consensus values". The data cleaning and warehousing process is also described in detail.

The principal contributions made by this paper may be summarised as follows:

1. A process for identifying temporal patterns in large longitudinal data sets over a sequence of time stamps.
2. An illustration of the application of the technique to a "real life" data set (including the data cleaning process required to facilitate this illustration).
3. The use of logic rules to address the joint issues of missing and anomalous data.

The rest of this paper is organised as follows. Further detail of the retinopathy application domain is given in Section 2. Some background with respect to longitudinal data mining, the concept of temporal pattern mining and the issue of missing values in data is given in Section 3. The adopted data warehousing and cleaning process is described in Section 4. The temporal pattern mining problem and its resolution is described in 5. An evaluation of the process is presented in

Section 6. A summary of the work described, the main findings and some closing observations are presented in section 7.

2 Diabetic Retinopathy Screening

The Royal Liverpool University Hospital (RLUH) has been a major centre for retinopathy research since 1991. Retinopathy is a generic term used to describe damage to the retina of the eye which can, in the long term, lead to visual loss. Retinopathy can result from a number of causes, for example: diabetes, age-related macular degeneration (AMD), high blood pressure and various genetic conditions. In diabetes the retinopathy progresses over a number of years through well characterised stages. Treatment comprises the application of laser to the retina and is most effective during the stages before vision is affected. Screening programmes for people with diabetes have recently been established across the UK to detect retinopathy and instigate prompt treatment.

RLUH has collected a substantial amount of data, over a considerable period, of time as part of its diabetic retinopathy research and screening programme. Screening takes place within the community and is conducted by technicians who perform photography and record data images on "lap-tops" which are then downloaded (typically) at the end of each day. Retinal images are graded at a central grading facility at a separate time, but within a few weeks, with results recorded into a database. If the level of disease detected in the retinal photographs is worse than a predetermined level, or if photographs are ungradable or unobtainable, then screenees are invited to a dedicated hospital outpatient clinic for further examination by an ophthalmologist using more specialised slit lamp biomicroscopy[1]. Data on retinal findings are entered into the database. This clinical assessment can occur several months after the initial photographic screening.

Four types of data associated with a single screening sequence are collected:

1. General demographic data.
2. Data on visual acuity (clarity of vision).
3. Data from grading of retinal images.
4. Data from biomicroscopy of the retina.

The full screening sequence is referred to as a "screening episode"

People with diabetes are usually screened once a year with the option to rescreen early (typically 6 months) depending on the presence of intermediate levels of disease indicating greater risk of progression. The RLUH screening programme currently deals with some 17,000 people with diabetes registered with family doctors within the Liverpool Primary Care Trust[2] per year. Overall details of some 20,000 patients have been recorded. Consequently a substantial amount of data is available for analysis. Some further details of the data collection are presented in the following sub-section.

[1] A high intensity light source instrument to facilitate examination of the human eye.

[2] A Primary Care Trusts (PCTs) are organisational units established to manage local health services in the UK.

2.1 Data Storage

Data collected from the diabetic retinopathy screening process described above is stored in a number of databases. The structure (tables) of these database reflect the mechanism whereby patients are processed and includes historical changes in the process. Screening commenced in 1991 when data was recorded in a bespoke database system called Epi-Info. Epi-Info was replaced with a more sophisticated system, Diabolos, in 1995, which describes the data used in this study. Diabolos, in turn, was replaced with a national database system, Orion, in 2005. The design and implementation of Orion does not lend itself to simple extraction of data for temporal pattern mining purposes and thus the data contained in this latest database system does not form part of the current study. Thus the study described here deals with data collected from 1995 to 2005.

The RLUH, as opposed to the screening programme, also maintains a clinical investigations database called ICE. This database includes information about biochemical “risk factors” that are known to be associated with progression of diabetic retinopathy. Not all patients included in the screening programme have records on ICE. The screening programme has its own Risk Factors database, maintained by the programme team, containing data mostly extracted from ICE.

In the context of temporal pattern mining there are therefore five tables used in this study of which the first four are held in the Diabolos system:

1. **Patient Details.** Table containing background information regarding individual patients.
2. **General.** Demographic patient details and visual acuity data.
3. **Photo Details.** Results from the photographic grading.
4. **Biomicroscopy.** Results from the slit lamp biomicroscopy in cases where this has been conducted.
5. **Risk Factors.** Results from blood pressure and biochemistry investigations known to be associated with an increased risk of progression of retinopathy.

3 Previous Work

This previous work section comprises three subsections, each focussing on one of the three Knowledge Discovery in Data (KDD) research domains encompassed by the work described in this paper: (i) longitudinal data mining, (ii) temporal pattern mining, and (iii) missing and anomalous data.

3.1 Longitudinal Data Mining

Longitudinal data is information comprising values for a set of data fields which are repeatedly collected for the same object over a sequence of sample points, as such it can be said to track the progress of the object in some context. The exemplar longitudinal data set is patient data, where information concerning a patient’s condition is repeatedly collected so as to track the patient’s progress.

Longitudinal data may be categorized in a number of ways, one suggested categorization is that of Singer and Willet [14] who identify *person-level* and *person-period* data sets. In a person-level data set each person (subject) has one record and multiple variables containing the data from each sampling. In a person-period data set each person (subject) has multiple records, one for each measurement occasion. Thus person-level data set has as many records as there are subjects in the sample, while a person-period data sets has many more records (one for each subject sampling event). The former is sometimes referred to as a *broad data structure* and the later as *long data structure* [16].

Longitudinal studies have variations regarding sample size, number of variables and number of time stamps. Broadly speaking, there are five main types of longitudinal study based on these characteristics [6]: (i) simultaneous cross-sectional studies, (ii) trend studies, (iii) times series studies, (iv) intervention studies and (v) panel studies. The work described in this paper may be described as a time series study, in order to identify trends contained in a person-period data set.

3.2 Temporal Pattern Mining

The objective of temporal pattern mining (or trend mining) is to discover temporal patterns in time stamped data. For example Nohuddin et al. [13] investigate the application of trend mining in cattle movement databases. With respect to diabetic retinopathy data the objective of the temporal pattern mining is to identify unexpected, previously unknown, trends in the data. However, the identification of known patterns is also seen as important as this would provide a means of validating the adopted approach. The process of frequent patten mining in static data tables is well established within the Knowledge Discovery in Data (KDD) community and can be traced back to early work on Association Rule Mining (ARM) as first espoused by Agrawal and Srikant [1]. Less attention has been applied to temporal pattern mining. There has been reported work on Temporal ARM (TARM) where association rules are mined from time stamped data.

The temporal pattern mining process described in this paper operates on binary value data sets (thus, where necessary, data must be transformed into this format using a process of normalisation and discretisation). The research described in this work also borrows from the field of Jumping and Emerging Patten (JEP) mining as first introduced by Dong and Li ([4]). The distinction between the work on JEPs, and that described in this paper, is that JEPs are patterns whose frequency increases (typically) between two data sets (although some work has been done on identifying JEPs across multiple data sets, for example Khan et al. [8]). JEP mining is usually also conducted in the context of classification (see for example [5]). The distinction between JEPs and the work described here is that the work is directed at pattens that change in a variety of pre-described ways over a sequence of data sets. To the best knowledge of the authors there is little reported work on temporal pattern mining or trend mining as defined above.

Zhu et al. [19], in the context of data stream mining, identify three processing models for temporal pattern mining: (i) Landmark, (ii) Damped and (iii) Sliding Windows. The Landmark model discovers all frequent patterns over the entire history of the data from a particular point in time called the "landmark". The Damped model, also known as the Time-Fading model, finds frequent patterns in which each time stamp is assigned a weight that decreases with "age" so that older records contribute less than more recent records. In the Sliding Window model the data is mined by sliding a "window" through the temporal dimension. A similar categorisation may be adopted with respect to temporal pattern mining. The work described in this paper adopts the Landmark model.

3.3 Missing and Anomalous Data

The problem of missing attribute values is well established in the context of data mining. The generally agreed view is that removing records with missing data is the least favoured option as this may introduce bias. The reduction of the overall data set size, by removing records that contain missing values, is not considered to be critical. There is significant scientific work to support this view. Approaches to the imputation of missing values has been extensively researched from a statistical perspective [7,11,12]. Example imputation methods include: nearest neighbour imputation, mean imputation, ratio imputation and regression imputation. The approach to missing data advocated in this paper is to define and implement a set of logical rules to address the missing value problem, this is discussed further in the following section.

4 Data Warehousing and Cleaning

For the study described in this paper, before any investigation of temporal pattern mining could commence the five database tables identified in Section 2 (Patient, General, Photo Details, Biomicroscopy and Risk factors) were combined into a single warehouse (i.e. a static data repository specifically intended for the application data mining and data analysis tools). The creation of the data warehouse required data anonimisation and data cleaning.

The anonimisation of the data tables was initiated by removing patient names. Although this was straightforward, this presented a second problem as in many cases the patient name was the "common key" linking database tables. An obvious candidate for a universal common key was patient NHS (National Health Service) numbers, however this was missing with respect to some 8000 records and consequently had to be added manually. The NHS number was then used for the construction of the data warehouse; on completion the NHS number was replaced by a sequential record number so that individual records could not be traced back to individual patients.

The next step after anonimisation was data cleaning. There were three principal issues to be addressed:

1. Missing values.
2. Contradictory values.
3. Duplicate records.

The first two issues were addressed by developing a set of *logic rules*. With respect to missing values the evidence of such a missing value could be interpreted in two ways: (i) that the value was either unknown or mistakenly omitted at time of collection, or (ii) the missing value indicated a negative response to a question suggested by the field, or (iii) the clinician considered the field to be inapplicable for the given case. For example some fields indicated responses to question such as "does the patient have one eye", to which, in many cases, the clinician had inserted a "yes" if the answer to the question was an affirmative and left the field blank otherwise (the latter can thus be interpreted as either a "no", or a "don't know"). A set of "if ... then ..." logical rules were therefore developed to address this issue. The logic rules were written in such a way that they could also be used for data validation purposes. The operation of these rules is best illustrated using some examples.

Consider the field *SeeGPRegularly*[3] featured in the Diabolos General Table. This field can have three possible values: 1 ("No"), 2 ("Yes") and 9 ("Don't know"). In the event of a missing value for this field may be derived from another field, in the set of database tables, *LastSeeGP*; asking when the patient last saw their GP for anything. The *LastSeeGP* field can have the following values: 1 ("Within last 6 months"), 2 ("Within last 6 to 12 months"), 3 ("More than a year ago") and 9 ("Don't know"). The logic rule is then as shown in Table 1 (the *null* value indicates a missing field). The rule states that if the value for *SeeGPRegularly* is missing and the value for *LastSeeGP* is also missing, or set to 9 ("Don't know"), we set the value for *SeeGPRegularly* to 9. If the patient has seen their GP with the last 12 months (*LastSeeGP* field set to 1 or 2) we set the value for *SeeGPRegularly* to 2 ("Yes"). Otherwise we set the value of *SeeGPRegularly* to 1.

Table 1. *SeeGPRegulary* Logic Rule

```
if ($SeeGPRegulary == null$) {
        if ($LastSeeGP == 9$) or ($LastSeeGP == null$) then ($SeeGPRegulary = 9$)
        if ($LastSeeGP == 1$) or ($LastSeeGP == 2$) then ($SeeGPRegulary = 2$)
        if ($LastSeeGP == 3$) then ($SeeGPRegulary = 1$)
        }
```

With respect to contradictory/anomalous values this issue can be exemplified by the *diAgeDiag* field, the age of the patient when diabetes was first diagnosed. Within the application domain this has been recognised as a question patients find very difficult to answer, and consequently clinicians responsible for gathering

[3] In the UK GP stands for "General Practitioner", essentially a family doctor; so the field is asking if the patient sees their doctor regularly.

data often leave this field blank if they feel that a patient is unable to give a definitive answer. In addition it was found that patients may give a different answer over different consultations which was believed to get less accurate with the passing of time. The rule adopted in this case was to take the first recorded value of the field as this was likely to be the most accurate.

The duplicate records issue, only prevalent in the Risk factor table, was addressed by issuing search queries to identify duplicate records and then manually removing the duplicates (a time consuming task).

On completion of the anonimisation and data cleaning processes the information contained in the data warehouse comprised comprised 1200 binary valued attributes derived from the 53 fields after normalisation/discretisation. The number of records remained more or less unchanged, at 150,000, each describing a single patient consultation (a small number of corrupted and duplicated records were removed). Longitudinal data sets could then be extracted from this warehouse using various parameters.

4.1 Episodes

From Section 2 the temporal window during which data associated with a single screening is collected, is referred to as an *episode*. Patients are usually screened once a year although there are many exceptions. For the temporal pattern identification process the annual sequence was taken as the “time stamp”. The number of screening episodes per patient that have been recorded varies (at time of writing) between one and twenty with an average number of five consultations. It should also be noted that in some cases a patient might not participate in an annual screening episode, in which case there was no record for that episode although this did not adversely affect the temporal pattern mining process. In some other cases the sequence of episodes terminated because the patient “dropped” out of the screening programme (was referred into the Hospital Eye Service; moved away; died).

The data associated with a single episode, as also noted above, may actually be recorded over several months. In some cases it was not clear whether a particular set of data entries belonged to a single episode or not. Some empirical evaluation indicated that the elapsed time between logging the initial screening data and (where appropriate) the results of biomicroscopy was less than 91 days. This was used as a working threshold to identify episode boundaries. For the research described here a window of 91 days was therefore used to collate data into a single “screening episode”.

The time lapse between screening episodes is typically twelve months although the data collection displays a great deal of variation resulting from practical considerations effecting the implementation of the screening programme (this is illustrated in Figure 1. As noted above, according to the nature of the retinopathy, additional episodes may take place. Consequently more than one consultation can take place per year in which case the second consultation was ignored.

In summary:

- The time stamps used in the temporal patten study are episode numbers.
- The study assumes one episode (consultation) per year; where more than one took place in each time stamp the earliest one was used.
- To associate appropriate patient data with a single episode a 90 day window was used.
- Where a specific 91 day window included multiple data records, the most recent data (within the window) was used.

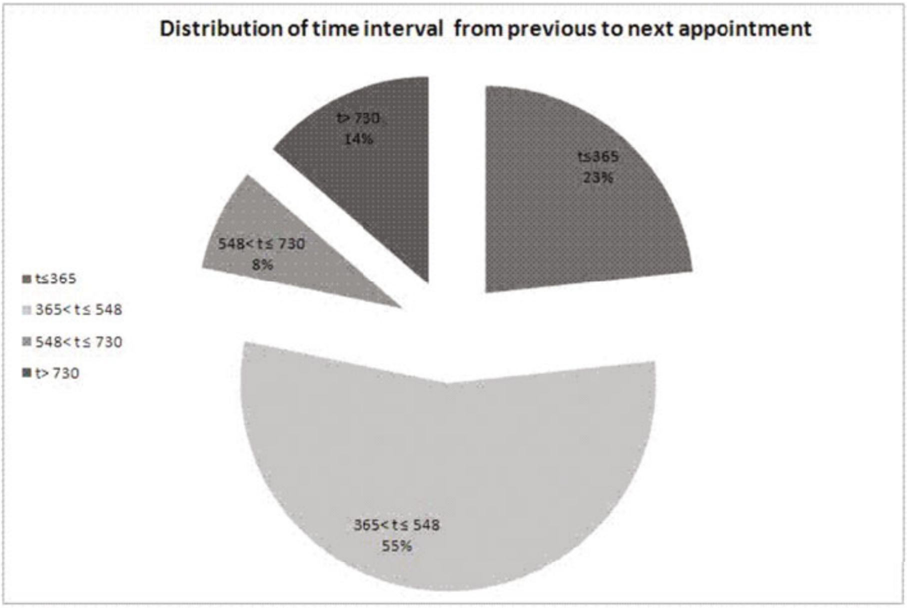


Fig. 1. Time lapse between screening (t = number of days between screenings)

4.2 Normalisation and Discretisation

The temporal pattern mining process (see below for further detail) operated using binary valued data only (frequent pattern mining is typically directed at binary value data). Thus the longitudinal data sets extracted from the data warehouse had to be converted in this format. The LUCS-KDD DN pre-processing software[4] was used for this process. Continuous values were *discretised* into prescribed k ranges giving rise to k binary valued (yes/no) attributes for a single field describing continuous data. Categorical valued fields were normalised so that a field that could have k values was described by k attributes (one per value).

[4] http://www.csc.liv.ac.uk/f̃rans/KDD/Software/LUCS-KDD-DN_ARM/lucs-kdd_DN.html

5 The Temporal Pattern Mining Process

Temporal data comprises a sequence of time stamped data sets. In the context of the work described here a temporal data set comprises a a data set D made up of a sequence of episodes $E_1, \cdots, E_n$ (where n is the number of episodes) and each episode comprises a set of records, $E = \{R_1, \cdots, R_m\}$ (where m is the number of records). The ith record within the jth episode is denoted by R_{ij}; the sequence of records R_{i1} to R_{im} denote the sequence of records associated with patient i for episode 1 to m. Each record comprises some subset of an identified global set of attributes A which in turn represent the possible field-values that exist in the original data set. The objective is to find patterns that exist across the set of episode (E) that feature specific trends. The pattens are defined in terms of subsets of A that occur frequently within episodes. The temporal patterns (trends) are then defined in terms of the changing support values between adjacent episodes. Thus a temporal pattern is defined as a tuple (a, S) where a is an itemset such that $a \subset A$, and $S = \{s_1, \cdots, s_n\}$ such that s_i is the support for the itemset a at episode i. Trends in temporal patterns are then defined using mathematical identities (*prototypes*). For example an increasing trend line would be defined as follows.

$$trend = \sum_{i=1}^{N-1} \frac{S_{i+1} - S_i}{S_i} + 1 \tag{1}$$

Thus if $\{S_{i+1}\}/S_i > 1$ for all i from $i = 1$ to $i = n - 1$, and the trend (Growth Rate) is greater than some user defined *Growth Rate Threshold*, p, then the associated attribute set is displaying an “increasing” trend. The value of p is selected according to the magnitude of the trend increase that the end user is interested in. Note that this increasing trend concept operates in a similar manner to the Emerging pattern (EP) concept [4], as described above, except that the patterns exist across many data sets whereas JEPs are normally determined with respect to two data sets. The similarity is sufficient to allow operational comparisons to be made as reported in the following section.

Decreasing trends and “constant” trends may be defined in a similar manner as follows. If $\{s_{i+1}\}/S_i < 1$ for all i from $i = 1$ to $i = n - 1$, and the trend (Growth Rate) is less than some user defined *Growth Rate Threshold*, p, then the associated attribute set is displaying a “decreasing” trend. Note that in this case the Growth Rate will be negative. If $\{s_{i+1}\}/S_i = 1 \pm k$ for all i from $i = 1$ to $i = n - 1$, and the trend (Growth Rate) is constant ($\pm k$), where k is a Tolerance Threshold, then that attribute set is said to be displaying a “constant” trend.

The temporal pattens were generated by applying a frequent pattern mining algorithm to each episode in a given longitudinal data set. The Total From Partial (TFP) algorithm [2,3] was used. TFP is a fast pattern mining algorithm that operates using the support-confidence framework [1]. The support threshold was used to limit the number of potential patterns that might be of interest. Note that for a temporal pattern to be recorded it must be frequent at all time stamps, therefore a low support threshold must be used.

6 Evaluation

The above temporal pattern mining process was evaluated using a three episode longitudinal data set extracted from the data warehouse (as defined above). Three episodes ($E_1, \cdots, E_3$) was chosen as this was anticipated to result in a large number of patterns. The data set used comprised 9,400 records. The first experiment reported here compares the operation of the advocated frequent pattern mining approach, in the context of the increasing trends proto-type, with the concept of Emerging Pattern (EP) mining. The comparison is made in terms of the number of temporal patterns (EPs) generated. Recall that JEP mining finds patterns that exist between pairs of data sets; i.e. E_1 and E_2, and E_2 and E_3 in this case: thus two sets of JEPs will be identified with respect to the given data. The results are presented in Figures 2 and 3. The figures indicate the number of discovered patterns using a number of Growth Thresholds (p) from $p = 1.1$ to $p = 1.8$ and three support thresholds ($s = 0.5$, $s = 0.05$ and $s = 0.005$). Figure 2 gives the number of patterns produced using the advocated approach, and Figure 3 the number of patterns using standard EP mining. Comparison of the figures indicates that, using the advocated approach, fewer patterns are produced than when adopting EP mining. From the figures it can also be seen that, as the Growth Rate Threshold (p) value is increased, the number of trends (EPs) decreases as only the "steeper" trends are discovered. The figures also confirm that, as expected, the number of identified patterns increases as the user defined support threshold (s) decreases.

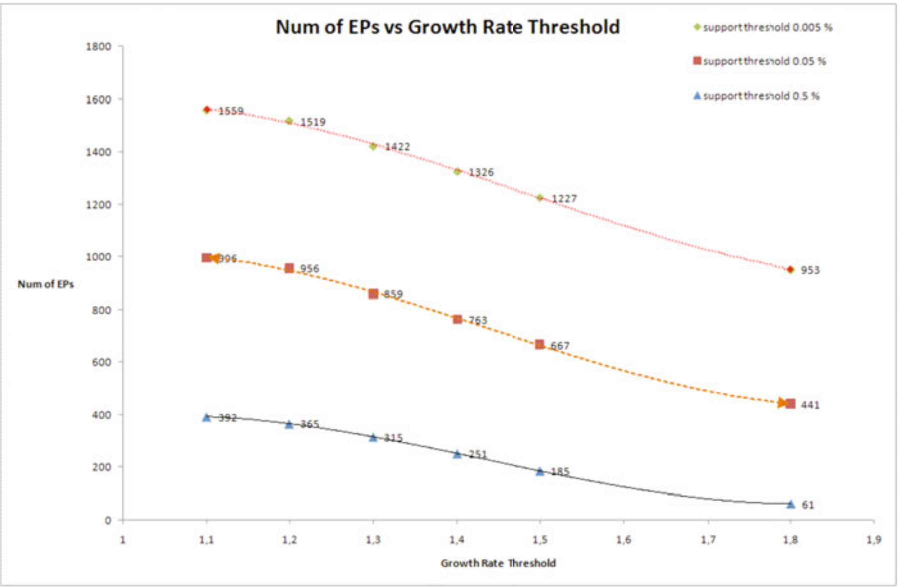


Fig. 2. Number of temporal patterns, increasing trends, identified using the advocated approach

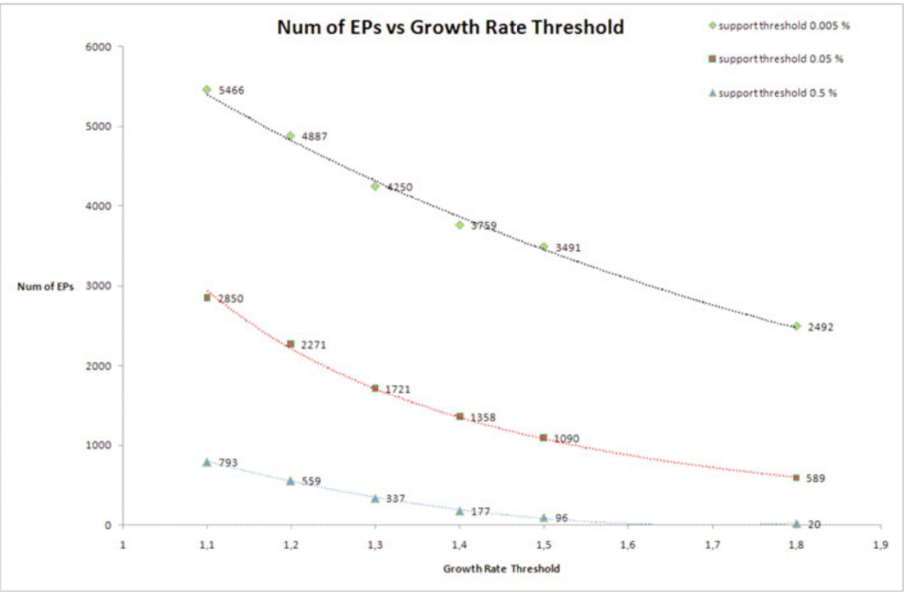


Fig. 3. Number of temporal patterns identified using EP mining (Dong and Li [4])

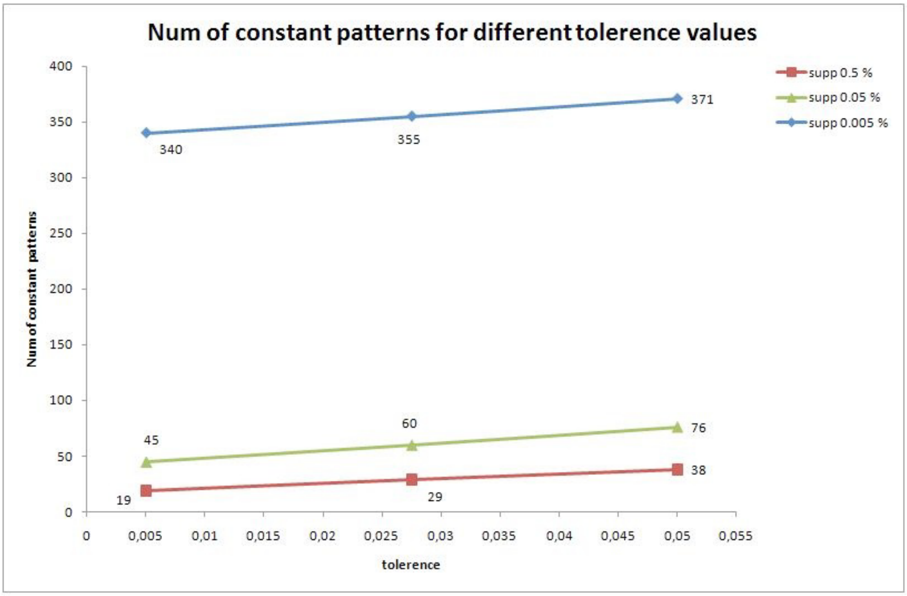


Fig. 4. Number of constant patterns using different values of k

Table 2. Number of Identified Increasing, Decreasing and Constant Patterns ($p = 1.1$, $k = 0.05$)

Support T'hold	Increasing	Decreasing	Constant	Total
0.005	1559	499	371	2429
0.05	996	378	76	1450
0.5	392	221	38	651

The second experiment considered the effect of the value of the Tolerance Threshold, k, on the number of detected constant trends. A range of k values (from 0.005 to 0.055) were used coupled with the sequence of support thresholds used in the previous experiments. The results are presented in Figure 4. From the graph it can be seen that the support threshold setting has a much greater effect on the number of constant trends identified than the value of k.

For completeness Table 2 presents a summary of the number of patterns discovered in each category (increasing, decreasing, constant) using a range of support thresholds. With respect to the increasing and decreasing trend patterns a Growth Rate Threshold, p, of 1.1 was used. With respect to the constant patterns a Tolerance Threshold, k, of 0.05. was used.

7 Summary and Conclusion

In this paper we have described an approach to temporal patten mining as applied within the context of a diabetic retinopathy application. The particular application was of interest because it comprised a large longitudinal data set that contained a lot of noise and thus presented a significant challenge in several areas. A mechanism for generating specific temporal patterns was described where the nature of the desired patterns is defined using prototypes (which are themselves defined mathematically). The technique was evaluated by considering the effect of changing the threshold values required by the system and comparing with an established Emerging Pattern (EP) mining approach. The paper also describes an interesting approach to data cleaning using the concept of logic rules to address issues of missing values and contradictory/anomalous values. The research team have been greatly encouraged by the results, and are currently working on more versatile mechanisms for defining prototypes, so that a greater variety of prototypes can be specified. For example the specification of a minimum and maximum p threshold. In addition novel techniques for interpretation of output in a clinical setting are being developed.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for mining Association Rules. In: Proc. 20th Very Large Data Bases conference (VLDB 1994), pp. 487–449 (1994)
2. Coenen, F.P., Leng, P., Ahmed, S.: Data Structures for association Rule Mining: T-trees and P-trees. IEEE Transactions on Data and Knowledge Engineering 16(6), 774–778 (2004)

3. Coenen, F.P., Leng, P., Goulbourne, G.: Tree Structures for Mining Association Rules. Journal of Data Mining and Knowledge Discovery 8(1), 25–51 (2004)
4. Dong, G., Li., J.: Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In: Proc. SIGKDD, pp. 43–52. ACM, New York (1999)
5. Fan, H., Kotagiri, R.: A Bayesian Approach to Use Emerging Patterns for classification. In: Proceedings of the 14th Australasian database conference, vol. 17, pp. 39–48 (2003)
6. van der Kamp, L.J.T., Bijleveld, C.C.J.H.: Methodological issues in longitudinal research. In: Bijleveld, C.C.J.H., van der Kamp, L.J.T., Mooijaart, A., van der Kloot, W., van der Leeden, R., van Der Burg, E. (eds.) Longitudinal Data Analysis, Designs Models and Methods, pp. 1–45. SAGE publications, Thousand Oaks (1988)
7. Kalton, G., Kasprzyk, D.: The treatment of missing survey data. Survey Methodology 12, 1–16 (1986)
8. Khan, M.S., Coenen, F., Reid, D., Tawfik, H., Patel, R., Lawson, A.: A Sliding Windows based Dual Support Framework for Discovering Emerging Trends from Temporal Data. To appear in KBS Journal (2010)
9. Kimm, S.Y.S., Glynn, N.W., Kriska, A.M., Fitzgerald, S.L., Aaron, D.J., Similo, S.L., McMahon, R.P., Barton, B.A.: Longitudinal changes in physical activity in a biracial cohort during adolescence. Medicine and Science in Sports and Exercise 32(8), 1445–1454 (2000)
10. Levy, M.L., Cummings, J.L., Fairbanks, L.A., Bravi, D., Calvani, M., Carta, A.: Longitudinal assessment of symptoms of depression, agitation, and psychosis in 181 patients with Alzheimer's disease. American Journal of Psychiatry 153, 1438–1443 (1996)
11. Little, R.J., Rubin, D.B.: Statistical Analysis with Missing Data, 2nd edn. John Wiley and Sons, New York (2002)
12. Mum̃oz, J.F., Rueda, M.: New imputation methods for missing data using quantiles. Journal of Computational and Applied Mathematics 232(2), 305–317 (2009)
13. Nohuddin, P.N.E., Coenen, F., Christley, R., Setzkorn, C.: Trend Mining in Social Networks: A Study Using A Large Cattle Movement Database. To appear, Proc. ibia Industrial Conf. on Data Mining. LNCS (LNAI), Springer, Heidelberg (2010)
14. Singer, J.D., Willet, J.B.: Applied longitudinal data analysis modelling change and event occurrence. Oxford University Press, Oxford (2003)
15. Skinner, J.D., Carruth, B.R., Wendy, B., Ziegler, P.J.: Children's Food Preferences A Longitudinal Analysis. Journal of the American Dietetic Association 102(11), 1638–1647 (2002)
16. Twisk, J.W.R.: Applied longitudinal data analysis for epidemiology: a practical guide. Cambridge University Press, Cambridge (2003)
17. Wagner, M., et al.: What Happens Next? Trends in Postschool Outcomes of Youth with Disabilities: The Second Comprehensive Report from the National Longitudinal Transition Study of Special Education Students. SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025-3493 (1992)
18. Yamaguchi, K., Tetro, A.M., Blam, O., Evanoff, B.A., Teefey, S.A., Middleton, W.D.: Natural history of asymptomatic rotator cuff tears: A longitudinal analysis of asymptomatic tears detected sonographically. Journal of Shoulder and Elbow Surgery 10(3), 199–203 (2001)
19. Zhu, Y., Shasha, D.: StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. In: Proc VLDB, pp. 358–369 (2002)

Selection of High Risk Patients with Ranked Models Based on the CPL Criterion Functions[*]

Leon Bobrowski

Faculty of Computer Science, Białystok Technical University
Institute of Biocyberinetics and Biomedical Engineering, PAS, Warsaw, Poland
leon@ibib.waw.pl

Abstract. Important practical problems in computer support medical diagnosis are related to screening procedures. Identification of high risk patients can serve as an example of such a problem. The identification results should allow to select a patient in an objective manner for additional therapeutic treatment. The designing of the screening tools can be based on the minimisation of the convex and piecewise linear (CPL) criterion functions. Particularly ranked models can be designed in this manner for the purposes of screening procedures.

Keywords: screening procedures, convex and piecewise linear (CPL) criterion functions, ranked models.

1 Introduction

One of the most important groups of problems in computer aided medical diagnosis are those related to screening procedures. We are considering screening procedures which are aimed at selecting high risk patients. High risk patients should be possibly early directed to a special therapeutic treatment. For example, the success of cancer therapy depends on early detection and beginning of this disease therapy.

The screening procedures usually result from certain probabilistic prognostic models. The survival analysis methods give a theoretical framework for designing screening procedures [1], [2]. In particular, the Cox model is commonly used in survival analysis for selection of high risk patients [3].

However, constraints met in practical implementation of screening procedures indicate limitations of the probabilistic modeling in this context. In practice, the screening procedures are designed on the basis of available medical databases which rarely fit in the statistical principles of parameters estimation. First of all, the number of cases (patients) in medical databases is usually too low and the number of parameters (features) describing particular patients is too high for reliable estimation of selected prognostic model parameters. Secondly, the probabilistic assumptions linked to particular prognostic models are often unverifiable.

[*] This work was supported by the by the NCBiR project N R13 0014 04, and partially financed by the project S/WI/2/2010 from the Białystok University of Technology, and by the project 16/St/2010 from the Institute of Biocybernetics and Biomedical Engineering PAS.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 432–441, 2010.

For these reasons, designing of effective and reliable screening procedures is still related to open research and implementation problems. Here we are examining the possibility of using ranked modeling in designing screening procedures [4]. In particular, we are taking into account linear ranked models designed through the minimisation of the convex and piecewise linear (*CPL*) criterion functions [5], [6]. These criterion functions are defined on the survival analysis data sets. An important problem analysed here is feature selection aimed at reducing ranked models dimensionality [7].

2 Feature Vectors and Ranked Relations Originating from Survival Analysis

Let us assume that m patients O_j collected in a given medical database are represented as n-dimensional feature vectors $\mathbf{x}_j[n] = [x_{j1},......, x_{jn}]^T$ or as points in the n-dimensional feature space $F[n]$ ($\mathbf{x}_j[n] \in F[n]$, $j = 1,.....,m$). The component x_{ji} of the vector $\mathbf{x}_j[n]$ is the numerical value of the i-th feature x_i of the patient (*object*) O_j. For example, the components x_{ji} can be the numerical results of diagnostic examinations of the given patient O_j. The feature vectors $\mathbf{x}_j[n]$ can be of a mixed type and represent a type of measurement (for example ($x_{ji} \in \{0,1\}$, or $x_{ji} \in R^1$).

We are taking into consideration the learning data set C built from m feature vectors $\mathbf{x}_j[n]$ which represent particular patients O_j:

$$C = \{\mathbf{x}_j[n]\} \;\; (j = 1,......., m) \tag{1}$$

Let us consider the relation "O_j is *less risky* than O_k" between selected patients O_j and O_k represented by the feature vectors $\mathbf{x}_j[n]$ and $\mathbf{x}_k[n]$. Such relation between patients O_j and O_j can implicate the *ranked relation* "$\mathbf{x}_j[n] \prec \mathbf{x}_k[n]$" between adequate feature vectors $\mathbf{x}_j[n]$ and $\mathbf{x}_k[n]$.

$$(O_j \text{ is } \textit{less risky}, \text{ then } O_k) \Rightarrow (\mathbf{x}_j[n] \prec \mathbf{x}_k[n]) \tag{2}$$

The relation "$\mathbf{x}_j[n] \prec \mathbf{x}_k[n]$" between the feature vectors $\mathbf{x}_j[n]$ and $\mathbf{x}_k[n]$ means that the pair $\{\mathbf{x}_j[n], \mathbf{x}_k[n]\}$ is *ranked*. The ranked relations between particular feature vectors $\mathbf{x}_j[n]$ and $\mathbf{x}_k[n]$ should result from additional knowledge about the patients O_j and O_k. Such additional knowledge could result from an information about *survival time* T_j of particular patients O_j collected in the given database.

Traditionally, the *survival analysis* data sets C_s have the below structure [1]:

$$C_s = \{\mathbf{x}_j[n], t_j , \delta_j\} \;\; (j = 1,.......,m) \tag{3}$$

where t_j is the *observed survival time* between the entry of the j-th O_j patient into the study and the end of the observation, δ_j is an indicator of failure of this patient ($\delta_j \in \{0,1\}$): $\delta_j = 1$ - means the end of observation in the event of interest (*failure*), $\delta_j = 0$ - means that the follow-up on the j-th patient ended before the event (*the right censored observation*). In this case ($\delta_j = 0$) information about survival time t_j is *not* *complete*. A great part of survival data set C_s can be censored. The survival analysis methods are based in a great part on the Cox model [2], [3]. The ranked models can be also used in the search for a solution of basic problems in survival analysis [4].

The *survival time* T_j can be defined in the below manner on the basis of the set C_s (3):

$$(\forall j = 1,.......,m) \;\; \textbf{\textit{if}} \;\; \delta_j = 1, \;\; \textbf{\textit{then}} \;\; T_j = t_j, \;\; \textbf{\textit{and}} \tag{4}$$
$$\textbf{\textit{if}} \;\; \delta_j = 0, \;\; \textbf{\textit{then}} \;\; T_j > t_j$$

Assumption: If the survival time T_j (4) of the *j*-th patients O_j is longer then the survival time T_j of the *j*-th patients O_j, then the patients O_j was *less risky* (2) then the patients O_k:

$$(T_j > T_k) \Rightarrow (O_j \text{ is } \textit{less risky} \text{ than } O_k) \Rightarrow (\mathbf{x}_j[n] \prec \mathbf{x}_k[n]) \tag{5}$$

This implication can be expressed also by using the observed survival time t_j and t_k (3):

$$(t_j > t_k \text{ and } \delta_k = 1) \Rightarrow (O_j \text{ is } \textit{less risky} \text{ than } O_k) \Rightarrow (\mathbf{x}_j[n] \prec \mathbf{x}_k[n]) \tag{6}$$

3 Linear Ranked Models

Let us consider such transformation of *n*-dimensional feature vectors $\mathbf{x}_j[n]$ on the ranked line $y = \mathbf{w}[n]^T\mathbf{x}_j[n]$, which preserves the ranked relations $''\mathbf{x}_j[n] \prec \mathbf{x}_k[n]''$ (2) as precisely as possible

$$y_j = y_j(\mathbf{w}[n]) = \mathbf{w}[n]^T\mathbf{x}_j[n] \tag{7}$$

where $\mathbf{w}[n] = [w_1,......,w_n]^T$ is the vector of parameters.

Definition 1: The relation $''\mathbf{x}_j[n] \prec \mathbf{x}_j[n]''$ (2) is fully preserved by the *ranked line* (7) if and only if the following implication holds:

$$(\forall (j, k)) \;\; \mathbf{x}_j[n] \prec \mathbf{x}_k[n] \Rightarrow y_j(\mathbf{w}[n]) < y_k(\mathbf{w}[n]) \tag{8}$$

The procedure of the ranked line designing can be based on the concept of positively and negatively oriented dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$ [6].

Definition 2: The ranked pair $\{\mathbf{x}_j[n],\mathbf{x}_{j'}[n]\}$ $(j < j')$ of the feature vectors $\mathbf{x}_j[n]$ and $\mathbf{x}_{j'}[n]$ constitutes the *positively oriented dipole* $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$ $(\forall (j, j') \in I^+)$ if and only if $\mathbf{x}_j[n] \prec \mathbf{x}_{j'}[n]$.

$$(\forall\, (j, j') \in I^{+}) \quad \mathbf{x}_{\mathrm{j}}[n] \prec \mathbf{x}_{\mathrm{j}'}[n] \tag{9}$$

Definition 3: The ranked pair $\{\mathbf{x}_{\mathrm{j}}[n], \mathbf{x}_{\mathrm{j}'}[n]\}$ $(j < j')$ of the feature vectors $\mathbf{x}_{\mathrm{j}}[n]$ and $\mathbf{x}_{\mathrm{j}'}[n]$ constitutes the *negatively oriented dipole* $\{\mathbf{x}_{\mathrm{j}}[n], \mathbf{x}_{\mathrm{j}'}[n]\}$ $(\forall (j, j') \in I^{-})$*,* if and only if $\mathbf{x}_{\mathrm{j}'}[n] \prec \mathbf{x}_{\mathrm{j}}[n]$.

$$(\forall\, (j, j') \in I^{-}) \quad \mathbf{x}_{\mathrm{j}'}[n] \prec \mathbf{x}_{\mathrm{j}}[n] \tag{10}$$

Definition 4: The line $\mathrm{y}(\mathbf{w}[n]) = \mathbf{w}[n]^{\mathrm{T}}\mathbf{x}[n]$ (7) is fully consistent (*ranked*) with the dipoles $\{\mathbf{x}_{\mathrm{j}}[n], \mathbf{x}_{\mathrm{j}'}[n]\}$ orientations if and only if

$$\begin{aligned} &(\forall\, (j, j') \in I^{+}) \quad \mathrm{y}_{\mathrm{j}}(\mathbf{w}[n]) < \mathrm{y}_{\mathrm{j}'}(\mathbf{w}[n]) \;\; \textit{and} \\ &(\forall\, (j, j') \in I^{-}) \quad \mathrm{y}_{\mathrm{j}}(\mathbf{w}[n]) > \mathrm{y}_{\mathrm{j}'}(\mathbf{w}[n]), \text{ where } j < j' \end{aligned} \tag{11}$$

All the relations "$\mathbf{x}_{\mathrm{j}}[n] \prec \mathbf{x}_{\mathrm{j}'}[n]$" (2) are fully preserved (8) on the line (7) if and only if all the above inequalities are fulfilled.

The problem of the ranked line designing can be linked to the concept of linear separability of two sets C^{+} and C^{-} of the differential vectors $\mathbf{r}_{\mathrm{jj}'}[n] = \mathbf{x}_{\mathrm{j}'}[n] - \mathbf{x}_{\mathrm{j}}[n]$ which are defined below:

$$\begin{aligned} C^{+} &= \{\mathbf{r}_{\mathrm{jj}'}[n] = (\mathbf{x}_{\mathrm{j}'}[n] - \mathbf{x}_{\mathrm{j}}[n]) \colon (j, j') \in I^{+}\} \\ C^{-} &= \{\mathbf{r}_{\mathrm{jj}'}[n] = (\mathbf{x}_{\mathrm{j}'}[n] - \mathbf{x}_{\mathrm{j}}[n]) \colon \; (j, j') \in I^{-}\}, \text{ where } j < j' \end{aligned} \tag{12}$$

We will examine the possibility of the sets separation C^{+} and C^{-} by a such hyperplane $H(\mathbf{w}[n])$, which passes through the origin $\mathbf{0}$ of the feature space $F[n]$:

$$H(\mathbf{w}[n]) = \{\mathbf{x}[n] \colon \mathbf{w}[n]^{\mathrm{T}}\mathbf{x}[n] = 0\} \tag{13}$$

where $\mathbf{w}[n] = [\mathrm{w}_1, \ldots\ldots, \mathrm{w}_{\mathrm{n}}]^{\mathrm{T}}$ is the vector of parameters.

Definition 5: The sets C^{+} and C^{-} (12) are linearly separable with the threshold equal to zero if and only if there exists such a parameter vector $\mathbf{w}^{*}[n]$ that:

$$\begin{aligned} &(\forall\, (j, j') \in I^{+}) \quad \mathbf{w}^{*}[n]^{\mathrm{T}}\, \mathbf{r}_{\mathrm{jj}'}[n] > 0 \\ &(\forall\, (j, j') \in I^{-}) \quad \mathbf{w}^{*}[n]^{\mathrm{T}}\, \mathbf{r}_{\mathrm{jj}'}[n] < 0 \end{aligned} \tag{14}$$

The above inequalities can be represented in the following manner:

$$\begin{aligned} (\exists \mathbf{w}^{*}[n])\, (\forall\, (j, j') \in I^{+}) \quad &\mathbf{w}^{*}[n]^{\mathrm{T}}\, \mathbf{r}_{\mathrm{jj}'}[n] \;\; \geq \;\; 1 \\ (\forall\, (j, j') \in I^{-}) \quad &\mathbf{w}^{*}[n]^{\mathrm{T}}\, \mathbf{r}_{\mathrm{jj}'}[n] \;\; \leq -1 \end{aligned} \tag{15}$$

Remark 1: If the parameter vector $\mathbf{w}^*[n]$ linearly separates (14) the sets C^+ and C^- (12), then the line $y_j(\mathbf{w}^*[n]) = \mathbf{w}^*[n]^T\mathbf{x}_j[n]$ is fully consistent (11) with the dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$ orientation.

4 *CPL* Criterion Functions

The separating hyperplane $H(\mathbf{w}[n])$ (13) could be designed through the minimisation of the convex and piecewise linear (*CPL*) criterion function $\Phi(\mathbf{w}[n])$ which is similar to the perceptron criterion function used in the theory of neural networks and pattern recognition [8], [9]. Let us introduce for this purpose the positive $\varphi_{jj'}^+(\mathbf{w}[n])$ and negative $\varphi_{jj'}^-(\mathbf{w}[n])$ penalty functions (Fig. 1):

$(\forall\, (j, j') \in I^+)$

$$\varphi_{jj'}^+(\mathbf{w}[n]) = \begin{cases} 1 - \mathbf{w}[n]^T\,\mathbf{r}_{jj'}[n] & \textit{if}\ \mathbf{w}[n]^T\mathbf{r}_{jj'}[n] < 1 \\ 0 & \textit{if}\ \mathbf{w}[n]^T\mathbf{r}_{jj'}[n] \geq 1 \end{cases} \tag{16}$$

and $(\forall\, (j, j') \in I^-)$

$$\varphi_{jj'}^-(\mathbf{w}[n]) = \begin{cases} 1 + \mathbf{w}[n]^T\mathbf{r}_{jj'}[n] & \textit{if}\ \mathbf{w}[n]^T\mathbf{r}_{jj'}[n] > -1 \\ 0 & \textit{if}\ \mathbf{w}[n]^T\mathbf{r}_{jj'}[n] \leq -1 \end{cases} \tag{17}$$

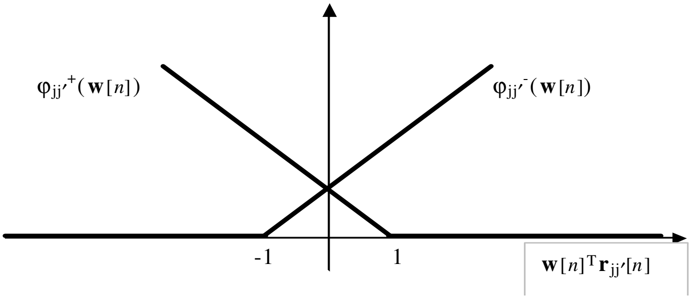


Fig. 1. The penalty functions $\varphi_{jj'}^+(\mathbf{w}[n])$ (16) and $\varphi_{jj'}^-(\mathbf{w}[n])$ (17)

The criterion function $\Phi(\mathbf{w}[n])$ is the weighted sum of the above penalty functions

$$\Phi(\mathbf{w}[n]) = \sum_{(j,j')\in I+} \alpha_{jj'}\, \varphi_{jj'}^+(\mathbf{w}[n]) + \sum_{(j,j')\in I^-} \alpha_{jj'}\, \varphi_{jj'}^-(\mathbf{w}[n]) \tag{18}$$

where $\alpha_{jj'}$ ($\alpha_{jj'} > 0$) is a nonnegative parameter (*price*) related to the dipole $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$ $(j < j')$

The criterion function $\Phi(\mathbf{w}[n])$ (18) is the convex and piecewise linear (*CPL*) function as the sum of such type of the penalty functions $\phi_{jj'}{}^{+}(\mathbf{w}[n])$ (16) and $\phi_{jj'}{}^{-}(\mathbf{w}[n])$ (17). The basis exchange algorithms, similarly to linear programming, allow to find a minimum of such functions efficiently, even in the case of large, multidimensional data sets C^+ and C^- (12) [10]:

$$\Phi^* = \Phi(\mathbf{w}^*[n]) \ = min\ \Phi(\mathbf{w}[n]) \geq 0 \tag{19}$$

The optimal parameter vector $\mathbf{w}^*[n]$ and the minimal value Φ^* of the criterion function $\Phi(\ \mathbf{w}[n])$ (18) can be applied to a variety of data ranking problems. In particular, the below *ranked model* can be designed this way.

$$y(\mathbf{w}^*[n]) = \mathbf{w}^*[n]^T\mathbf{x}[n] \tag{20}$$

Lemma 1: The minimal value Φ^* (19) of the criterion function $\Phi(\mathbf{w}[n])$ (18) is nonnegative and equal to zero if and only if there exists such a vector $\mathbf{w}^*[n]$ that the ranking of the points $y_j = \mathbf{w}^*[n]^T\mathbf{x}_j[n]$ on the line (7) are fully consistent (11) with the dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$ orientations.

The proof of this *Lemma* can be found in the earlier paper [6].

The modified criterion function $\Psi_\chi(\mathbf{w}[n])$ which includes additional *CPL* penalty functions in the form of the absolute values $|\ w_i\ |$ multiplied by the *feature costs* γ_i has been introduced for the purpose of feature selection [7].

$$\Psi_\lambda(\mathbf{w}[n]) \ = \ \Phi(\mathbf{w}[n]) + \ \lambda \sum_{i \in I} \gamma_i\ |w_i| \tag{21}$$

where λ ($\lambda \geq 0$) is the *cost level*, and $I = \{1,......,n\}$.

The criterion function $\Psi_\lambda(\mathbf{w}[n])$ (21), similarly to the function $\Phi(\mathbf{w}[n])$ (18) is convex and piecewise-linear (*CPL*). The basis exchange algorithms allow to find efficiently the optimal vector of parameters (*vertex*) $\mathbf{w}_\lambda[n]$ of the function $\Psi_\lambda(\mathbf{w}[n])$ with different values of parameter λ [10]:

$$(\exists(\mathbf{w}_\lambda[n]))\ \ (\forall\mathbf{w}[n])\ \ \Psi_\lambda(\mathbf{w}[n]) \geq \Psi_\lambda(\mathbf{w}_\lambda[n]) = \ \Psi_\lambda{}^\wedge \tag{22}$$

The parameters $\mathbf{w}_\lambda[n] = [w_{\lambda 1},....., w_{\lambda n}]^T$ (22) define the optimal separating hyperplane $H(\mathbf{w}_\lambda[n])$ (13). Such features x_i which have the weights $w_{\lambda i}$ equal to zero ($w_{\lambda i} = 0$) in the optimal vector $\mathbf{w}_\lambda[n]$ (22) can be reduced without changing the location of the optimal separating hyperplane $H(\mathbf{w}_\lambda[n])$ (13). As a result, the below rule of the feature reduction based on the components $w_{\lambda i}$ of the optimal vector of parameters $\mathbf{w}_\lambda[n] = [w_{\lambda 1},....., w_{\lambda n}]^T$ (22) has been proposed [7]:

$$(w_{\lambda i.} = \ 0)\ \Rightarrow \text{(the feature } x_i \text{ is reduced)} \tag{23}$$

The minimal value (22) of the *CPL* criterion function $\Psi_\lambda(\mathbf{w}[n])$ (21) represents an optimal balance between linear separability of the sets C^+ and C^- (12) and features costs determined by the parameters λ and γ_i. We can remark that a sufficiently increased value of the *cost level* λ in the minimized function $\Psi_\lambda(\mathbf{w}[n])$ (21) results in an increase number of the reduced features x_i (23). The dimensionality of the feature $F[n]$ can be reduced arbitrarily by a successive increase of the parameter λ in the criterion function $\Psi_\lambda(\mathbf{w}[n])$ (21). Such method of feature selection has been named *relaxed linear separability* [7].

The feature selection procedure is an important part of designing ranked models (20). The feature selection process is aimed at reducing the maximal number of unimportant features x_i. The *reduced ranked model* $y(\mathbf{w}_\lambda'[n'])$ can be defined by using the optimal vector of parameters $\mathbf{w}_\lambda[n]$ (22) and the rule (23):

$$y(\mathbf{w}_\lambda'[n']) = \mathbf{w}_\lambda'[n']^T\mathbf{x}[n'] \tag{24}$$

where $\mathbf{w}_\lambda'[n']$ is such vector of parameters which is obtained from the optimal vector $\mathbf{w}_\lambda[n]$ (22) by reducing components $w_{\lambda i}$ equal to zero ($w_{\lambda i.}= 0$), and $\mathbf{x}[n']$ is the reduced feature vector (23) obtained in the same way as $\mathbf{w}_\lambda'[n']$.

The dimensionality reduction of the prognostic model (24), which is based on the relaxed linear separability method allows, among others, to enhance *risk factors* x_i influencing given disease.

5 Selection of High Risk Patients

The reduced ranked model (24) can be used in selection of high risk patients O_j. These models define transformations of the multidimensional feature vectors $\mathbf{x}_j[n']$ (1) on the points y_j (7) which represent particular patients O_j on the ranked line. As a result, the *ranked sequence* of patients $O_{j(i)}$ can be obtained:

$$O_{j(1)},\ O_{j(2)}, \ldots\ldots.,\ O_{j(m)}, \text{ where} \tag{25}$$
$$y_{j(1)} \geq y_{j(2)} \geq \ldots\ldots \geq y_{j(m)}$$

The patients $O_{j(i)}$ with the largest values $y_{j(i)}$ which are situated on the top of the above sequence can be treated as *high risk patients*. The patients $O_{j(i)}$ which are situated at the beginning of this sequence can be treated as a *low risk*.

The models (20) or (24) can be applied not only to the patients O_j represented by the feature vectors $\mathbf{x}_j[n]$ from the set C (1). Let us assume that a new patient O_a is represented by the feature vector $\mathbf{x}_a[n]$. The model (24) allows to compute the point $y_a = \mathbf{w}_\lambda'[n']^T\mathbf{x}_a[n']$ on the ranked line. If the new patient O_a is located at the top of the ranked sequence (25),) next to the high risk patients $O_{j(i)}$, then it can be also treated as a *high risk*. In other cases, O_a should not be treated as a *high risk* patient.

The transformation (24) can be treated as a *prognostic model* of a given disease ω_k development. Such model can represent a *main trend* in the disease ω_k development. The model is designed on the basis of information contained in the *survival analysis* data sets C_s (3).

A very important problem in practice is quality evaluation of the prognostic models (24). One of the possibilities is to use the *ranked error rate* $e_r(\mathbf{w}_\lambda'[n'])$ the model (24) evaluation.

$$e_r(\mathbf{w}_\lambda'[n']) = m_r'(\mathbf{w}_\lambda'[n']) \,/\, m_r \tag{26}$$

where m_r is the number of positive (9) and negative (10) dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$, where $j < j'$. $m_r'(\mathbf{w}_\lambda'[n'])$ is the number of such dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$, which are wrongly oriented (not consistent with the rule (11)) on the line $y(\mathbf{w}_\lambda'[n'])$ (24).

If the same dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$ are used for the model (24) designing and for the model evaluation, then the error rate estimator $e_r(\mathbf{w}_\lambda'[n'])$ (26) is *positively biased* [11]. The error rate estimator $e_r(\mathbf{w}_\lambda'[n'])$ (26) is called the *apparent error* (*AE*). The *cross-validation* techniques are commonly used in model evaluation because these techniques allow to reduce the *bias* of the error estimation.

In accordance with the cross-validation *p – folds* procedure, the set of all the dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$ (9), (10) is divided in the p near equal parts P_i (for example p = 10). During one step the model is designed on the dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$ belonging to $p - 1$ (*learning*) parts P_i and evaluated on the elements of one (*testing*) part $P_{i'}$. Each part P_i serves once as the testing part $P_{i'}$ during successive p steps. The *cross-validation error rate* (*CVE*) $e_{CVE}(\mathbf{w}_\lambda'[n'])$ is obtained as the mean value of the error rates $e_r(\mathbf{w}_\lambda'[n'])$ (26) evaluated on the p testing parts $P_{i'}$.

The *leave one method* is the particular case of the *p – folds* cross-validation, when p is equal to the number m_r ($p = m_r$) of the positively (9) and negatively (10) dipoles $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$. In this case, each testing part $P_{i'}$ contains exactly one element $\{\mathbf{x}_j[n], \mathbf{x}_{j'}[n]\}$.

The *relaxed linear separability* method of feature selection can be used during designing the reduced ranked models $y(\mathbf{w}_\lambda'[n'])$ (24) [7]. In accordance with this method, a successive increase of the *cost level* λ in the minimized function $\Psi_\lambda(\mathbf{w}[n])$ (21) causes a reduction of additional features x_i (23). In this way, the less important features x_i are eliminated and the descending sequence of feature subspaces $F_k[n_k]$ ($n_k > n_{k+1}$) is generated. Each feature subspace $F_k[n_k]$ in the below sequence can be linked to some value λ_k of the cost level λ (21):

$$\begin{gathered} F[n] \supset F_1[n_1] \supset F_2[n_2] \supset \ldots\ldots\ldots \supset F_k[n_k], \text{ where} \\ 0 \le \lambda_0 < \lambda_1 < \lambda_2 < \ldots\ldots < \lambda_k \end{gathered} \tag{27}$$

Particular feature subspaces $F_k[n_k]$ (27) have been evaluated by using the cross-validation error rate (*CVE*) $e_{CVE}(\mathbf{w}_\lambda'[\,n_k])$ (26). The below figure shown an example of experimental results. The evaluation results of descending sequence (27) of feature subspaces $F_k[n_k]$ is shown on this figure:

The above Figure illustrates the feature reduction process which begins from the dimensionality n = 100. In this case, the data set C (1) contained about m = 200 feature vectors $\mathbf{x}_j[n]$. It can be seen on the Figure 1, that it exists such feature subspaces $F_k[n_k]$ of dimensionality n_k = 30 or more, which allow to obtain the fully consistent (11) ranked lines $y(\mathbf{w}[n_k])$ (7).

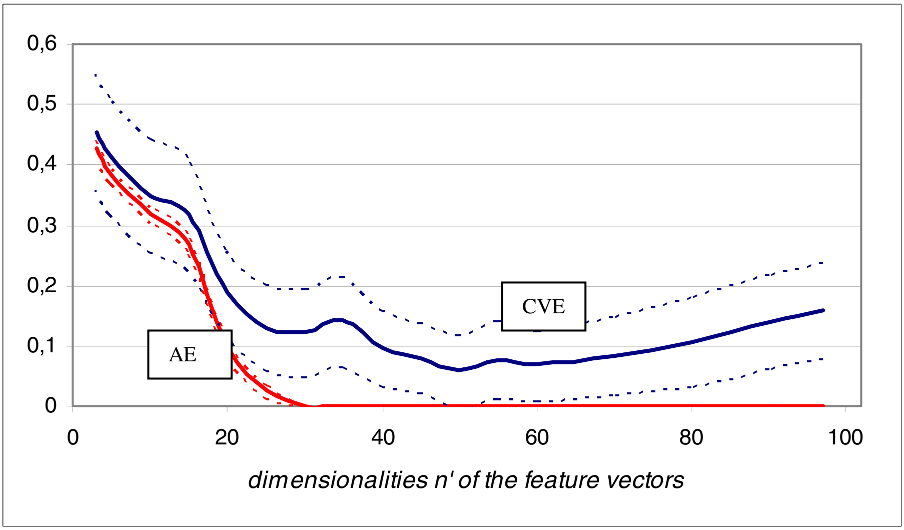


Fig. 2. The apparent error (*AE*) (26) and the cross-validation error (*CVE*) in different feature subspaces $F_k[n_k]$ of the sequence (27). The upper solid line represents the cross-validation error (*CVE*), the lower solid line represents the apparent error (*AE*) (26). The broken line represents the standard deviation.

In accordance with the relaxed linear separability method, the process of successive reduction of the less important features x_i should be stopped at the dimensionality $n_k' \approx 50$, where the cross-validation error rate $e_{CVE}(\mathbf{w}_\lambda'[n'])$ (26) reaches its lowest value. The feature subspace $F_k[n_k']$ is treated as the optimal one in accordance with this method of feature selection [7].

6 Concluding Remarks

Ranked modeling has been applied here to designing linear prognostic models $y(\mathbf{w}_\lambda'[n'])$ on the basis of the survival analysis data set C_s (3). The described designing process is based on multiple minimization of the convex and piecewise linear (*CPL*) criterion function $\Psi_\lambda(\mathbf{w}[n])$ (21) defined on the data set C_s (3). The basis exchange algorithms allow to carry out such multiple minimization efficiently.

The process of ranked models designing described here includes feature selection stage, which is based on the relaxed linear separability method. This method of feature selection is linked with the evaluation of prognostic models $y(\mathbf{w}_\lambda'[n'])$ (24) by using the cross-validation techniques.

The linear ranked models designing has been discussed here in the context of the screening procedures aimed at selecting high risk patients. High risk patients should be possibly early detected for the purpose of special therapeutic treatment. The proposed solution can be applied also to other areas. For example, the bankruptcy in economy or the mechanical reliability problems can be analyzed and solved in a similar manner. The patients O_j could be replaced by sets of dynamical objects or events E_j.

One of the important problems in ranked modeling could be feature decomposition of nonlinear family of ranked relations (9), (10) into a set of linear families. The single linear model which does not fit well to all ranked relations should be replaced in this case by a family of well fitted local ranked models.

References

[1] Marubini, E., Valsecchi, M.G.: Analysing Survival Data from Clinical Trials and Observational Studies. John Wiley & Sons, Chichester (1995)

[2] Klein, J.P., Moeschberger, M.L.: Survival Analysis, Techniques for Censored and Truncated Data. Springer, NY (1997)

[3] Cox, D.R.: Regression Model and Life Tables (with Discussion). Journal of the Royal Statistical Society B, 187–220 (1972)

[4] Bobrowski, L.: Ranked modeling of risk on the basis of survival data. In: ICSMRA 2007 - International Conference on Statistical Methods for Risk Analysis, Lisbon (2007)

[5] Bobrowski, L., Łukaszuk, T., Wasyluk, H.: Ranked modeling of causal se-quences of diseases for the purpose of early diagnosis. In: Kącki, E., Rudnicki, M., Stempczyńska, J. (eds.) Computers in Medical Activity. Advances in Intelli-gence and Soft Computing, vol. 65, pp. 23–31. Springer, Heidelberg (2009)

[6] Bobrowski, L.: Ranked linear models and sequential patterns recognition. Pattern Analysis & Applications 12(1), 1–7 (2009)

[7] Bobrowski, L., Łukaszuk, T.: Feature selection based on relaxed linear separabilty. Biocybernetics and Biomedial Engineering 29(2), 43–59 (2009)

[8] Duda, O.R., Hart, P.E., Stork, D.G.: Pattern Classification. J. Wiley, New York (2001)

[9] Bobrowski, L.: Eksploracja danych oparta na wypukłych i odcinkowo-liniowych funkcjach kryterialnych (Data mining based on convex and piecewise linear (CPL) criterion func-tions), Technical University Białystok (in Polish) (2005)

[10] Bobrowski, L., Niemiro, W.: A method of synthesis of linear discriminant function in the case of nonseparabilty. Pattern Recognition 17, 205–210 (1984)

[11] Johnson, R.A., Wichern, D.W.: Applied Multivariate Statistical Analysis. Prentice- Hall, Inc., Englewood Cliffs (1991)

Medical Datasets Analysis: A Constructive Induction Approach

Wiesław Paja and Mariusz Wrzesień

University of Information Technology and Management,
Institute of Biomedical Informatics, Sucharskiego 2, 35-225 Rzeszow, Poland
{WPaja,MWrzesien}@wsiz.rzeszow.pl

Abstract. The main goal of our research was to compile new methodology for building simplified learning models in a form of decision rule set. Every investigated source informational dataset was extended by application of constructive induction method to get a new, additional, descriptive attribute, and then sets of decision rules were developed for source and for extended database, respectively. In the last step, obtained set of rules were optimized and compared to earlier set of rules.

Keywords: Constructive induction, decision rule, medical dataset, melanocytic skin lesion, mental disease, heart disease.

1 Introduction

The main task of data mining is to assist users in extracting useful information or knowledge, which is usually represented as a form of rule due to its easy understandability and interpretability, from the rapidly growing volumes of data [2]. Among various techniques in data mining, classification rule mining are one of the major and traditional.

The problem of knowledge representation by means of decision rules is an important issue in many areas of machine learning domain. Decision rules have simple and understandable structure, however, in many practical applications – even of quite trivial origin – the number of rules in learning models can be disastrous. One can easily imagine that for example, the learning model developed to distinguish between flu and pneumonia, but consisting, say, 10 000 rules, will never be accepted by medical doctors. For this reason our research were devoted to the development of learning models displaying high efficiency and - at the same time – consisting of possibly low number of rules.

2 Theoretical Background

Inductive learning algorithms used commonly for development of sets of decision rules can cause the appearance of some specific anomalies in learning models [10]. This anomalies can be grouped as follows [6]:

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 442–449, 2010.

- *redundancy*: identical rules, subsumed rules, equivalent rules, unusable rules,
- *consistency*: ambiguous rules, conflict rules, rules with logical inconsistency,
- *reduction*: reduction of rules, canonical reduction of rules, specific reduction of rules, elimination of unnecessary attributes,
- *completeness*: logical completeness, specific (physical) completeness, detection of incompleteness, and identification of missing rules.

These irregularities in learning models can be fixed (and sometimes removed) using some schemes generally known as verification and validation procedures [12]. Validation tries to establish the correctness of a system with respect to its use for a particular domain and environment. In short, we can agree that validation is interpreted as "building the right product", whereas verification as "building the product right". It has been argued that the latter is a prerequisite and subtask of the former.

Analysis of available literature suggest the expectation that application of the constructive induction mechanism over a source information database, prior to development of rule sets, can lead to more effective learning models. Constructive induction idea was introduced by Michalski, later four types of the methodology was proposed [11]: data-driven constructive induction (DCI), hypothesis-driven constructive induction (HCI), knowledge-driven constructive induction (KCI), multistrategy constructive induction (MCI).

An impact of constructive induction methods on data mining operations is described in details in [9].

3 Investigated Datasets

The three different medical datasets were used in this research. The first one concerns melanocytic skin lesion which is a very serious skin and lethal cancer. It is a disease of contemporary time, the number of melanoma cases is constantly increasing, due to, among other factors, sun exposure and a thinning layer of ozone over the Earth, [2]. Statistical details on this data are given in [6]. Descriptive attributes of the data were divided into four categories: *Asymmetry*, *Border*, *Color*, and *Diversity* of structures (further called for short Diversity). The variable *Asymmetry* has three different values: *symmetric spot*, *one axial asymmetry* and *two axial asymmetry*. *Border* is a numerical attribute with values from *0* to *8*. *Asymmetry* and *Border* are single-value attributes. The remaining two attributes, *Color* and *Diversity*, are multivalent attributes. *Color* has six possible values: *black*, *blue*, *dark brown*, *light brown*, *red* and *white*. Similarly, *Diversity* has five values: *pigment dots*, *pigment globules*, *pigments network*, *structureless areas* and *branched streaks*. Therefore, we introduced six single-valued variables describing color and five single-valued variables describing diversity of structure. In all of these *11* attributes the values are *0* or *1*, *0* means lack of the corresponding property and *1* means the occurrence of the property. This dataset consists of 548 cases diagnosed by medical specialists using histopathological tests. All cases are assigned into four decision classes: *benign nevus*, *blue nevus*, *suspicious melanoma* and *melanoma malignant*.

The second dataset, with mental diseases cases, contains description of patients that were examined using the ***M****innesota* ***M****ultiphasic* ***P****ersonality* ***I****nventory* (MMPI) from the psychic disturbances perspective. Examination results are presented in the form of profile. Patient's profile is a data vector consisting of fourteen attributes. More exactly, a data vector consists of three parts:

- *Validity* part (validity scales): *lie, infrequency, correction*;
- *Clinical* part (clinical scales): *hypochondriasis, depression, hysteria, psychopathic deviate, masculinity-femininity, paranoia, psychasthenia, schizophrenia, hypomania, social introversion*;
- *Group* part – a class to which the patient is classified.

Dataset consists of over *1700* cases classified by clinic psychologist. Each case is assigned to one of *20* classes. Each class corresponds to one of nosological type: *norm, neurosis, psychopathy, organic, schizophrenia, syndrome delusion, reactive psychosis, paranoia, manic state, criminality, alcoholism, drag induction, simulation, dissimulation, deviational answering style 1, deviational answering style 2, deviational answering style 3, deviational answering style 4, deviational answering style 5, deviational answering style 6*.

The third dataset is a one of medical datasets available at *UCI Machine Learning Repository*. It is called *Heart Disease Data Set*. This database contains *76* attributes, but in this research like in most of published experiments refer to using a subset of *14* of them: *Age, Sex, Chest Pain Type, Resting Blood Pressure, Serum Cholestoral, Fasting Blood Sugar, Resting Electrocardiographic, Maximum Heart Rate, Exercise Induced Angina, Oldpeak, Slope Of The Peak, Number Of Major Vessels, Thal, Class*. The decision field refers to the presence of heart disease in the patient. More detailed information about this data are given on the UCI repository website.

Some summary characteristics of investigated datasets are presented in table below.

Table 1. Datasets characteristics

Data Set	*Number of instances*	*Number of attributes*	*Number of classes*	*Attribute type*
Melanocytic skin lesions	548	14	4	categoric
Mental disease	1705	14	20	numeric
Heart disease	303	14	2	categoric numeric

4 Methodology Used

Two pathways were used in the research (see Fig. 1). One of them was devoted to set of rules generated from primary source of knowledge (i.e. standard decision table), it is denoted as learning model (1). However, in the second path the decision table was expanded by inclusion of a new, additional attribute obtained by means of constructive induction mechanism. In both pathways decision rules were developed using an

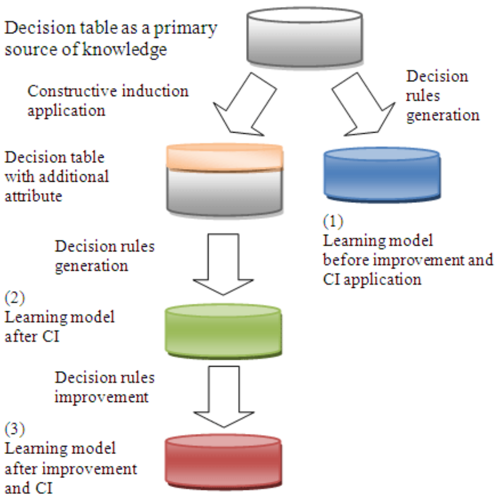

Decision table as a primary source of knowledge
Constructive induction application
Decision rules generation
Decision table with additional attribute
(1) Learning model before improvement and CI application
Decision rules generation
(2) Learning model after CI
Decision rules improvement
(3) Learning model after improvement and CI

Fig. 1. Methodology used in the research

in house developed algorithm GTS (***G**eneral-**T**o-**S**pecific*) [3]. Obtained in this way sets of rules (denoted as learning model (2)) were improved using *Rule*SEEKER system [7]. All three learning models were then evaluated – via testing the classification accuracy of unseen cases.

4.1 Brief Description of a New Constructive Induction Algorithm

Our new constructive induction algorithm, called **CIBN** (***C**onstructive **I**nduction based on **B**elief **N**etworks*) creates a new descriptive attribute (a new column in the decision table). In the process of expansion of the source decision table, a new column-vector is created; its all cells are filled with numerical values obtained in the following way: to create a new attribute, the information from belief network [5] (generated for investigated dataset) is used (see Fig. 2). The approach is limited to numerical, thus, in the first step all categorical or nominal variables were converted to numerical form. In the process of construction *Naive Bayes* classifier was used.

Descriptive attributes used in the original (source) decision table affect a course of the development cell-values of the new column-vector, according to the general formula (1):

$$\begin{gathered} Cell\_value_{n,constr.ind.} = Factor\_1 \cdot V_{A1,n} + Factor\_1 \cdot V_{A3,n} + \\ + Factor\_2 \cdot V_{A1,n} + Factor\_2 \cdot V_{A4,n} + Factor\_3 \cdot V_{A5,n} \end{gathered} \tag{1}$$

where:

Factor_N – is a coefficient related with a level of the structure of belief network, $V_{AN,n}$ – is a value of a cell in the *n*-th case for *N*-th attribute.

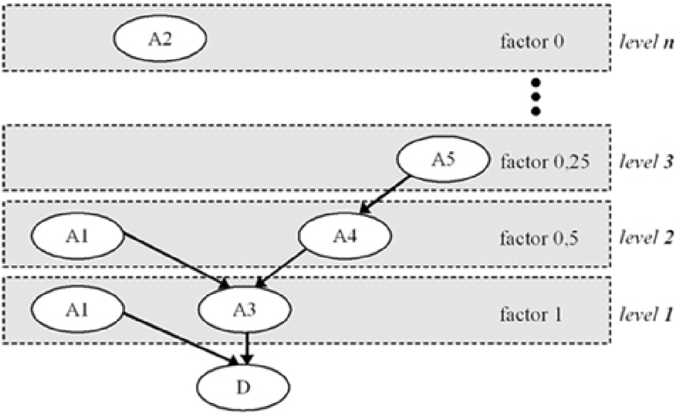


Fig. 2. Belief network scheme - A1, A2...A5 – description attributes, D – decision attribute

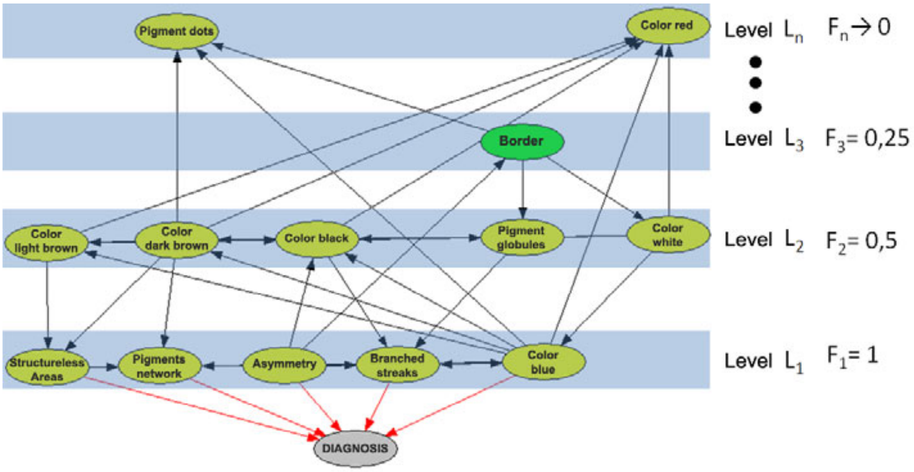


Fig. 3. Example of Bayesian network created on the basis of melanocytic skin lesions dataset

On figure 3 the example of CIBN algorithm application is shown. There is example of Bayesian network created on the basis of source melanocytic skin lesions dataset. As it is stated on Fig. 3 the most significance, direct influence on the diagnosis attribute have five attributes on level L_1: *Structureless areas, Pigments network, Asymmetry, Branched streaks* and *Color blue*. Thus, these attributes have factor equal to 1 in constructive induction process. Similarly, the next five attributes, on level L_2, have indirect influence on decision attribute, in that way, these attributes have factor equal to *0.5*, etc. Using this schema a new attribute (called by us *CI* attribute) could be calculated (see equation 2). So, for example the attribute *Structureless areas* has factor equal to *1+0.5* due to direct influence on *Diagnosis* and indirect influence by *Pigment network* attribute.

$$\begin{aligned}CI = {} & 4.125 \cdot Asymmetry + 1.375 \cdot Border + 3.625 \cdot Color\ white + 4.25 \cdot \\ & Color\ blue + 2.625 \cdot Color\ dark\ brown + 1.25 \cdot Color\ light\ brown + \\ & 1.25 \cdot Color\ black + 0 \cdot Color\ red + 0.75 \cdot Pigment\ globules + 0 \cdot \\ & Pigment\ dots + 1.5 \cdot Structureless\ areas + 1.5 \cdot \\ & Branched\ streaks + 1 \cdot Pigment\ network\end{aligned} \tag{2}$$

This algorithm was implemented in *Plane*SEEKER system [1], and next analysis on selected information dataset was performed.

4.2 Short Description of the Optimizing Algorithm

The main optimizing algorithm applied in the research was implemented in the system *Rule*SEEKER [8], and was based on an exhaustive application of a collection of generic operations:

- *finding and removing redundancy*: the data may be overdetermined, that is, some rules may explain the same cases. Here, redundant (excessive) rules were analyzed, and the redundant rule (or some of the redundant rules) was (were) removed, provided this operation did not increase the error rate;
- *finding and removing of incorporative rules*, another example when the data may be overdetermined. Here, some rule(s) being incorporated by another rule(s) were analyzed, and the incorporative rule(s) was (were) removed, provided this operation did not increase the error rate;
- merging rules: in some circumstances, especially when continuous attributes were used for the description of objects being investigated, generated learning models contained rules that are more specific than they should be. In these cases, more general rule(s) were applied, so that they cover the same investigated cases, without making any incorrect classifications;
- *finding and removing of unnecessary rules*: sometimes rules developed by the systems used were unnecessary, that is, there were no objects classificated by this rules. Unnecessary rule(s) was (were) removed, provided this operation did not increase the error rate;
- *finding and removing of unnecessary conditions*: sometimes rules developed by the systems used contain unnecessary conditions, that were removed, provided this operation did not increase the error rate;
- *creating of missing rules*: sometimes developed models didn't classify all cases from learning set. Missing rules were generated using a set of unclassified cases;
- *discovering of hidden rules*: this operation generates a new rule by combination of similar rules, containing the same set of attributes and the same – except one – attribute values;
- *rule specification*: some rules caused correct and incorrect classifications of selected cases, this operation divides considered rule into few rules by adding additional conditions;
- *selecting of final set of rules*: there were some rules that classify the same set of cases but have different composition, simpler rule stayed in a set.

5 Results of Experiments

The results of improvement of learning models are gathered in Table 2. All experiments were performed using well known 10-fold cross validation method. The column, indexed by (1), contains the basic information about each learning model (in the form of set of decision rules) developed for the source database. Next column denoted as (2) presents information after application of constructive induction method, e.g. after adding of *CI* attribute. The last column denoted as (3) contains results after adding *CI* and optimization of learning model.

Table 2. Average number of rules and error rate of the developed learning models

Characteristic of learning model	*Data set*	*(1)*	*(2)*	*(3)*
Number of rules	Melanocytic skin lesions	153	92	48
	Mental disease	508	497	385
	Heart disease	125	131	88
Error rate [%]	Melanocytic skin lesions	14.29	7.69	7.14
	Mental disease	39.24	27.33	22.85
	Heart disease	44.44	32.94	30.74

As it is stated in table 2, all datasets were improved by decreasing of number of rules and error rate. The greatest influence of experiments are observed in case of *Melanocytic skin lesions* dataset. This model, contained rather large number of rules (153), and the error rate was on the level of ~14%. Just after application of constructive induction algorithm (column(2)), the number of rules dropped roughly 40%, and additionally – what is very interesting – also the error rate was distinctly decreased (from 14.29% to 7.69%). In the next step of the developed methodology (see column (3)), the optimization algorithm (see paragraph 4.2), the set of decision rules was smaller (92 rules vs. 48 rules). It means, that the decrease of number of rules was even larger, i.e. about 48%. In case of *Mental disease* dataset similar results are observed. Number of rules decreased about 20%, and in the same time, the error rate decreased nearly half. Similar results could be found in case of third dataset devoted to *Heart diseases*.

6 Conclusions

It should be stressed, that the truncation of the learning model did not spoil its efficiency; the error rate is much smaller. Thus, it may be assumed that the combined algorithm of constructive induction, based on data taken from belief networks and some improvement of decision rule sets performed quite satisfactorily in classification of selected medical datasets. In the future a comparison of calculated *CI* attribute in case of melanocytic skin lesions dataset with well known in medicine TDS parameter is going to be done. Presumably using this method it could be possible to find some general parameter combined using other descriptive attributes for every investigated datasets.

Another important issues is to compare proposed method against an approach that adds a new attribute constructed with a lineal function approximation of the class and also how it compares against other rule-based classifiers and how these rule-based systems are affected by the inclusion of the new attribute. It would be investigated in future research.

References

1. Blajdo, P., Grzymala-Busse, J.W., Hippe, Z.S., Knap, M., Marek, T., Mroczek, T., Wrzesien, M.: A suite of machine learning programs for data mining: chemical applications. In: Debska, B., Fic, G. (eds.) Information Systems in Chemistry 2, University of Technology Editorial Office, Rzeszow, pp. 7–14 (2004)
2. Friedman, R.J., Rigel, D.S., Kopf, A.W.: Early detection of malignant melanoma: the role of physician examination and self-examination of the skin. CA Cancer J. Chim. 35, 319–331 (1996)
3. Hippe, Z.S.: Machine Learning – a Promising Strategy for Business Information Systems? In: Abramowicz, W. (ed.) Business Information Systems 1997, pp. 603–622. Academy of Economics, Poznan (1997)
4. Hippe, Z.S., Bajcar, S., Blajdo, P., Grzymala-Busse, J.P., Grzymala-Busse, J.W., Knap, M., Paja, W., Wrzesien, M.: Diagnosing Skin Melanoma: Current versus Future Directions. TASK Quarterly 7(2), 289–293 (2003)
5. Jensen, F.V.: Bayesian Networks and Decision Graphs. Springer, Heidelberg (2001)
6. Ligeza, A.: Logical Foundations for Rule-Based Systems. Springer, Heidelberg (2006)
7. Liu, H., Sun, J., Zhang, H.: Post-processing of associative classification rules using closed sets. Expert Systems with Application 36, 6659–6667 (2009)
8. Paja, W.: RuleSEEKER – a New System to Manage Knowledge in form of Decision Rules. In: Tadeusiewicz, R., Ligeza, A., Szymkat, M. (eds.) Computer Methods and Systems, Ed. Office, ”Oprogramowanie Naukowo-Techniczne”, Cracow, pp. 367–370 (2005) (in polish)
9. Ram, A., Santamaria, J.C.: Continuous Case-Based Reasoning. Artificial Intelligence 90, 25–77 (1997)
10. Spreeuwenberg, S., Gerrits, R.: Requirements for successful verification in practice. In: Susan, M.H., Simmons, G. (eds.) Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference 2002, Pensacola Beach, Florida, USA, May 14-16, AAAI Press, Menlo Park (2002)
11. Wnek, J., Michalski, R.S.: Hypothesis-driven Constructive Induction in AQ17-HCI: A Method and Experiments. Machine Learning 14(2), 139–168 (1994)
12. Gonzales, A.J., Barr, V.: Validation and verification of intelligent systems. Journal of Experimental & Theoretical Artificial Intelligence 12, 407–420 (2000)

Regression Models for Spatial Data: An Example from Precision Agriculture

Georg Ruß and Rudolf Kruse

Otto-von-Guericke-Universität Magdeburg
georg.russ@ieee.org

Abstract. The term *precision agriculture* refers to the application of state-of-the-art GPS technology in connection with small-scale, sensor-based treatment of the crop. This data-driven approach to agriculture poses a number of data mining problems. One of those is also an obviously important task in agriculture: yield prediction. Given a precise, geographically annotated data set for a certain field, can a season's yield be predicted?

Numerous approaches have been proposed to solving this problem. In the past, classical regression models for non-spatial data have been used, like regression trees, neural networks and support vector machines. However, in a cross-validation learning approach, issues with the assumption of statistical independence of the data records appear. Therefore, the geographical location of data records should clearly be considered while employing a regression model. This paper gives a short overview about the available data, points out the issues with the classical learning approaches and presents a novel spatial cross-validation technique to overcome the problems and solve the aforementioned yield prediction task.

Keywords: Precision Agriculture, Data Mining, Regression, Modeling.

1 Introduction

In recent years, information technology (IT) has become more and more part of our everyday lives. With data-driven approaches applied in industry and services, improvements in efficiency can be made in almost any part of nowadays' society. This is especially true for agriculture, due to the modernization and better affordability of state-of-the-art GPS technology. A farmer nowadays harvests not only crops but also growing amounts of data. These data are precise and small-scale – which is essentially why the combination of GPS, agriculture and data has been termed *precision agriculture*.

In those agriculture (field) data, often a large amount of information is contained, yet hidden. This is usually information about the soil and crop properties enabling a higher operational efficiency – appropriate techniques should therefore be applied to find this information. This is a common problem for which the term *data mining* has been coined. Data mining techniques aim at finding those patterns or information in the data that are both valuable and interesting to the farmer.

A specific problem commonly occurring is *yield prediction*. As early into the growing season as possible, a farmer is interested in knowing how much yield he is about to expect. The ability to predict yield used to rely on farmers' long-term knowledge of particular fields, crops and climate conditions. However, this knowledge can be expected

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 450–463, 2010.

to be available in the data collected during normal farming operations throughout the season. A multitude of sensor data are nowadays collected, measuring a field's heterogeneity. These data are precise, often highly correlated and carry spatial information which must not be neglected.

Hence, the problem of yield prediction encountered can be treated as a problem of data mining and, specifically, multi-dimensional regression. This article will serve as a reference of how to treat a regression problem on spatial data with a combination of classical regression techniques and a number of novel ideas. This article will furthermore serve as a continuation of [17]: in the previous article, the spatial data were treated with regression models which do not take the spatial relationships into account. The current work aims to check the validity of the *statistical independence* assumption inherent in classical regression models in conjunction with spatial data. Based upon the findings, spatial regression will be carried out using a novel clustering idea during a cross-validation procedure. The results will be compared to those obtained while neglecting the spatial relationships inherent in the data sets.

1.1 Research Target

The main research target of this work is to improve and further substantiate the validity of *yield prediction* approaches using multi-dimensional regression modeling techniques. Previous work, mainly the regression work presented in [17,21], will be used as a baseline for this work. Some of the issues of the previous approach will be clearly pointed out in this article. Nevertheless, this work aims to improve upon existing yield prediction models and, furthermore, incorporates a generic, yet novel spatial clustering idea into the process. Therefore, different types of regression techniques will be incorporated into a novel spatial cross-validation framework. A comparison of using spatial vs. non-spatial data sets shall be presented.

1.2 Article Structure

This article will start with a brief introduction into the area of precision agriculture and a more detailed description of the available data in Section 2. This will be followed by an outline of the key techniques used in this work, embedded into a data mining workflow presented in Section 3. The results obtained during the modeling phase will be presented in Section 4. The article will be completed with a short conclusion in Section 5, which will also point out further lines of research.

2 Data Description

With the recent advances in technology, ever larger amounts of data are nowadays collected in agriculture during standard farming operations. This section first gives a short categorization of the data into four classes. Afterwards, the actual available data are presented. The differences between spatial and non-spatial data are pointed out.

2.1 Data Categorization

A commonality among data collected in agriculture is that every data record has a spatial location on the field, usually determined via (differential) GPS with a high degree of precision. These data can roughly be divided into four classes as follows:

Yield Mapping has been a standard approach for many years. Based on maps of previous years' yields, recommendations of farming operations for the current season are determined.

Topography is often considered a valuable feature for data mining in agriculture. The spatial location of data points (longitude, latitude) is a standard variable to be used in spatial modeling. Furthermore, variables like elevation, slope and derivatives of those values can be obtained easily.

Soil Sampling is a highly invasive means of acquiring data about a field. Furthermore, it is labour-intensive and therefore rather expensive. Obtaining a high resolution of soil sampling data therefore requires lots of effort. From soil sampling, variables like organic matter, available minerals, water content etc. can be derived.

Remote Sensing recently has become a rather cheap and high-resolution data source for data-driven agricultural operations. It usually consists of aerial or satellite imaging using multiple spectral bands at different times into the vegetation period. From those images, vegetation indices are derived and used for assessing the crop status.

2.2 Available Data

The data available in this work were collected during the growing season of 2007 on two fields north of Köthen, Germany. The data for the two fields, called *F440* and *F611*, respectively, were interpolated using kriging [23] to a grid with 10 by 10 meters grid cell sizes. Each grid cell represents a record with all available information. The fields grew winter wheat, where nitrogen fertilizer was distributed over three application times during the growing season.

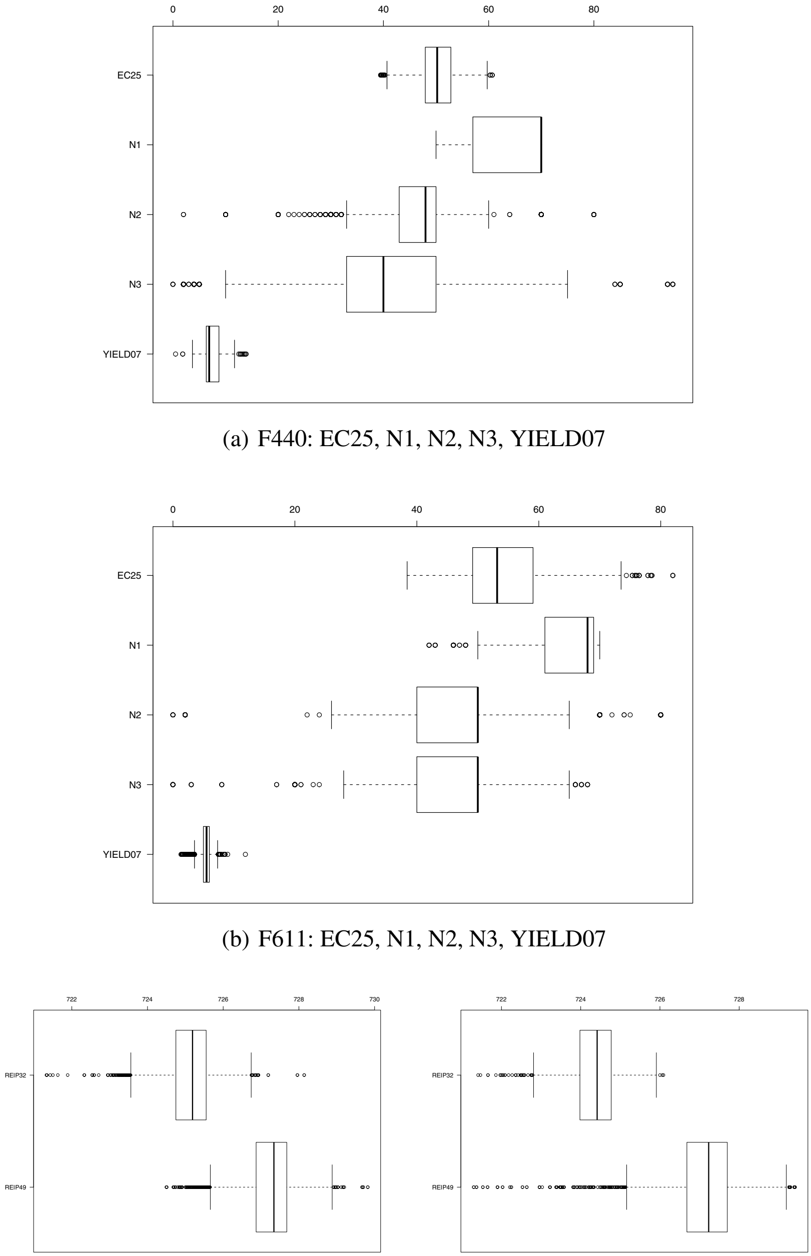
Overall, for each field there are six input attributes – accompanied by the respective current year's yield (2007) as the target attribute. Those attributes will be described in the following. In total, for the F440 field there are 6446 records, for F611 there are 4970 records, thereof none with missing values and none with outliers. A short statistical summary of the fields and variables can be found in Figure 1. In the following sections, further details about the individual attributes is provided.

2.3 YIELD07

Here, yield is measured in metric tons per hectare ($\frac{t}{ha}$). For the yield ranges for the respective years and sites, see Figures 1(a) and 1(b).

2.4 Apparent Electric Conductivity – EC25

A non-invasive method to discover and map a field's heterogeneity is to measure the soil's apparent electrical conductivity. It is assumed that the EC25 readings are closely related to soil properties which would otherwise have to be sampled in a time-consuming



(a) F440: EC25, N1, N2, N3, YIELD07

(b) F611: EC25, N1, N2, N3, YIELD07

(c) F440: REIP32, REIP49

(d) F611: REIP32, REIP49

Fig. 1. Statistical Summary for the two available data sets (F440, F611)

and expensive manner. Commercial sensors such as the EM-38[1] are designed for agricultural use and can measure small-scale conductivity to a depth of about 1.5 metres. There is no possibility of interpreting these sensor data directly in terms of its meaningfulness as yield-influencing factor. But in connection with other site-specific data, as explained in the rest of this section, there could be coherences. For a more detailed analysis of this particular sensor, see, e.g. [5]. For the range of EC25 values encountered in the available data, see Figures 1(a) and 1(b).

2.5 Vegetation – REIP32, REIP49

The *red edge inflection point* (REIP) is a second derivative value calculated along the red edge region of the spectrum, which is situated from 680 to 750nm. Dedicated REIP sensors are used in-season to measure the plants' reflection in this spectral band. Since the plants' chlorophyll content is assumed to highly correlate with the nitrogen availability (see, e.g. [13]), the REIP value allows for deducing the plants' state of nutrition and thus, the previous crop growth. For further information on certain types of sensors and a more detailed introduction, see [9] or [24]. Plants that have less chlorophyll will show a lower REIP value as the red edge moves toward the blue part of the spectrum. On the other hand, plants with more chlorophyll will have higher REIP values as the red edge moves toward the higher wavelengths. Obviously, later into the growing season the plants are expected to have a higher chlorophyll content, which can easily be assessed by visually comparing the REIP values in Figures 1(c) and 1(d). The numbers in the REIP32 and REIP49 names refer to the growing stage of winter wheat, as defined in [11].

2.6 Nitrogen Fertilizer – N1, N2, N3

The amount of fertilizer applied to each subfield can be measured easily. Since it is a variable that can and should be influenced by the farmer, it does not appear in the preceding categorization. Fertilizer is applied at three points in time into the vegetation period, which is the standard strategy for most of Northwestern Europe [15]. The ranges in the data sets can be obtained from Figures 1(a) and 1(b).

2.7 Spatial vs. Non-spatial Data Treatment

According to [7], *spatial autocorrelation* is the correlation among values of a single variable strictly attributable to the proximity of those values in geographic space, introducing a deviation from the independent observations assumption of classical statistics. Given a spatial data set, spatial autocorrelation can be determined using Moran's I ([14]) or semivariograms. Spatial autocorrelation appears in such diverse areas as econometrics [1], geostatistics [6] and social sciences [10], among others. In practice, it is usually also known from the data configuration whether spatial autocorrelation is existent. For further information it is referred to, e.g., [6].

In previous articles using the above data, such as [17,21], the main focus was on finding a suitable regression model to predict the current year's yield sufficiently well.

[1] Trademark of Geonics Ltd, Ontario, Canada.

However, it should be noted that the used regression models, such as neural networks [18,19] or support vector regression [17], among others, usually assume statistical independence of the data records. However, with the given geo-tagged data records at hand, this is clearly not the case, due to (natural) spatial autocorrelation. Therefore, the spatial relationships between data records have to be taken into account. The following section further elaborates upon this topic in detail.

3 Regression Techniques on Spatial Data

Based on the findings at the end of the preceding section, this section will present a novel regression model for data sets which exhibit spatial autocorrelation. In classical regression models, data records which appear in the training set must not appear in the test set during a cross-validation learning setup. Due to classical sampling methods which do not take spatial neighborhoods of data records into account, this assumption may be rendered invalid when using non-spatial models on spatial data. This leads to overfitting (overlearning) and underestimates the true prediction error of the regression model. Therefore, the core issue is to avoid having neighboring or the same samples in training and testing data subsets during a cross-validation approach.

As should be expected, the data sets F440 and F611 exhibit spatial autocorrelation. Therefore, classical regression models must either be swapped against different ones which take spatial relationships into account or may be adapted to accommodate spatial data. In order to keep standard regression modeling techniques such as neural networks, support vector regression, bagging, regression trees or random forests as-is, a meta-approach will be presented in the following. In a nutshell, it replaces the standard sampling approach of the cross-validation process with an approach that is aware of spatial relationships.

3.1 From Classical to Spatial Cross-Validation

Traditionally, k-fold cross-validation for regression randomly subdivides a given data set into two (without validation set) or three parts: a training set, a validation set and a test set. A ratio of 6:2:2 for these sets is usually assumed appropriate. The regression model is trained on the training set until the prediction error on the validation set starts to rise. Once this happens, the training process is stopped and the error on the test set is reported for this fold. This procedure is repeated k times, with the root mean squared error (RMSE) often used as a performance measure.

The issue with spatial data is that, due to spatial autocorrelation, almost identical data records may end up in training and test set, such that the model overfits the data and underestimates the error. Therefore, one possible solution might be to ensure that only a very small number (if any) of neighboring and therefore similar samples end up in training and test subsets. This may be achieved by adapting the sampling procedure for spatial data. Once this issue has been accommodated, the cross-validation procedure may continue as-is. A rather straightforward approach using the geo-tagged data is described in the following.

3.2 Employing Spatial Clustering for Data Sampling

Given the data sets F440 and F611, a spatial clustering procedure can be employed to subdivide the fields into spatially disjunct clusters or zones. The clustering algorithm can easily be run on the data map, using the data records' longitude and latitude. Depending on the clustering algorithm parameters, this results in a tesselation map which does not consider any of the attributes, but only the spatial neighborhood between data records. A depiction of this clustering process can be found in Figures 2(a) and 2(b). Standard k-means clustering was used with a setting of $k = 20$ clusters per field for demonstration purposes. In analogy to the non-spatial regression treatment of these data records, now a spatially-aware cross-validation regression problem can be handled using the k zones of the clustering algorithm as an input for k-fold cross-validation. Standard models, as described below, can be used straightforwardly, without requiring changes to the models themselves. The experimental setup and the results are presented in the following section.

It should be noted that this spatial clustering procedure is a broader definition of the standard cross-validation setup. This can be seen as follows: when refining the clustering further, the spatial zones on the field become smaller. The border case is reached when the field is subdivided into as many clusters as there are data records, i.e. each data record describes its own cluster. In this special case, the advantages of spatial clustering are lost since no spatial neighborhoods are taken into account in this approach. Therefore, the number of clusters should be seen as a tradeoff between precision and statistical validity of the model.

3.3 Regression Techniques

In previous work ([17,21]), numerous regression modeling techniques have been compared on similar data sets to determine which of those modeling techniques works best. Although those models were run in a non-spatial regression setup, it is assumed that the relative differences between these models will also hold in a spatial cross-validation regression setup. In the aforementioned previous work, support vector regression has been determined as the best modeling technique when comparing the models' root mean squared prediction error. Hence, in this work support vector regression will serve as a benchmark technique against which further models will have to compete. Experiments are conducted in R [16], a link to the respective scripts is provided in Section 5.

Support Vector Regression. Support Vector Machines (SVMs) are a supervised learning method discovered by [2]. However, the task here is regression, so the focus is on support vector regression (SVR) in the following. A more in-depth discussion can be found in [8]. Given the training set, the goal of SVR is to approximate a linear function $f(x) = \langle w,x \rangle + b$ with $w \in \mathbb{R}^N$ and $b \in \mathbb{R}$. This function minimizes an empirical risk function defined as

$$R_{emp} = \frac{1}{N} \sum_{i=1}^{N} L_\varepsilon(\hat{y} - f(x)), \tag{1}$$

where $L_\varepsilon(\hat{y} - f(x)) = \max((|\xi| - \varepsilon), 0)$. $|\xi|$ is the so-called slack variable, which has mainly been introduced to deal with otherwise infeasible constraints of the optimization problem, as has been mentioned in [22]. By using this variable, errors are basically ignored as long as they are smaller than a properly selected ε. The function here is called ε-insensitive loss function. Other kinds of functions can be used, some of which are presented in chapter 5 of [8].

To estimate $f(x)$, a quadratic problem must be solved, of which the dual form, according to [12] is as follows:

$$max_{\alpha,\alpha^*} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)K(x_i,x_j) - \varepsilon\sum_{i=j}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i - \alpha_i^*) \quad (2)$$

with the constraint that $\sum_{j=1}^{N}(\alpha_i - \alpha_i^*) = 0, \alpha_i, \alpha_i^* \in [0,C]$. The regularization parameter $C > 0$ determines the tradeoff between the flatness of $f(x)$ and the allowed number of points with deviations larger than ε. As mentioned in [8], the value of ε is inversely proportional to the number of support vectors. An adequate setting of C and ε is necessary for a suitable solution to the regression problem.

Furthermore, $K(x_i,x_j)$ is known as a kernel function which allows to project the original data into a higher-dimensional feature space where it is much more likely to be linearly separable. Some of the most popular kernels are radial basis functions (equation 3) and a polynomial kernel (equation 4):

$$K(x,x_i) = e^{-\frac{||x-x_i||^2}{2\sigma^2}} \quad (3)$$

$$K(x,x_i) = (\langle x,x_i \rangle + 1)^{\rho} \quad (4)$$

The parameters σ and ρ have to be determined appropriately for the SVM to generalize well. This is usually done experimentally. Once the solution for the above optimization problem in equation 2 is obtained, the support vectors can be used to construct the regression function:

$$f(x) = \sum_{i=1}^{N}(\alpha_i - \alpha_i^*)K(x,x_i) + b \quad (5)$$

In the current experiments, the *svm* implementation from the *e1071* R package has been used.

Random Forests and Bagging. In previous work ([17]), one of the presented regression techniques were regression trees. They were shown to be rather successful, albeit in a non-spatial regression setup. Therefore, this article considers an extension of regression trees: random forests. According to [4], random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. In the version used here, the random forest is used as a regression technique. Basically, a random forest is an ensemble method that consists of many regression trees and outputs a combined result of those trees as a prediction for the target variable. Usually, the generalization error for forests converges to a limit as the number of trees in the forest becomes large.

Let the number of training cases be N and the number of variables in the regression task be M. Then, each tree is constructed using the following steps:

1. A subset with size m of input variables is generated. This subset is used to determine the decision at a node of the tree; $m \ll M$.
2. Take a bootstrap sample for this tree: choose N times with replacement from all N available training cases. Use the remaining cases to estimate the tree's regression error.
3. Randomly choose m variables from which to derive the regression decision at that node; repeat this for each node of the tree. Calculate the best tree split based on these m variables from the training set.

It should be noted that each tree is fully grown and not pruned. This is a difference from normal regression tree construction. Random forests mainly implement the key ideas from bagging, which is therefore explained in the following.

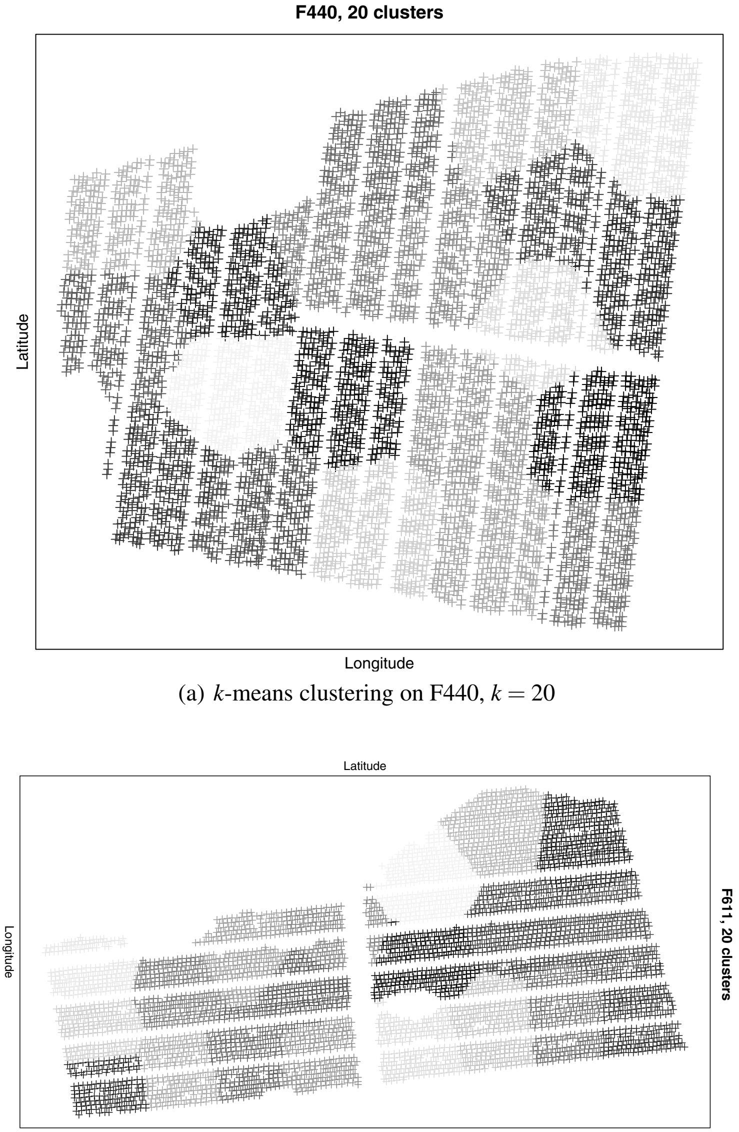
Bootstrap aggregating (or bagging) has first been described in [3]. It is generally described as a method for generating multiple versions of a predictor and using these for obtaining an aggregate predictor. In the regression case, the prediction outcomes are averaged. Multiple versions of the predictor are constructed by taking bootstrap samples of the learning set and using these as new learning sets. Bagging is generally considered useful in regression setups where small changes in the training data set can cause large perturbations in the predicted target variables. Since random forests are a special case of bagging where regression trees are used as the internal predictor, both random forests and bagging should deliver similar results. Both techniques are available in the R packages *randomForest* and *ipred*. Running them on the available data sets should therefore deliver similar results, since the bagging implementation in the R *ipred* package internally uses regression trees for prediction as well. Therefore, the main difference between random forests and bagging in this article is that both techniques are implicitly run and reported with different parameters.

Performance Measurement. The performance of the models will be determined using the root mean squared error (RMSE). For the RMSE, first the difference between an actual target value y_a and the model output value y is computed. This difference is squared and averaged over all training examples before the root of the mean value is taken, see equation 6.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=j}^{n}(y_i - y_{a,i})^2} \tag{6}$$

4 Results

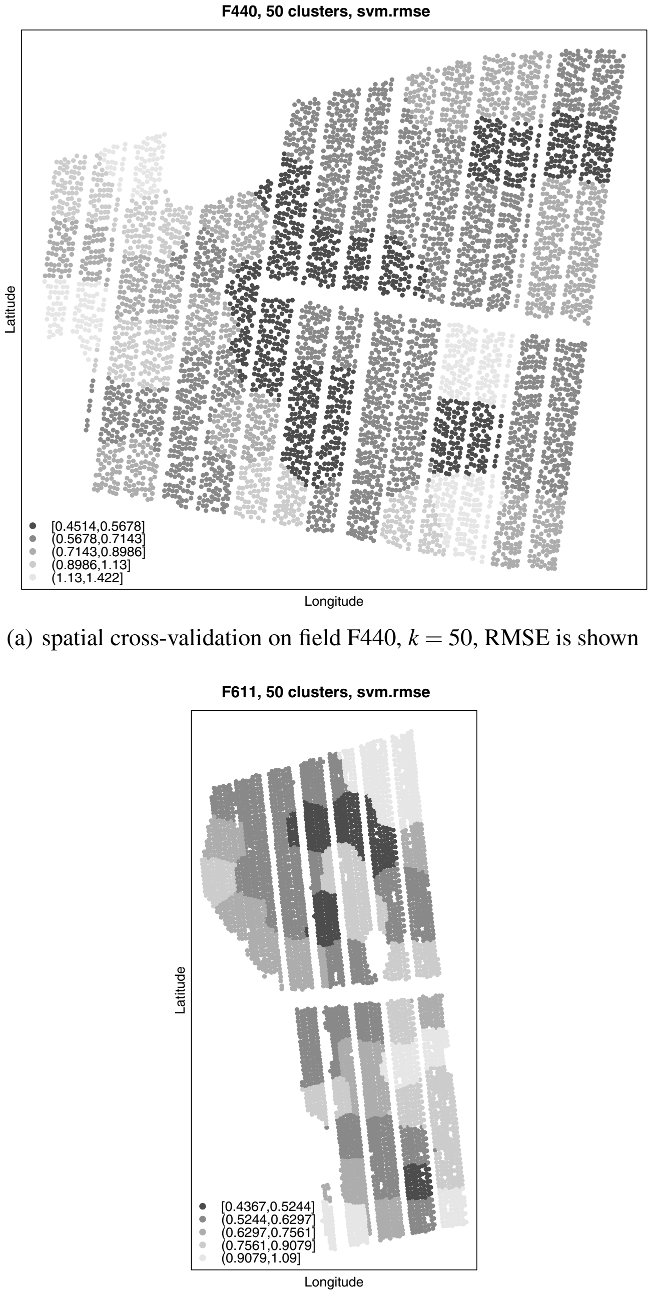
As laid out in the preceding sections, the main research target of this article is to assess whether existing spatial autocorrelation in the data sets may fail to be captured in standard, non-spatial regression modeling setups. The approach consists of a simple comparison between a spatial and a non-spatial setup.



(a) k-means clustering on F440, $k = 20$

(b) k-means clustering on F611, $k = 20$

Fig. 2. k-means clustering on F440 and F611 (the bottom figure has been rotated by 90 degrees)



(a) spatial cross-validation on field F440, $k = 50$, RMSE is shown

(b) spatial cross-validation on field F611, $k = 50$, RMSE is shown

Fig. 3. Results for spatial cross-validation on F440/F611 fields, using 50 clusters and support vector regression

non-spatial setup. The non-spatial setup is similar to the one presented in [17], although different data sets are used. A standard cross-validation procedure is performed, where k is the number of folds. Support vector machines, random forests and bagging are trained on the training set. The squared errors on the test set are averaged and the square root is taken. The resulting value is reported in Table 1.

spatial setup. Since the amount of research effort into spatial data sets is rather sparse when compared to this special setup, a simple, yet effective generic approach has been developed. The spatial data set is clustered into k clusters using the k-means algorithm (see Figure 2). This non-overlapping partitioning of the data set is then used in a spatial cross-validation setup in a straightforward way. This ensures that the number of neighboring data points (which are very similar due to spatial autocorrelation) in training and test sets remains small. The root mean squared error is computed similarly to the non-spatial setup above and may optionally be displayed (see Figure 3).

The results in Table 1 confirm that the spatial autocorrelation inherent in the data set leads classical, non-spatial regression modeling setups to a substantial underestimation of the prediction error. This outcome is consistent throughout the results, regardless of the used technique and regardless of the parameters.

Furthermore, it could be shown that for these particular data sets, random forests or bagging yield more precise predictions than support vector regression. However, the standard settings of the respective R toolboxes were used in both the spatial and the non-spatial setup, therefore the difference between these setups will remain similar regardless of parameter changes. Nevertheless, changes to model parameters might slightly change the outcome of the prediction accuracy and the ranking of the models in terms of root mean squared error. The drawback is that parameter tuning via grid search easily extends computation times by orders of magnitude.

Moreover, the spatial setup can be easily set to emulate the non-spatial setup: set k to be the number of data records in the data set. Therefore the larger the parameter k is

Table 1. Results of running different setups on the data sets F440 and F611; comparison of spatial vs. non-spatial treatment of data sets; root mean squared error is shown, averaged over clusters/folds; k is either the number of clusters in the spatial setup or the number of folds in the non-spatial setup

		F440		F611	
	k	spatial	non-spatial	spatial	non-spatial
Support Vector Regression	10	1.06	0.54	0.73	0.40
	20	1.00	0.54	0.71	0.40
	50	0.91	0.53	0.67	0.38
Random Forest	10	0.99	0.50	0.65	0.41
	20	0.92	0.50	0.64	0.41
	50	0.85	0.48	0.63	0.39
Bagging	10	1.09	0.59	0.66	0.42
	20	1.01	0.59	0.66	0.42
	50	0.94	0.58	0.65	0.41

set, the smaller the difference between the spatial and the non-spatial setup should be. This assumption also holds true for almost all of the obtained results.

5 Conclusions and Future Work

This article presented a central data mining task: regression. Based on two data sets from precision agriculture, a continuation and improvement over previous work ([17,21]) could be achieved. The difference between spatial data and non-spatial data was pointed out. The implications of spatial autocorrelation in these data sets were mentioned. From a statistical and machine learning point of view, neighboring data records in a spatially autocorrelated data sets should not end up in training and test sets since this leads to a considerable underestimation of the prediction error, possibly regardless of the used regression model.

It can be concluded that it is indeed important to closely consider spatial relationships inherent in the data sets. As a suggestion, the following steps should be taken: for those data, the spatial autocorrelation should be determined. If spatial autocorrelation exists, standard regression models must be adapted to the spatial case. A straightforward and illustrative approach using simple *k*-means clustering has been described in this article.

5.1 Future Work

Despite having improved and validated upon the yield prediction task, the data sets carry further information. Two rather interesting task are *variable importance* and *management zones*.

The first refers to the question which of the variables is actually contributing most to the yield prediction task. This has practical implications for the farmers and sensor-producing companies. A first non-spatial approach has been presented in [20] as a standard feature selection approach, which should accommodate the spatial relationships in future implementations. The bagging approach presented in this article might be considered.

The second refers to discovering interesting zones on the (heterogeneous) field which should be managed differently from each other. This is a classical data mining question where the *k*-means approach used in this article is likely to be considered.

Further material, including the R scripts for creating the figures in this article and computing the results, can be found at `http://research.georgruss.de/?cat=24`.

References

1. Anselin, L.: Spatial Econometrics, pp. 310–330. Basil Blackwell, Oxford (2001)
2. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory, pp. 144–152. ACM Press, New York (1992)
3. Breiman, L.: Bagging predictors. Technical report, Department of Statistics, Univ. of California, Berkeley (1994)
4. Breiman, L.: Random forests. Machine Learning 45(1), 5–32 (2001)

5. Corwin, D.L., Lesch, S.M.: Application of soil electrical conductivity to precision agriculture: Theory, principles, and guidelines. Agron J. 95(3), 455–471 (2003)
6. Cressie, N.A.C.: Statistics for Spatial Data. Wiley, New York (1993)
7. Griffith, D.A.: Spatial Autocorrelation and Spatial Filtering. In: Advances in Spatial Science, Springer, New York (2003)
8. Gunn, S.R.: Support vector machines for classification and regression. Technical Report, School of Electronics and Computer Science, University of Southampton, Southampton, U.K (1998)
9. Liu, J., Miller, J.R., Haboudane, D., Pattey, E.: Exploring the relationship between red edge parameters and crop variables for precision agriculture. In: 2004 IEEE International Geoscience and Remote Sensing Symposium, vol. 2, pp. 1276–1279 (2004)
10. Goodchild, M., Anselin, L., Appelbaum, R., Harthorn, B.: Toward spatially integrated social science. International Regional Science Review 23, 139–159 (2000)
11. Meier, U.: Entwicklungsstadien mono- und dikotyler Pflanzen. Biologische Bundesanstalt für Land- und Forstwirtschaft, Braunschweig, Germany (2001)
12. Mejía-Guevara, I., Kuri-Morales, Á.: Evolutionary feature and parameter selection in support vector regression. In: Gelbukh, A., Kuri Morales, Á.F. (eds.) MICAI 2007. LNCS (LNAI), vol. 4827, pp. 399–408. Springer, Heidelberg (2007)
13. Middleton, E.M., Campbell, P.K.E., Mcmurtrey, J.E., Corp, L.A., Butcher, L.M., Chappelle, E.W.: "Red edge" optical properties of corn leaves from different nitrogen regimes. In: 2002 IEEE International Geoscience and Remote Sensing Symposium, vol. 4, pp. 2208–2210 (2002)
14. Moran, P.A.P.: Notes on continuous stochastic phenomena. Biometrika 37, 17–33 (1950)
15. Neeteson, J.J.: Nitrogen Management for Intensively Grown Arable Crops and Field Vegetables, ch. 7, pp. 295–326. CRC Press, Haren (1995)
16. R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2009) ISBN 3-900051-07-0
17. Ruß, G.: Data mining of agricultural yield data: A comparison of regression models. In: Perner, P. (ed.) Advances in Data Mining. Applications and Theoretical Aspects. LNCS, vol. 5633, pp. 24–37. Springer, Heidelberg (2009)
18. Ruß, G., Kruse, R., Schneider, M., Wagner, P.: Estimation of neural network parameters for wheat yield prediction. In: Bramer, M. (ed.) AI in Theory and Practice II, July 2008. Proceedings of IFIP-2008, vol. 276, pp. 109–118. Springer, Heidelberg (2008)
19. Ruß, G., Kruse, R., Schneider, M., Wagner, P.: Optimizing wheat yield prediction using different topologies of neural networks. In: Verdegay, J., Ojeda-Aciego, M., Magdalena, L. (eds.) Proceedings of IPMU 2008, June 2008, pp. 576–582. University of Málaga (2008)
20. Ruß, G., Kruse, R., Schneider, M., Wagner, P.: Visualization of agriculture data using self-organizing maps. In: Allen, T., Ellis, R., Petridis, M. (eds.) Applications and Innovations in Intelligent Systems, January 2009. Proceedings of AI-2008, vol. 16, pp. 47–60, BCS SGAI. Springer, Heidelberg (2009)
21. Ruß, G., Kruse, R., Wagner, P., Schneider, M.: Data mining with neural networks for wheat yield prediction. In: Perner, P. (ed.) ICDM 2008. LNCS (LNAI), vol. 5077, pp. 47–56. Springer, Heidelberg (2008)
22. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Technical report, Statistics and Computing (1998)
23. Stein, M.L.: Interpolation of Spatial Data: Some Theory for Kriging (Springer Series in Statistics). Springer, Heidelberg (June 1999)
24. Weigert, G.: Data Mining und Wissensentdeckung im Precision Farming - Entwicklung von ökonomisch optimierten Entscheidungsregeln zur kleinräumigen Stickstoff-Ausbringung. PhD thesis, TU München (2006)

Trend Mining in Social Networks: A Study Using a Large Cattle Movement Database

Puteri N.E. Nohuddin[1], Rob Christley[2], Frans Coenen[1], and Christian Setzkorn[2]

[1] Department of Computer Science, University of Liverpool, UK
[2] School of Veterinary Science, University of Liverpool and National Centre for Zoonosis Research, Leahurst, Neston, UK
{Puteri.Nohuddin,coenen,robc,c.setzkorn}@liverpool.ac.uk

Abstract. This paper reports on a mechanism to identify temporal spatial trends in social networks. The trends of interest are defined in terms of the occurrence frequency of time stamped patterns across social network data. The paper proposes a technique for identifying such trends founded on the Frequent Pattern Mining paradigm. The challenge of this technique is that, given appropriate conditions, many trends may be produced; and consequently the analysis of the end result is inhibited. To assist in the analysis, a Self Organising Map (SOM) based approach, to visualizing the outcomes, is proposed. The focus for the work is the social network represented by the UK's cattle movement data base. However, the proposed solution is equally applicable to other large social networks.

Keywords: Social Network Analysis, Trend Mining, Trend Visualization.

1 Introduction

A *Social Network* is an interconnected structure that describes communication (of some form) between individuals. Social Network Mining is concerned with the identification of patterns within such networks. Typical applications include: the identification of disease spreading patterns from dynamic human movement [2], monitoring users' topics and roles in email distributions [15], and filtering product ratings from online customer networks for viral marketing strategies [7]. Social network mining approaches tend to be founded on graph mining or network analysis techniques. Typically, social network mining is undertaken in a static context, a "snapshot" is taken of the network which is then analysed. Little work has been done on the dynamic aspects of social network mining. Further, most current work does not take into consideration the relationship between network nodes and their associated geographical location. The work described in this paper seeks to address these two issues.

In the context of this paper, social networks are viewed in terms of a sequence of snapshots taken of the network at discrete time intervals. The patterns of interest are identified using Frequent Pattern Mining (FPM) techniques, which relate groups of data items that frequently appear together. An adaptation of the Total From Partial (TFP) FPM algorithm [5] is used, Trend Mining TFP (TM-TFP), to discover trends in sequences of time stamped social networks. The trends are described in terms of sets of frequency measures associated with specific frequent patterns that occur across the

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 464–475, 2010.

network. These sequences are conceptualised as time series or“trend lines”. However, using the TM-TFP approach, given appropriate input conditions, a great many such trends may be identified. The trends of interest are very much application dependent. We may be interest in increasing or decreasing trends, or trends that represent seasonal changes. In addition, we may wish to identify “flat” trend lines, or sudden changes in trends. This paper therefore also suggests a visualisation technique, founded on a Self Organising Map (SOM) approach, to cluster similar trends. This allows end users to focus on the application dependent trend lines of interest.

The focus of the work described in this paper is the UK's cattle movement database. This is a UK government funded initiative, managed by The Department for Environment, Food and Rural Affairs (Defra), introduced in 1998 in response to a cattle disease epidemic. The database records the movement of all cattle between pairs of locations in Great Britain. As such, these locations can be viewed as network nodes, and the movement of cattle as weighted links between node pairs. All data entries are time stamped, and thus snapshots of the network can be obtained. Sequences of patterns, trends, can thus be identified within the network. Similarly, each location (node) is referenced geographically and thus a spatial element can be added into the analysis. The activity evidenced in the databases was grouped at monthly intervals to obtain a sequence of “snapshots”. For any given month, the number of network nodes was in the region of 56,300, and the number of links in the region of 73,600, in other words the size of each time stamped network was substantial.

The overall contribution of this paper may thus be summarized as follows. Firstly, a mechanism for identifying spatial-temporal trends in large social networks is described. Secondly, a technique is presented to support the analysis/visualisation of the outcomes by clustering similar trends. Thirdly, a “real-life” application of the techniques is presented and evaluated.

The rest of this paper is organized as follows. Section 2 provides a brief background of FPM, social network mining and the SOM technology used. Section 3 describes the cattle movement database (social network) application. An overview of the proposed spatio-temporal social network trend mining framework is then given in Section 4. A full evaluation of the framework is reported in Section 5; and, finally, Section 6 provides some conclusions.

2 Background

This section provides some necessary background to the work described in the rest of this paper. It commences with a brief review of FPM, then continues with a consideration of current work in trend mining, social network mining and SOM technique.

FPM was first popularized in the context of Association Rule Mining (ARM). The catalyst ARM algorithm is generally acknowledged to be the Apriori algorithm [1]. Many alternative ARM and FPM algorithms have since been proposed. The FPM algorithm used with respect to the work reported in this paper is the TFP (Total From Partial) algorithm [5]. TFP uses a tree structure, the P-tree, in which partial support counts are stored; and a second tree structure, the T-tree (a reverse set enumeration tree data structure) facilitates fast “look up” ([4]). TFP offers advantages, with respect to many other FPM algorithms, in terms of computational efficiency and storage.

Trend mining is concerned with the identification of patterns that change over time. Trends are typically defined in terms of time series. One example is Google Trends, a public web facility recently introduced by Google to identify trends associated with keyword search volume across various global regions and in various languages [22]. Trend recognition processes can be applied to qualitative and also to quantitative data, such as forecasting financial market trends based on numeric financial data, and usage of text corpi in business news [18]. In the context of this paper, trends are defined in terms of the changing frequency of individual patterns. A similar concept has been used in the context of Jumping Emerging Pattern (JEP) mining. For example in Khan et al. [9], a moving window was used to identify such patterns (patterns whose support changes significantly over time).

There has been a rapid increase in attention, within the data mining community, regarding social network analysis. This is because of the demand to exploit knowledge from the large amount of data that has been collected with respect to the social behavior of users in online environments. A social network depicts the structure of some social entities, and normally comprise actors who are connected through one of more class of links [20]. To analyze this structure, many social network analysis techniques have been proposed which map and measure the relationships and flows between people, organizations, groups, computers or web sites. Social network mining can be applied in a static context, which ignores the temporal aspects of the network; or in a dynamic context, which takes temporal aspects into consideration. In a static context, we typically wish either: (i) to find patterns that exist across the network, or (ii) cluster (group) subsets of the networks, or (iii) build classifiers to categorize nodes and links. In addition, given the dynamic context, we wish to identify trends or change points within social networks. A further point of interest is the geographical relationships represented by nodes. Given the availability of spatio-temporal data, we can determine the relationship between nodes by evaluating the spatio-temporal co-occurences of events in social networks [14]. Thus, we are able to detect changes or abnormal patterns in comunnication (movement) behaviour in the network.

Self Organising Maps (SOMs), or Self-Organizing Feature Map (SOFM), were first proposed by Kohonen [11,10]. Essentially, SOMs are a neural network based technique designed to reduce the number of data dimensions in some input space by projecting it onto a "map", which plots the similarities of the input data by grouping similar data items together (i.e. clustering the data). The algorithm is based on unsupervised and competitive learning, and typically operates by first initialising a $n \times m$ matrix of nodes where each node is to be associated with a cluster. Currently, there is no scientific method for determining the best value for n, i.e. to identify how many clusters should be represented by a SOM, however the $n \times m$ value does define a maximum number of clusters; in most cases, on completion of the SOM algorithm, some nodes will be empty [6]. Since SOM are based on competitive learning, the output nodes on the map compete among each other to be stimulated to represent the input data. Eventually, some nodes can be empty without any input vectors. The authors implemented a SOM using Matlab toolbox functions. The toolbox provided functions, based on the Kohonen SOM algorithm, that determine the distance between input data and represented similar input data on the map [19]. A SOM approach was adopted because it could group similar trends, and thus enhance the analysis of the TM-TFP result, without requiring prior input of the number of desired clusters ($n \times m$). It also represented a "tried and tested" approach that had been successfully used in many engineering applications such as patterns recognition and process monitoring [12].

3 The Cattle Movement Data Base

The work described in this paper focuses on the UK Cattle Tracing System (CTS) database. Nevertheless, the proposed technique is equally applicable to other types of data sets with similar properties. The CTS is maintained by the British Cattle Movement Service (BCMS). The CTS database is the core information source for Defra's RADAR (Rapid Analysis and Detection of Animal-related Risks) database with regard the birth, death and movement of cattle in Great Britain. The required recording is undertaken using a range of mechanisms including post, telephone and a dedicated website [21]. Cattle movements can be one off movements to final destinations, or movements between intermediate locations [16]. In short, the movement types include cattle imports, movements between locations, on movement in terms of births and off movements in terms of death. CTS was introduced in September 1998, and updated in 2001 to support the disease control activities. Currently, the CTS database holds some 155 Gb of data.

The CTS database comprises a number of tables, the most significant of which are the $animal$, $location$ and $movement$ tables. The animal table gives information about each individual animal, referenced by an ID number, such as the breed of the animal (185 different breads are recorded) and animal date of brith. The location table gives details about individual locations, again referenced by a unique ID number, such as its grid coordinates (Easting and Northing) and location type. The most common location types are Agricultural Holding, Landless Keeper, Market, Common Land and slaughterhouses. A total of thirteen different categories of location are recognised within the database. The movement table, in the context of the work described here, is the most significant. Each record in the movement table describes the movement of one animal from a *sender* location to a *receiver* location. The table also includes the date of the movement.

For the work described in this paper, information was extracted from these tables into a single data warehouse that comprised sets of time stamped data "episodes". The temporal granularity used through out was one month. The number of CTS records represented in each data episode was about 400,000, each record representing a cattle movement instance. Each record in the warehouse comprised: (i) a time stamp, (ii) the number of cattle moved, (iii) the breed, (iv) the senders location in terms of Eastings and Northings, (v) the "type" of the sender's location, (vi) the receivers location in terms of Eastings and Northings, and (vii) the "type" of the receiver's location. If two different breeds of cattle were moved at the same time from the same sender location to the same receiver location this would generate two records in the warehouse. The maximum number of cattle moved between a pair of locations for a single time stamp could be more than 40 animals.

In summary, the cattle movement warehouse can be interpreted as a social network where nodes described farms or holding areas and links described cattle movements between farms and/or holding areas. The number of animals moved, of a given breed type, between any pair of nodes, was regarded as a link weighting. The spatial magnitude of movement between nodes can be derived from the location grid values.

By applying the proposed spatio-temporal trend mining technique to the CTS social network, trends describing cattle movements, across time and space, can be identified. Within these trends we can also identify sub-trends, i.e. trends contained within trends. As noted above, a trend describes the fluctuations of the frequency of a frequent pattern over time. Frequent patterns often contain sub-patterns (sub-sets) which also have trend lines associated with them. Trend-lines belonging to such patterns are identified as sub-trends of the trends associated with the parent (super-set) patterns. Generally, sub-trends display similar trends to their parent trends, but not necessarily so.

Some previous studies of the CTS database have been conducted. Green and Kao [8], who conducted an analysis of the CTS database, confirmed that the number of movement record decreased as the distance between farm location increase. Another significant example is the study undertaken by Robinson and Christley [17], who identified a number of trends in the database. These trends demonstrated that the UK cattle population was in constant flux, and that both seasonal and long term patterns might be identified to support model predictions and surveillance strategies. However, Robinson and Christley's study was very "high level" and considered temporal trends across the entire CTS database without acknowledging spatial factors. The approach describes in this paper serves to identify trends that exhibit both spatial and temporal aspects.

4 The Social Network Trend Mining Framework

This section provides an overview of the proposed social network trend mining framework. The framework comprises two principal components: (i) the trend mining unit, and (ii) the visualization unit. A block diagram giving a high level view of the framework is presented in Figure 1.

The Trend Mining unit (represented by the elements in the top half of Figure 1) is responsible for the identification of trend lines, one per identified pattern, across the input social network. An established FPM algorithm, TFP [5] (introduced in Section 2) was extended for this purpose. The resulting software, Trend Mining TFP (TM-TFP), took as input a sequence of time stamped binary valued data tables, and a minimum support threshold (S); and produced a sequence of trend lines. It should be noted that if a particular pattern, at a particular time stamp, fell below the support threshold it was deemed to be "not relevant" and an occurance value of 0 was recorded in the trend line, however the pattern was not "thrown away". TM-TFP utilizes the T-tree and P-tree data structures used in TFP. These were essentially set enumeration tree structures designed to enhance the efficiency of the TFP algorithm. In addition, TM-TFP made use of a further tree data structure, the TM tree, that combined the identified frequent item sets into trend lines.

The visualization unit was developed using the SOM Toolbox in Matlab [19]. The toolbox made use of functions and learning processes based on the Kohonen SOM algorithm. The authors extended the software to produce prototype and trend line maps. As will be illustrated in the following section, the prototype map displays the characteristics of each identified trend line cluster, while the trend lines map gives information regarding the number of trends that exist in each cluster.

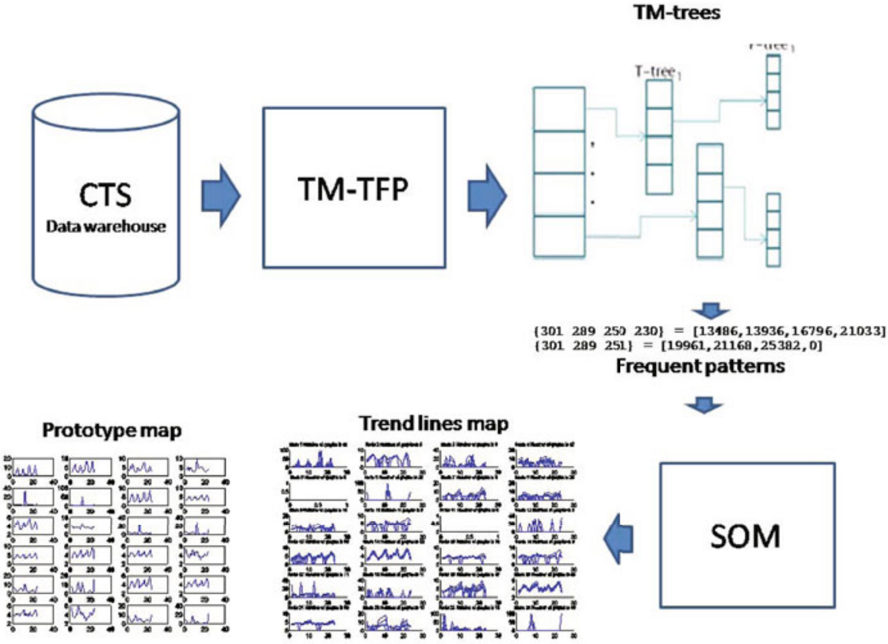


Fig. 1. Block diagram outlining social network trend mining framework

5 Evaluation

To evaluate the proposed framework, numerous experiments were conducted, the most significant of which are reported in this section. All the reported experiments were directed at the cattle movement database introduced in Section 3. This section is divided into three Sub-sections as follows. Sub-section 5.1 describes the necessary preprocessing conducted on the input data. Sub-section 5.2 reports on the evaluation of the TM-TFP algorithm. Sub-section 5.3 considers the SOM visulalisation/clustering mechanism adapted to support the anlysis of the trend lines identifyed using TM-TFP.

5.1 Data Preprocessing, Normalisation and Discretisation

Several subsets of the CTS database were used for the evaluation. These ranged from between six month's of data to two year's of data time stamped at monthly intervals. There were twenty four monthly data sets collected from within the time period of January 2005 to December 2006 inclusive.

The FPM algorithm used, TFP [5], in common with most other FPM algorithms, operates using binary valued data. The data had therefore to be normalised and discretized. The Easting and Northing coordinate values were divided into k kilometer sub-ranges, experiments using both $k = 50$ and $k = 100$ were conducted. The effect of this ranging was to sub-divide the geographic area represented by the CTS into a $k \times k$ grids, allowing for the inclusion of trends that express both spatial and temporal relationships.

The number of cattle moved value (m) was discretised into five sub ranges: $m \leq 10$, $11 \leq m \leq 20$, $21 \leq m \leq 30$, $31 \leq m \leq 40$ and $m > 40$. The non-linear distribution was used so that a roughly equal proportion of records was included in each sub-range. The end result of the normalisation/discretisation exercise was a table schema comprising 265 attributes where $k = 100$, and 305 attributes where $k = 50$.

5.2 Evaluation of TM-TFP

This Sub-section presents an analysis of the proposed TM-TFP algorithm. Experiments were conducted using a sequence of support thresholds: $S = 2\%$, $S = 5\%$ and $S = 8\%$. Four different *temporal windows* were used, 6 months, 12 months, 18 months and 24 months, corresponding to 6, 12, 18 and 24 data episodes respectively. In this analysis, the Easting and Northing coordinate of datasets were normalised according to 100 km grid squares (i.e. $k = 100$).

TM-TFP identified frequent patterns within each time stamped data episode, with their individual support values. These patterns were then related across the data to identify trend lines. The total number of trend lines identified, using support thresholds of 2%, 5% and 8% are presented in Table 1. For example, for the six month data input, for $S = 2\%$, 1993 trend lines were identified; while with $S = 5\%$, 523 trend lines were identified, and with $S = 8\%$ 222 trend lines. From Table 1, it can be seen that a great many trend lines are discovered. As the temporal window is increased there is a corresponding (although slight) increase in the number of identified trend lines. As would be expected, increasing the support value had the effect of decreasing the number of identified patterns (trends), but at the risk of missing potentially significant trends.

Table 1. Number of trend lines identified using TM-TFP algorithm when $k = 100$

Duration	Support Threshold		
(months)	2	5	8
6	1993	523	222
12	2136	563	242
18	2175	570	248
24	2204	580	257

Table 2. Run time values (seconds) using the TM-TFP algorithm

Duration	Support Threshold		
(months)	2	5	8
6	39.52	29.09	28.11
12	78.58	58.91	55.43
18	114.85	88.12	83.02
24	156.96	118.61	115.48

For completeness, Table 2 presents a sequence of "run time" values for the reported sequence of experiments so as to give an indication of the time complexity of the TM-TFP trend mining algorithm. Increases in the size of the temporal window (number of data episodes) and the support thresholds gave rise to corresponding linear increases in TM-TFP run time.

5.3 Evaluation of SOM Visualisation

From the above, it can be seen that a great many trends can be identified, the number of trends increases as the temporal window size is increased, and the support value is decreased. This presents a challenge concerning the analysis and interpretation of the identified trend lines. More generally, we can observe that powerful mining algorithms, such as TM-TFP, may overwhelm the end user with too many patterns. Further, the trend lines produced by algorithms such as TM-TFP can occur in many shapes.

The proposed solution is to use a SOM approach to cluster the trends, and consequently ease the interpretation. As noted in Section 2, a SOM consists of several nodes with associated prototypes. A prototype is a vector of the same dimensionality as the original data. The training algorithm adjusts the prototypes, enabling the SOM to spatialy order high-dimensional data in much lower dimensional space whilst maintaining complicated underlying relationships and distributions. Time series can be presented to the trained SOM whereby the node with the most similar prototype is declared the winner. Similarity may be determined in a number of ways but the most straight forward is the Euclidean distance measure. Due to the training process, we can expect similar time series to be won by spatially close nodes. Hence, if nodes are arranged as a two dimensional grids, we may expect nodes in the top left corner to win time series of very different shapes in comparison to those in the lower right corner. This is a very useful property of SOMs. Furthermore, large numbers of time series can be reduced to a few meaningful coordinates whilst maintaining their underlying relationships. Domain experts can combine neighbouring nodes to form clusters, reducing the number of patterns even further. This might lead to a handful of clusters containing (say) steady, increasing, decreasing and constantly changing time series.

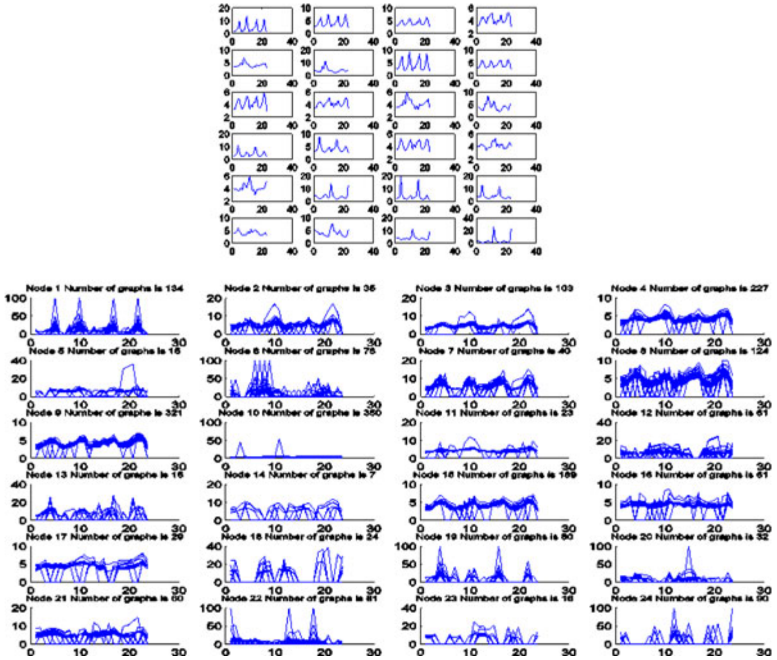


Fig. 2. SOM frequent patterns($S = 2\%$)

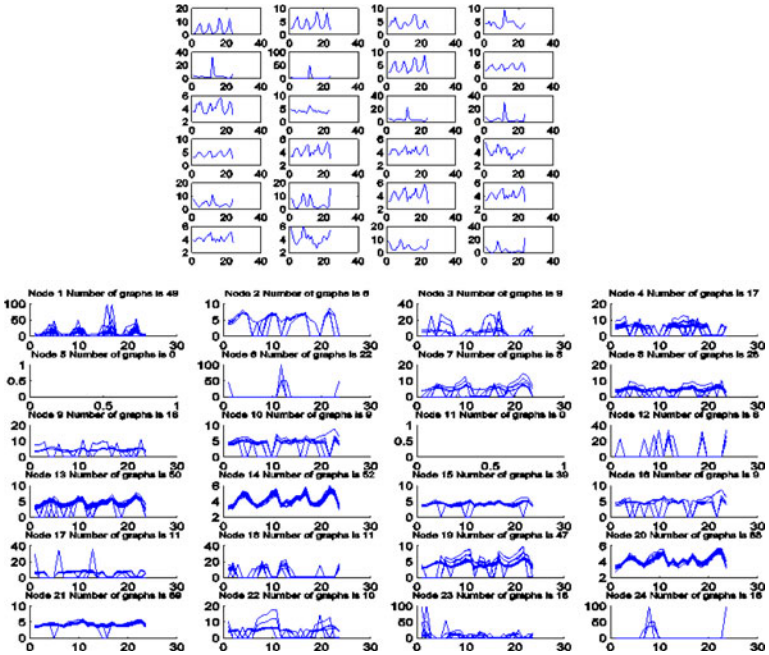


Fig. 3. SOM frequent patterns ($S = 5\%$)

For the analysis described in this Sub-section, the SOM was seeded with 6×4 nodes, each node representing a cluster. The authors have tested several sizes of SOM to generate clusters of trend lines. However, there is no specific method to determine the most appropriate size of a SOM map, and a 6×4 was found to be the most effective. The bigger the $n \times m$ size of a SOM map the greater the possible number of clusters that may be identified. For example, in medical images clustering, it has been shown that a higher size SOM gives a better image clustering results [3]. For discussion purposes, in this Sub-section, nodes are sequentially numbered by row, reading from top-left to bottom-right. Recall that each trend line represents a sequence of support values. Thus, the authors applied a distance function[1] and a neighbourhood function[2] to determine similarity. The visualization results are presented in Figures 2, 3 and 4 for a sequence of support thresholds ($S = 2\%$, $S = 5\%$ and $S = 8\%$) using the twenty four month data set, the largest of the four data sets experimented with. Each figure comprises a pair of "maps", the top map presents the prototypes for the identified clusters and the below map the trends contained in each cluster. As might be expected more clusters were found with $S = 2\%$ than with $S = 5\%$ and $S = 8\%$. Further analysis verified that subsets of the frequent patterns tended to be included in the same cluster as their supersets. Thus, in each cluster, only certain attribute labels/columns were combined;

[1] Euclidean function is the distance function used to calculate shortest distance.

[2] Gaussian function is used to determine the neighbourhood size on the map.

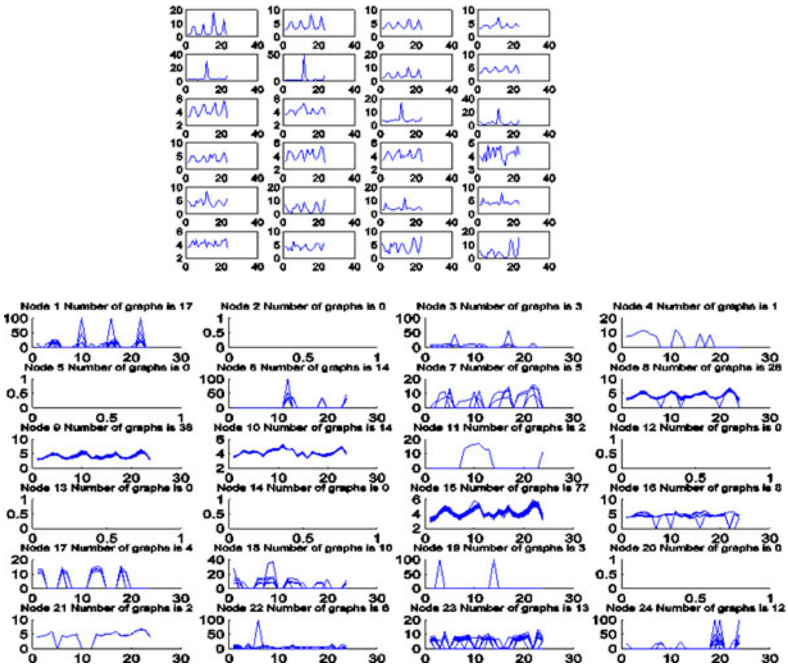


Fig. 4. SOM frequent patterns ($S = 8\%$)

therefore providing for more effective, focused and understandable result interpretation by the end user.

Some specific trend examples (taken from Figures 2, 3 and 4) are: (i) the pattern (Sender holding locations in area B and movement 25-30 animals of Breed British Friesian and receiver holding locations in area D) is in constant flux across the 24 months (area B and D are identifiers for specific grid squares in the geographic area under consideration), and (ii) the pattern (Movement 20-25 animals and Breed Highland and gender female) is static across the time period. The largest number of trends contained in a single cluster was 360 (node 10 in Figure 2). Some nodes, for example numbers 2, 3, 4, 7, 8, 9 and 15 in Figure 2, show "clear-cut" types of trend lines in the respective clusters. Regardless of the "outlier" trend lines shown in each node, the prototype SOM portrays the prototype trends in each node.

The maps presented in Figures 3 and 4 (for $S = 5\%$ and $S = 8\%$) show similar clusters. Further analysis established that this was because use of a lower support threshold resulted mostly in the identification of subsets of those identified using $S = 8\%$ (although for $S = 2\%$ additional patterns were also discovered that were not found using $S = 8\%$).

In all cases, the maps identified similar prototypes indicating that if we are only interested in identifying prototypes a higher support threshold, which offers the advantage of greater computational efficiency, is sufficient. By applying the SOM technique, the authors were able to provide a lower dimension of prototypes to represent a substantial number of generated trend lines. In each node, trend lines exhibit the frequent itemsets which include all the possible combination of attributes that carry spatial and temporal features. Thus, the changes or fluctuation can be perceived easily for further analysis.

6 Conclusions

In this paper, the authors have described a framework for identifying and visualising spatio-temporal trends in social networks. The trends were defined in terms of trend lines (in effect time-series) representing the frequency of occurrence of individual patterns with time. Firstly, the trends were identified (the first element of the two part framework) using an extension of the TFP algorithm, TM-TFP. The challenge of TM-TFP was that, given a realistically sized database, a great many trends could be identified making them difficult to analyse. Secondly, to addressed the analysis issue, the framework used a SOM based clustering mechanism which allowed for similar trends to be grouped together. The advantage offered was that this would allow decision makers, and other end users, to focus on relevant trends.

The framework was evaluated using the UK's Cattle Tracking System (CTS) data base. More specifically, it was applied to sequences of snapshots of the database covering from six months to two years of data. The main findings may be summarized as follows:

1. TM-TFP can successfully identify trends in large social networks with reasonable computational efficiency.
2. The SOM clustering/visualisation technique provides a useful mechanism for grouping similar trends.
3. From the SOM, to ease decision makers to spot trend lines in each cluster for further investigation to be taken.

The research team have been greatly encouraged by the results produced to date and are currently investigating further mechanisms where by understanding and advanced analysis of the identified trends can be facilitated.

References

1. Aggrawal, C., Yu, P.: A Condensation Approach to Privacy Preserving Data Mining. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) EDBT 2004. LNCS, vol. 2992, pp. 183–199. Springer, Heidelberg (2004)
2. Read, J.M., Eames, K.T.D., Edmunds, W.J.: Dynamic Social Networks and the implications for the spead of infectious disease. J. R. Soc. Interface 5, 1001–1007 (2008)
3. Chalabi, Z., Berrached, N., Kharchouche, N., Ghellemallah, Y., Mansour, M., Mouhadjer, H.: Classification of the Medical Images by the Kohonen Network SOM and LVQ. Journal of Applied Sciences 8(7), 1149–1158 (2008)
4. Coenen, F.P., Leng, P., Ahmed, S.: Data Structures for association Rule Mining: T-trees and P-trees. IEEE Transactions on Data and Knowledge Engineering 16(6), 774–778 (2004)

5. Coenen, F.P., Goulbourne, G., Leng, P.: Computing Association Rules Using Partial Totals. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 54–66. Springer, Heidelberg (2001)
6. Cottrell, M., Rousset, P.: A powerful Tool for Analyzing and Representing Multidimensional Quantitative and Qualitative Data. In: Cabestany, J., Mira, J., Moreno-Díaz, R. (eds.) IWANN 1997. LNCS, vol. 1240, pp. 861–871. Springer, Heidelberg (1997)
7. Domingos, P.: Mining Social Networks for Viral Marketing. IEEE Intelligent Systems 20(1), 80–82 (2005)
8. Green, D.M., Kao, R.R.: Data quality of the Cattle Tracing System in Great Britain. Veterinary Record 161(13), 439–443 (2007)
9. Khan, M.S., Coenen, F., Reid, D., Tawfik, H., Patel, R., Lawson, A.: A Sliding Windows based Dual Support Framework for Discovering Emerging Trends from Temporal Data. KBS Journal (to be appeared 2010)
10. Kohonen, T.: The Self Organizing Maps. Neurocomputing 21, 1–6 (1998)
11. Kohonen, T.: The Self Organizing Maps. Series in Information Sciences, vol. 30. Springer, Heidelberg (1995)
12. Kohonen, T., Oja, E., Simula, O., Visa, A., Kangas, J.: Engineering applications of the Self-Organizing Map. Proceedings of the IEEE 84(10), 1358–1384 (1996)
13. Krause, G., Blackmore, C., Wiersma, S., Lesneski, C., Woods, C.W., Rosenstein, N.E., Hopkins, R.S.: Marijuana use and Social Networks in a Community Outbreak of Meningococcal Disease. South Medical Journal 94(5), 482–485 (2001)
14. Lauw, H., Lim, E., Pang, H., Tan, T.: Social Network Discovery by Mining Spatio-Temporal Events. Computational & Mathematical Organization Theory 11(2), 97–118 (2005)
15. McCallum, A., Corrada-Emmanuel, A., Wang, X.: Topic and Role Discovery in Social Networks with Experiments on Enron and Academic Email. Journal of Artificial Intelligence Research 30, 249–272 (2007)
16. Mitchell, A., Bourn, D., Mawdsley, J., Wint, W., Clifton-Hadley, R., Gilbert, M.: Characteristics of cattle movements in Britain: An analysis of records from the Cattle Tracing System. Animal Science 80, 265–273 (2005)
17. Robinson, S., Christley, R.M.: Identifying temporal variation in reported births, deaths and movements of cattle in Britain. BMC Veterinary Research, 2–11 (2006)
18. Streibel, O.: Trend Mining with Semantic-Based Learning. In: Proceedings of CAiSE-DC (2008)
19. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: Self-Organizing Map in Matlab: the SOM Toolbox. In: Proceedings of the Matlab DSP Conference (200)
20. Wasserman, S., Faust, K.: Social Network Analysis: Methods and Applications. Cambridge University Press, Cambridge (2006)
21. Defra. Livestock movements, identification and tracing: Cattle Tracing System, http://www.defra.gov.uk/foodfarm/farmanimal/movements/cattle/cts.htm
22. Google Trends, http://www.google.com/intl/en/trends/about.html

Spam Email Filtering Using Network-Level Properties

Paulo Cortez[1], André Correia[1], Pedro Sousa[3], Miguel Rocha[3], and Miguel Rio[2]

[1] Dep. of Information Systems/Algoritmi, University of Minho,
4800-058 Guimarães, Portugal
pcortez@dsi.uminho.pt, andrecrr@live.com.pt
http://www3.dsi.uminho.pt/pcortez

[2] Dep. of Informatics, University of Minho, 4710-059 Braga, Portugal
{pns,mrocha}@di.uminho.pt

[3] Department of Electronic and Electrical Engineering, University College London,
Torrington Place, WC1E 7JE, London, UK
m.rio@ee.ucl.ac.uk

Abstract. Spam is serious problem that affects email users (e.g. phishing attacks, viruses and time spent reading unwanted messages). We propose a novel spam email filtering approach based on network-level attributes (e.g. the IP sender geographic coordinates) that are more persistent in time when compared to message content. This approach was tested using two classifiers, Naive Bayes (NB) and Support Vector Machines (SVM), and compared against bag-of-words models and eight blacklists. Several experiments were held with recent collected legitimate (ham) and non legitimate (spam) messages, in order to simulate distinct user profiles from two countries (USA and Portugal). Overall, the network-level based SVM model achieved the best discriminatory performance. Moreover, preliminary results suggests that such method is more robust to phishing attacks.

Keywords: Anti-Spam filtering, Text Mining, Naive Bayes, Support Vector Machines.

1 Introduction

Email is a commonly used service for communication and information sharing. However, unsolicited e-mail (spam) emerged very quickly after email itself and currently accounts for 89% to 92% of all email messages sent [13]. The cost of sending these emails is very close to zero, since criminal organizations have access to millions of infected computers (known as botnets) [17]. Spam consumes resources, such as time spent reading unwanted messages, bandwidth, CPU and disk [7]. Also, spam is an intrusion of privacy and used to spread malicious content (e.g. phishing attacks, online fraud or viruses).

The majority of the current anti-spam solutions are based on [3]: Content-Based Filtering (CBF) and Collaborative Filtering (CF). CBF is the most popular anti-spam approach, using message features (e.g. word frequencies) and

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 476–489, 2010.

Data Mining (DM) algorithms (e.g. Naive Bayes) to discriminate between legitimate (ham) and spam messages. CF works by sharing information about spam messages. One common CF variant is the DNS-based Blackhole List (DNSBL), also known as blacklist, which contains known IP addresses used by spammers. CF and CBF can also be combined. For example, a blacklist is often used at a server level to tag a large number of spam. The remaining spam can be detected later by using a personalized CBF at the client level (e.g. Thunderbird SpamBayes, http://www.entrian.com/sbwiki).

Spam content is very easy to forge in order to confuse CBF filters. For example, normal words can be mixed into spam messages and this heavily reduces the CBF performance [15]. In contrast, spammers have far less flexibility in changing network-level features. Yet, the majority of the spam research gives attention to content and the number of studies that address network-level properties is scarce. In 2005, Leiba et al. [11] proposed a reputation learning algorithm that is based on the network path (from sender to receiver) of the message. Such algorithm obtained a high accuracy when combined with a CBF bayesian filter. Ramachandran and Feamster [17] have shown that there are spam/ham differences for several network-level characteristics (e.g. IP address space), although the authors did not test these characteristics to filter spam using DM algorithms. More recently, transport-level properties (e.g. TCP packet stream) were used to classify spam messages, attaining a classification accuracy higher than 90% [1].

In this paper, we explore network-level characteristics to discriminate spam (see Section 2.1). We use some of the features suggested in [17] (e.g. operating system of sender) and we also propose new properties, such as the IP geographic coordinates of the sender, which have the advantage of aggregating several IPs. Moreover, in contrast with previous studies (e.g. [11,1]), we collected emails from two countries (U.S. and Portugal) and tested two classifiers: Naive Bayes and Support Vector Machines (Section 2.2). Furthermore, our approach is compared with eight DNSBLs and CBF models (i.e. bag-of-words) and we show that our strategy is more robust to phishing attacks (Section 3).

2 Materials and Methods

2.1 Spam Telescope Data

To collect the data, we designed and developed the spam telescope repository. The aim of this repository is to perform a longitudinal and controlled study by gathering a significant slice of the world spam traffic. Spam was harvested by setting several spam traps; i.e. fake emails that were advertised through the Internet (e.g. Web pages). To collect ham, we created email addresses what were inscribed in moderated mailing lists with distinct topics. Figure 1 shows the proportions of mailing list topics that were used in our datasets. For both spam and ham collection, we tried to mimic real users from two countries: U.S. and Portugal (PT). For instance, we first registered a U.S. domain (.com) and then set the corresponding Domain Name System (DNS) Mail Exchange (MX) record. Next, the USA spam traps were advertised in USA popular Web sites and the

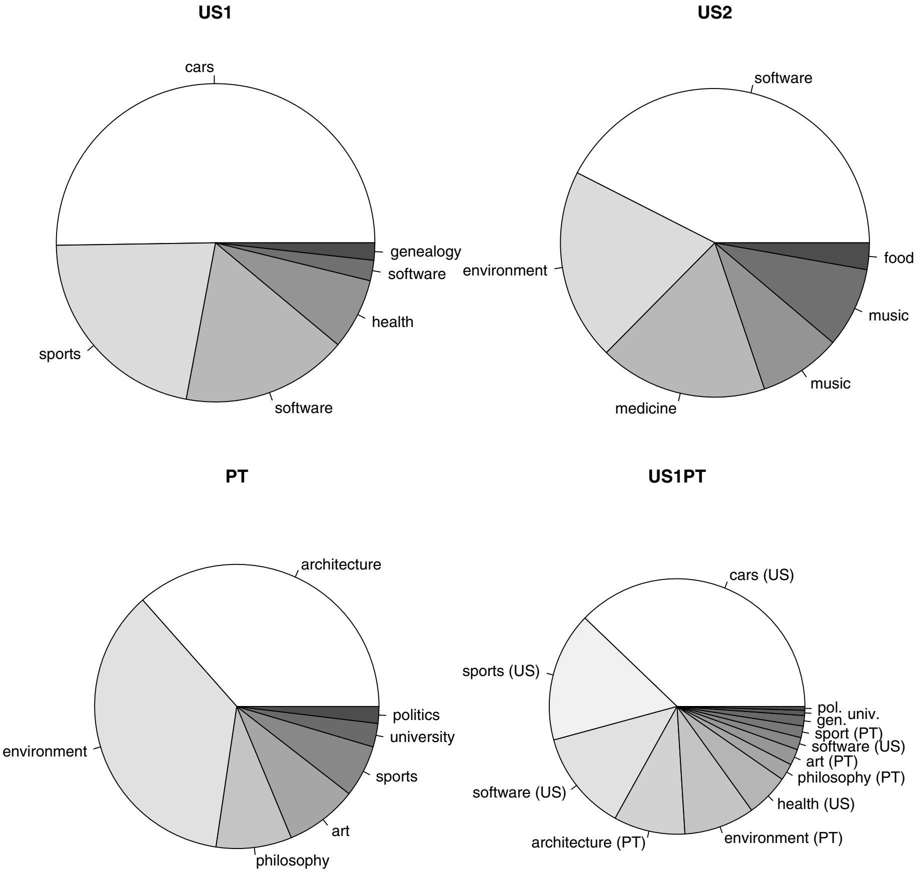


Fig. 1. Pie charts showing the distribution of mailing list topics for each dataset

USA ham emails were inscribed in 12 USA mailing lists. A similar procedure was taken to harvest the Portuguese messages (e.g. .pt domain).

All spam telescope messages were gathered at a specially crafted server. This server uses virtual hosting to redirect addresses from distinct Internet domains and runs a customized Simple Mail Transfer Protocol (SMTP) program called Mail Avenger (http://www.mailavenger.org). We set Mail Avenger to tag each received message with the following information:

- IP address of the sender and a traceroute to this IP;
- Operating System (OS) of the sender, as estimated from a passive p0f TCP fingerprint;
- lookup at eight DNSBLs: cbl.abuseat.org (B1), dnsbl.sorbs.net (B2), bl.spamcop.net (B3), sbl-xbl.spamhaus.org (B4), dul.dnsbl.sorbs.net (B5), zen.spamhaus.org (B6), psbl.surriel.com (B7) and blackholes.five-ten-sg.com (B8).

Table 1. Network-level attributes

Attribute	**Domain**
NHOP – number of hops/routers to sender	{8,9,. . .,65}
Lat. – latitude of the IP of sender	[-42.92°,68.97°]
Long. – longitude of the IP of sender	[-168.10°,178.40°]
OS – operating system of sender	{windows,linux,other,unknown}

Table 2. Summary of the Spam Telescope corpora

setup	**ham main**	**#mailing**	**#ham**	**total**	**spam**	**time**
	language	**lists**	**senders**	**size**	**/ham**	**period**
US1	English	6	343	3184	1.0	[23/Apr./09,9/Nov./09]
US2	English	6	506	3364	1.0	[21/Apr./09,9/Nov./09]
PT	Portuguese	7	230	1046	1.0	[21/May/09,9/Nov./09]
US1PT	Eng./Port.	13	573	4230	1.0	[23/Apr./09,9/Nov./09]
USWBS	English	6	257	612	0.2	[21/Apr./09,9/Nov./09]

The four network-level properties used in this study are presented in Table 1. Instead of using direct IP addresses, we opt for geographic coordinates (i.e. latitude and longitude), as collected by querying the free http://ipinfodb.com database. The geographic features have the advantage of aggregating several IPs. Also, it it known that a large fraction of spam comes from specific regions (e.g. Asia) [17]. The NHOP is a distance measure that was computed using the traceroute command. The passive OS signatures were encoded into four classes: windows – if from the MS family (e.g. windows 2000); linux (if a linux kernel is used); other (e.g. Mac, freebsd, openbsd, solaris); and unknown (if not detected).

In this study, we collected recent data, from April 21st April to November 9th 2009. Five datasets were created in order to mimic distinct and realistic user profiles (Table 2). The US1 set uses ham from 6 mailing lists whose members are mostly U.S. based, while US2 contains ham from different U.S. lists and that is more spread through the five continents (Figure 2). Regarding the spam, the data collected from the U.S. traps was added into US1 and US2, while PT includes only features extracted from Portuguese traps. The mixture of ham and spam was based on the time that each message was received (date field), which we believe is more realistic than the sampling procedure adopted in [14]. Given the large number of experiments addressed in this work, for US1, US2 and PT we opted to fix the global spam/ham ratio at 1. Yet, it should be noted that the spam/ham ratios fluctuate through time (right of Figure 5). The fourth set (US1PT) merges the data from US1 and PT, with the intention of representing a bilingual user (e.g. Portuguese but working in U.S.). Finally, the U.S. Without Blacklist Spam (USWBS) contains ham and spam from US2. The aim is to mimic a hybrid blacklist-filter scenario, thus all spam that was detected by any of the eight DNSBLs was removed from US2. For this last set, we set

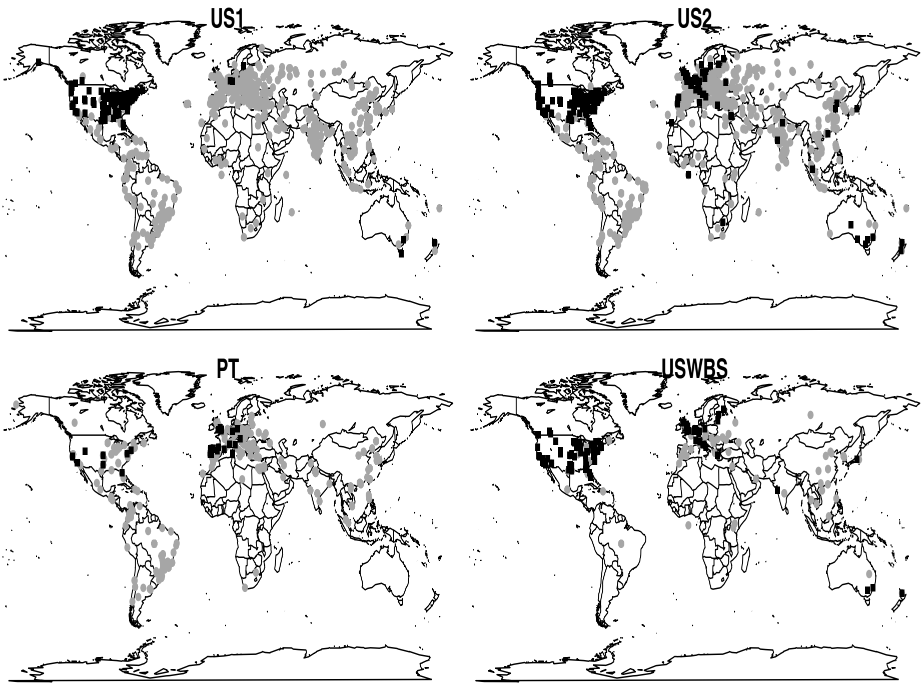


Fig. 2. Distribution of geographic IP of sender (black squares denote ham, gray circles show spam) for the used datasets

the spam/ham ratio to a realistic 0.2 value, since in such scenario most spam should be previously detected by the DNSBLs. Figures 2, 3 and 5 show several examples of ham/spam differences when analyzing the network-level attributes.

2.2 Spam Filtering Methods

We adopted two DM classifiers, Naive Bayes (NB) and Support Vector Machine (SVM), using the R statistical tool [16] (e1071 and kernlab packages) [16]. The NB algorithm is widely adopted by anti-spam filtering tools [3]. It computes the probability that an email message $j \in \{1, \ldots, N\}$ is spam (class s) for a filter trained over $\mathcal{D}$ data with N examples:

$$p(s|\mathbf{x}_j) = \beta \cdot p(s) \prod_{i=1}^{m} p(x_i|s) \tag{1}$$

where β is normalization constant that ensures that $p(s|\mathbf{x}) + p(\neg s|\mathbf{x}) = 1$, $p(s)$ is the spam frequency of dataset $\mathcal{D}$ and x_i denotes the input feature $i \in \{1, \ldots, m\}$. The $p(x_i|s)$ estimation depends on the NB version. We used the multi-variate

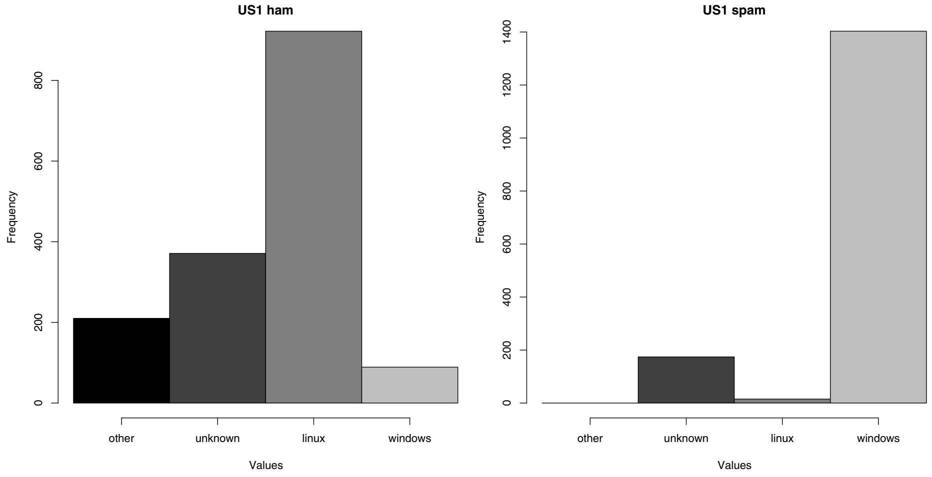


Fig. 3. Operating system histograms for the US1 dataset (left ham, right spam)

Gauss NB that is implemented in the R tool [14]:

$$p(x_i|c) = \frac{1}{\sigma_{i,c}\sqrt{2\pi}} \exp - \frac{(x_{ij} - \mu_{i,c})^2}{2\sigma_{i,c}^2} \quad (2)$$

where it is assumed each attribute (x_i) follows a normal distribution for each $c = s$ or $c = \neg s$ categories and the mean ($\mu_{i,c}$) and typical deviation ($\sigma_{i,c}$) are estimated from $\mathcal{D}$.

The Support Vector Machine (SVM) is a more powerful and flexible learner, capable of complex nonlinear mappings [5]. SVMs are particularly suited for classification tasks and thus they have been naturally applied to spam detection [8]. The basic idea is transform the input $\mathbf{x}_j \in \Re^m$ into a high f-dimensional feature space by using a nonlinear mapping. Then, the SVM finds the best linear separating hyperplane, related to a set of support vector points, in the feature space. The transformation depends on a nonlinear mapping that does not need to be explicitly known but that depends of a kernel function. We opted for the popular gaussian kernel, which presents less parameters and numerical difficulties than other kernels (e.g. polynomial):

$$K(\mathbf{x}_j, \mathbf{x}'_j) = exp(-\gamma||\mathbf{x}_j - \mathbf{x}'_j||^2),\ \gamma > 0 \quad (3)$$

The probabilistic output SVM computes [12]:

$$\begin{array}{c} f(\mathbf{x}_j) = \sum_{p \in SV} y_p \alpha_p K(\mathbf{x}_p, \mathbf{x}_j) + b \\ p(s|\mathbf{x}_j) = 1/(1 + exp(Af(\mathbf{x}_j) + B)) \end{array} \quad (4)$$

where SV is the set of support vectors, $y_j \in \{-1, 1\}$ is the output for message j (if spam y_j=1, else $y_j = -1$), b and α_p are coefficients of the model, and A

and B are determined by solving a regularized maximum likelihood problem. Under this setup, the SVM performance is affected by two parameters: γ, the parameter of the kernel, and C, a penalty parameter. Since the search space for these parameters is high, we heuristically set the least relevant parameter to $C = 3$ [4]. For NSV and to avoid overfitting, γ is set using a grid search (i.e. $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$). During this search, the training data was further split into training (first 2/3 of $\mathcal{D}$) and validation sets (last 1/3). Then, the best γ (i.e. with the highest AUC in the validation set) was selected and the model was retrained with all $\mathcal{D}$ data. Since the WSV model requires much more computation (with up to 3000 features when compared with the 4 NSV inputs), for this model we set $\gamma = 2^{-3}$.

DM models such as NB and SVM are harder to interpret when compared with simpler methods (e.g multiple regression). Still, it is possible to extract knowledge in terms of input relevance by using a sensitivity analysis procedure [6]. This procedure is applied after the training phase and analyzes the model responses when the inputs are changed. Let $p(s|\mathbf{x}(l))$ denote the output obtained by holding all input variables at their average values except x_a, which varies through its entire range with $l \in \{1, \ldots, L\}$ levels. If a given input variable ($x_a \in \{x_1, \ldots, x_m\}$) is relevant then it should produce a high variance (V_a). Thus, its relative importance (R_a) can be given by:

$$\begin{array}{ll} V_a & = \sum_{l=1}^{L} (p(s|\mathbf{x}(l)) - \overline{p(s|\mathbf{x}(l))})^2/(L-1) \\ R_a & = V_a / \sum_{i=1}^{m} V_i \times 100\,(\%) \end{array} \tag{5}$$

In this work, we propose novel filters based on network-level inputs and compare these with bag-of-words models and blacklists. For the first two classes of filters, we tested both NB and SVM algorithms using either network based attributes or word frequencies. The complete set of models includes:

- NNB and NSV, NB and SVM classifiers using the four inputs from Table 1;
- WNB and WSV, NB and SVM using word frequencies;
- Eight blacklist based models (B1,...,B8), where spam probabilities are set to $p(s|\mathbf{x}_j) = 1$ if the IP is present in the corresponding DNSBL, else it is 0;
- finally, the All Blacklist (AB) method that outputs $p(s|\mathbf{x}_j) = 1$ if any of the eight DNSBLs was activated, otherwise it returns 0.

Regarding the bag-or-words models (WNB and WSV), we used the preprocessing adopted in [7]. First, all attachments are removed. In the case of ham, all mailing list signatures are also deleted. Then, word frequencies are extracted from the subject and body message (with the HTML tags previously removed). Next, we apply a feature selection that is based in ignoring any words whose frequency is lower than 5 in the training set ($\mathcal{D}$) and then selecting up to the 3000 most relevant words according to a mutual information criterion. Finally, we apply a TF-IDF and length normalization transform to the word frequencies. All preprocessing was performed using the perl [2] and R languages [16].

2.3 Evaluation

To access the predictive performances, we adopted the realistic incremental retraining evaluation procedure, where a mailbox is split into batches $b_1, \ldots, b_n$ of k adjacent messages ($|b_n|$ may be less than k) [14]. For $i \in \{1, \ldots, n-1\}$, the filter is trained with $\mathcal{D} = b_1 \cup \ldots \cup b_i$ and tested with the messages from b_{i+1}.

For a given probabilistic filter, the predicted class is given by: s if $p(s|\mathbf{x}_j) > D$, where $D \in [0.0, 1.0]$ is a decision threshold. For a given D and test set, it is possible to compute the true (TPR) and false (FPR) positive rates:

$$\begin{aligned} TPR &= TP/(TP+FN) \\ FPR &= FP/(FP+TN) \end{aligned} \tag{6}$$

where TP, FP, TN and FN denote the number of true positives, false positives, true negatives and false negatives. The receiver operating characteristic (ROC) curve shows the performance of a two class classifier across the range of possible threshold (D) values, plotting FPR (x-axis) versus TPR (y-axis) [9]. The global accuracy is given by the area under the curve ($AUC = \int_0^1 ROCdD$). A random classifier will have an AUC of 0.5, while the ideal value should be close to 1.0. With the incremental retraining procedure, one ROC is computed for each b_{i+1} batch and the overall results are presented by adopting the vertical averaging ROC (i.e. according to the FPR axis) algorithm presented in [9]. Statistical confidence is given by the t-student test [10].

3 Experiments and Results

We tested all methods from Section 2.2 in all datasets from Table 2 and using a retraining evaluation with a batch size of $k = 100$ (a reasonable value also adopted in [14]). The obtained results are summarized as the mean of all test sets (b_{i+1}, $i \in \{1, \ldots, n-1\}$), with the respective 95% confidence intervals and shown in Tables 3 and 4. To increase clarity, we only show the best blacklist (B6) in Table 3. In the tables, the best values are in **bold**, while underline denotes a statistical significance (i.e. p-value<0.05). In Table 3, the significance was computed for a paired t-test comparison (with Bonferroni correction) of the network-level approach against AB and the corresponding bag-of-words method (e.g. NSV vs AB and WSV). In Table 4, the paired t-test is performed against the second best blacklist (B4).

Under the AUC metric and for all setups, the NSV method is the best choice and the obtained results are of high quality (from 95.3% to 99.8%). The NNB is the second best filter for the last three datasets. It is also interesting to notice that both NSV and NNB are robust to a geographic spread of the ham origin, since there is only a slight decrease (0.4 and 0.8 pp) when comparing US2 and US1 filtering performances. For WSV, the detection capability is higher when there is Portuguese ham (PT and US1PT). This was an expected behavior, since most spam is written in English. The bag-of-words performances decrease

Table 3. Comparison among the main filters (AUC test set results, in %)

setup	B6	AB	WNB	WSV	NNB	NSV
US1	98.0±0.8	98.9±0.5	73.0±4.7	75.8±2.2	98.7±0.6	**99.8**±0.2
US2	98.1±0.7	98.9±0.6	65.4±2.7	77.0±2.3	97.9±0.8	**99.4**±0.5
PT	83.9±4.5	89.0±3.4	71.4±7.5	82.1±5.1	95.6±1.5	**97.3**±1.6
US1PT	94.5±0.9	96.3±0.8	68.4±3.4	78.2±2.0	98.2±0.5	**99.2**±0.4
USWBS	50.0±0.0	50.0±0.0	50.1±0.3	63.6±7.6	94.7±3.4	**95.3**±3.6

Table 4. Blacklist filter performances (AUC test set results, in %)

setup	B1	B2	B3	B4	B5	B6	B7	B8
US1	87.6±1.2	80.2±1.8	80.8±1.6	87.7±1.2	58.7±1.1	**98.0**±0.8	74.3±2.6	67.1±2.6
US2	88.7±1.2	80.3±2.0	79.4±2.0	88.9±1.2	59.2±1.0	**98.1**±0.7	74.0±2.6	67.5±3.0
PT	76.0±3.8	74.7±2.1	69.1±3.7	78.0±4.2	59.0±3.0	**83.9**±4.5	65.5±3.0	63.0±4.5
US1PT	84.5±1.0	79.0±1.5	77.2±1.1	85.2±0.9	58.7±1.1	**94.5**±0.9	72.2±1.8	66.2±2.0

substantially for the last setup, showing that the spam that is not detected in blacklists is more difficult to classify based on content. However, our network-level based methods still obtained high AUC values, around 95%. When using the same inputs, the SVM algorithm is always better when compared with NB, with an average improvement of 1.2 pp for the network-level features and 9.7 pp for the bag-of-words attributes.

Regarding the blacklist comparison (Table 4), B6 is clearly the best filter. Overall, the second best DNSBL is B4, followed by B1. For all setups, three blacklists (B5, B7 and B8) are outperformed by the WSV model. B5 is the worst filter, with no average AUC value above 60%. For all DNSBLs except B5, the worst performance is achieved for the Portuguese dataset (PT). This outcome was expected, since the tested blacklists are international and thus may fail in mapping more country specific spam.

The full ROC analysis is given in Figure 4. To increase clarity, we only selected the best and worst blacklists (B6 and B5). The ROC curve allows the definition of different filtering profiles, according to the user needs. In the studied datasets, the blacklists never output a false positive. Thus, for B6 and AB, the TPR values are high when FPR is zero. For the spam domain, this is an important point of the ROC curve, since often the cost of losing normal e-mail (FP) is much higher than receiving spam (FN). This is particularly true if the email client action is set to delete messages marked as spam. For this decision point, AB, followed by B6, are the best filters, except for US1 and USWBS, where NSV is the best option. For larger admissible values of FPR, NSV gives the best TPR values. It should be noted that for some users, this is an interesting scenario, as the cost of receiving spam can also be high, due to an higher vulnerability to phishing attacks, viruses or online fraud, while not all ham is important. Since often email clients move messages marked as spam to a different folder, false positives could still be read by the user.

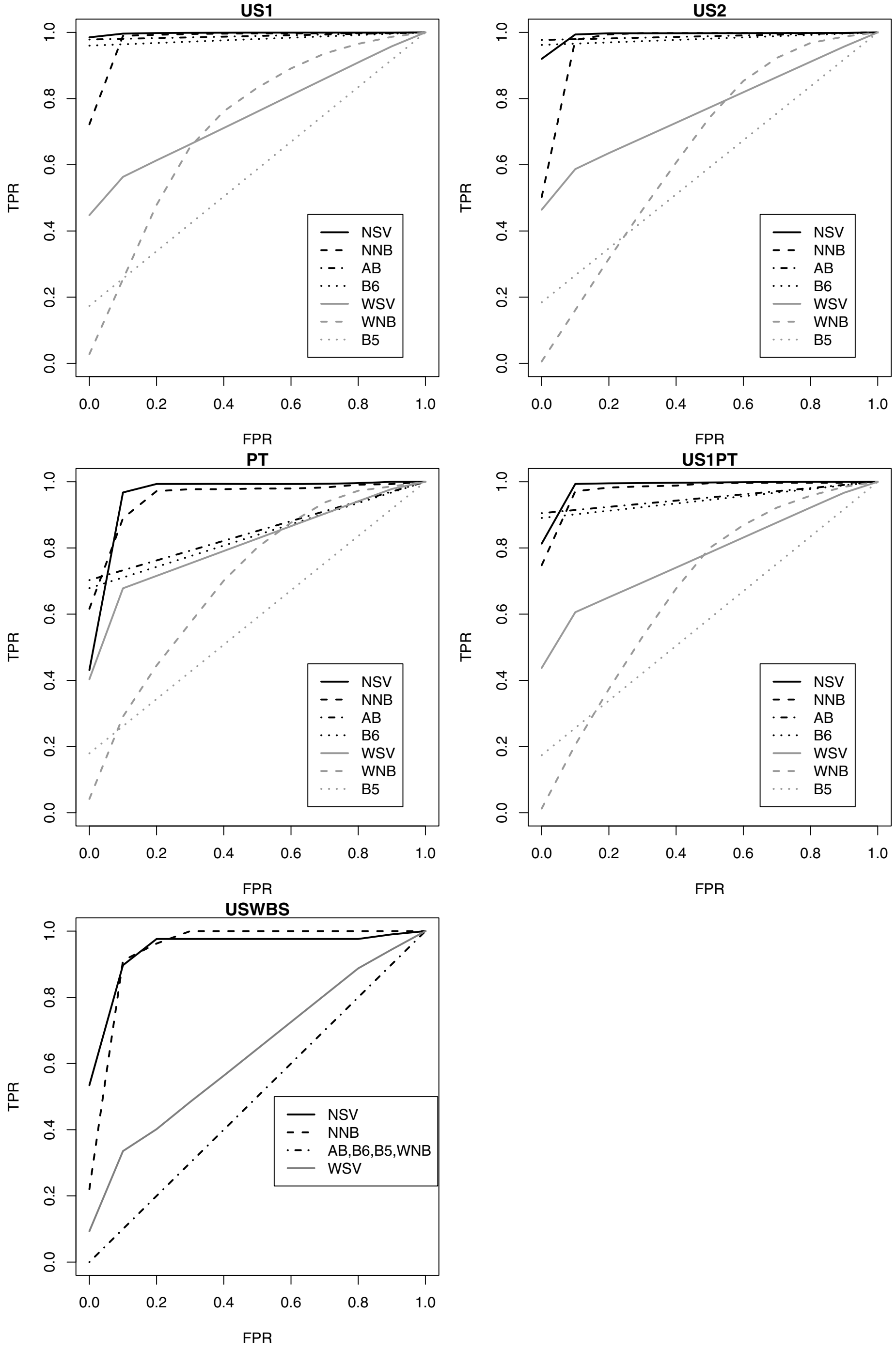


Fig. 4. Average test set ROC curves

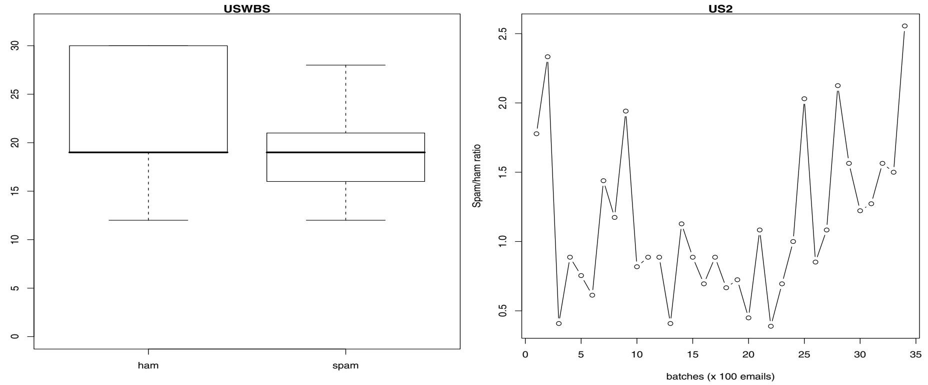


Fig. 5. NHOP ham/spam box plots for the USWBS dataset (minimum, median and maximum values, left) and spam/ham ratio evolution for the US2 dataset (right)

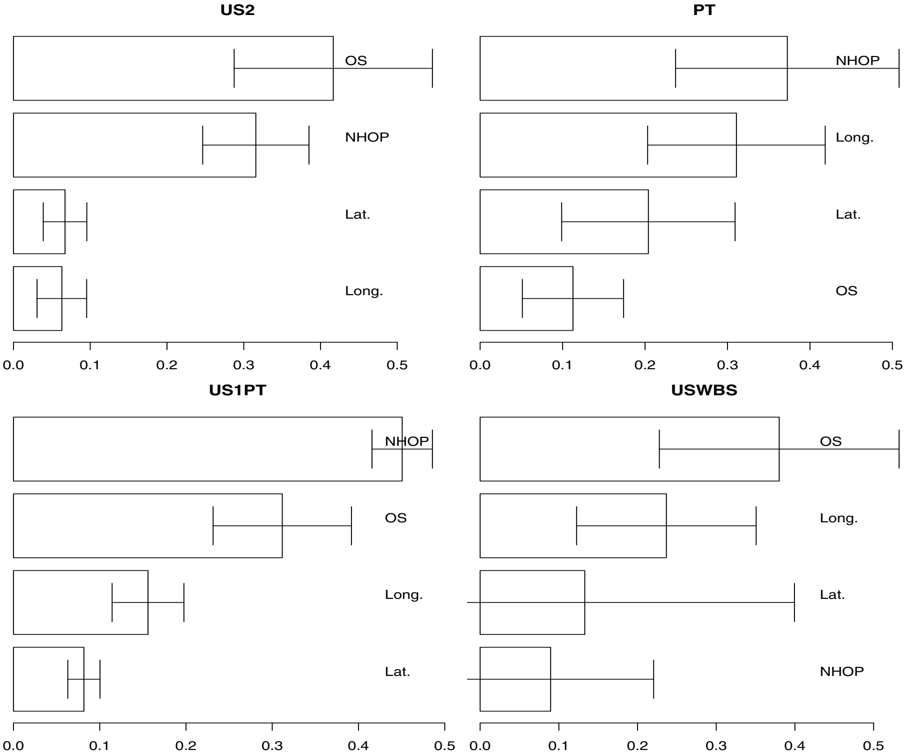


Fig. 6. Average input importances for the NSV model (hinges denote confidence intervals)

Table 5. Filter responses to phishing messages (values above 0.5 are in **bold**)

setup	B6	AB	WNB	WSV	NNB	NSV
US1	0.00	**1.00**	0.00	**0.62**	**1.00**	**0.99**
US1	0.00	**1.00**	0.00	0.36	**1.00**	**0.98**
US2	**1.00**	**1.00**	**1.00**	0.28	**1.00**	**1.00**
PT	0.00	0.00	**1.00**	0.35	**1.00**	**0.91**
US1PT	0.00	0.00	**1.00**	0.29	0.00	**0.96**

The average network-level feature importances for NSV are plotted in Figure 6. The bar plots show the $\overline{R_a}$ values, while the whiskers denote the 95% confidence intervals. The US1 importance bars are not shown, since they are similar to US2. All four attributes contribute to the model, although their relative influences vary. For example, the operating system (OS) is the most relevant feature for the US datasets, although it is the least important input for PT. On the other hand, the length of the message path (NHOP) is most relevant attribute for PT and US1PT.

To study the filtering vulnerability to phishing email attacks, we searched within the datasets for spam messages asking for user password details (e.g. related to a bank online account). Five messages were found and the respective spam probability predictions ($p(s|\mathbf{x}_j)$) are shown in Table 5. The first column of the table shows the dataset that contained such messages. Although the number of examples is not enough for a more definitive conclusion, the results seem to favor the network-level based methods. For a decision threshold of $D = 0.5$, NSV detects all attacks, while NNB predicts four. The less robust methods are B6 and WSV.

4 Conclusions

In this work, we proposed a new spam filtering approach that is based on four network-level attributes: message path length in terms of number of routers (NHOP), geographic coordinates (i.e. latitude and longitude) and operating system of the sender. We tested two data mining (DM) classifiers, Naive Bayes (NB) and Support Vector Machines (SVM) and also targeted two countries from different continents and with different main languages (i.e. U.S. and Portugal). Since our network-level properties are not currently monitored by filtering systems, we created and developed a new spam repository, called spam telescope. This repository includes real legitimate (ham) and non legitimate (spam) messages. The ham was collected from several mailing lists, while the spam was captured from email traps (fake addresses advertised through the Web). Several experiments were carried out, where a realistic mixture of spam and ham was used to simulate distinct user profiles.

When comparing with Content-Based filters (CBF), i.e. bag-of-words, and eight DNS-based Blackhole Lists (DNSBL), the NSV method (SVM fed with the four network-level features) obtained the best discriminatory performance, with high quality results (from 95.3% to 99.8%). The NSV method requires much less computation than the respective bag-of-words filter. Also, in contrast with the blacklist methods, it does not require communication with other servers, since the free geographic IP database that we used can be installed locally. Moreover, preliminary results suggest that NSV is more robust to phishing email attacks.

Based on the achieved results, we advise the use of the NSV filter, which provides a high true positive rate (i.e. detects most of the spam). To reduce false positives (i.e. ham marked as spam), this method could be used after a first phase blacklist filtering. Yet, for an effective blacklisting, it should be considered a careful DNSBL server selection or (even better) use of multiple DNSBLs.

Spammers and anti-spammers are in a continuous struggle. The research community has devoted a large attention to improve CBF. Yet, as argued in [17], spammers can easily change content to confuse CBF filters but network-level properties are more persistent in time. For example, a large portion of current spam comes from botnets. Most spammers are greedy and want a massive distribution of spam, thus they do not care about the location of a given controlled machine. Furthermore, some operating systems (e.g. Windows) are more vulnerable to botnet control by malicious software. Hence, we believe it is more difficult for spammers to surpass network-level based filters. As future work, we intend to enlarge the experiments to other countries (e.g. Spain) and access the full NSV robustness against phishing attacks by harvesting more phishing emails.

Acknowledgments

This work is supported by FCT grant PTDC/EIA/64541/2006. We also wish to thank David Manzieres for supporting Mail Avenger.

References

1. Beverly, R., Sollins, K.: Exploiting transport-level characteristics of spam. In: 5th Conference on Email and Anti-Spam, CEAS (2008)
2. Bilisoly, R.: Practical text mining with Perl. Wiley Publishing, Chichester (2008)
3. Blanzieri, E., Bryl, A.: A survey of learning-based techniques of email spam filtering. Artificial Intelligence Review 29(1), 63–92 (2008)
4. Cherkassy, V., Ma, Y.: Practical Selection of SVM Parameters and Noise Estimation for SVM Regression. Neural Networks 17(1), 113–126 (2004)
5. Cortes, C., Vapnik, V.: Support Vector Networks. Machine Learning 20(3), 273–297 (1995)
6. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems 47(4), 547–553 (2009)
7. Cortez, P., Lopes, C., Sousa, P., Rocha, M., Rio, M.: Symbiotic Data Mining for Personalized Spam Filtering. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI 2009), pp. 149–156. IEEE, Los Alamitos (2009)

8. Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. IEEE Transactions on Neural networks 10(5), 1048–1054 (1999)
9. Fawcett, T.: An introduction to ROC analysis. Pattern Recognition Letters 27, 861–874 (2006)
10. Flexer, A.: Statistical Evaluation of Neural Networks Experiments: Minimum Requirements and Current Practice. In: Proceedings of the 13th European Meeting on Cybernetics and Systems Research, Vienna, Austria, vol. 2, pp. 1005–1008 (1996)
11. Leiba, B., Ossher, J., Rajan, V.T., Segal, R., Wegman, M.: SMTP path analysis. In: Proceedings of the Second Conference on E-mail and Anti-Spam, CEAS (2005)
12. Lin, H.T., Lin, C.J., Weng, R.C.: A note on Platts probabilistic outputs for support vector machines. Machine Learning 68(3), 267–276 (2007)
13. MAAWG. Email Metrics Program: The Network Operators' Perspective. Report #10 – third and fourth quarter 2008, Messaging Anti-Abuse Working Group, S. Francisco, CA, USA (March 2009)
14. Metsis, V., Androutsopoulos, I., Paliouras, G.: Spam Filtering with Naive Bayes – Which Naive Bayes? In: Third Conference on Email and Anti-Spam, CEAS (2006)
15. Nelson, B., Barreno, M., Chi, F., Joseph, A., Rubinstein, B., Saini, U., Sutton, C., Tygar, J., Xia, K.: Exploiting Machine Learning to Subvert Your Spam Filter. In: 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats, pp. 1–9. ACM Press, New York (2008)
16. R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2009), ISBN 3-900051-00-3 http://www.R-project.org
17. Ramachandran, A., Feamster, N.: Understanding the Network-Level Behavior of Spammers. In: ACM (ed.) SIGCOMM 2006, pp. 291–302 (2006)

Domain-Specific Identification of Topics and Trends in the Blogosphere

Rafael Schirru, Darko Obradović, Stephan Baumann, and Peter Wortmann

German Research Center for Artificial Intelligence (DFKI)
Knowledge Management Department
Kaiserslautern & Berlin, Germany
{schirru,obradovic,baumann}@dfki.de,
p_wortma@cs.uni-kl.de

Abstract. Staying tuned to the trends and opinions in a certain domain is an important task in many areas. E. g., market researchers want to know about the acceptance of products. Traditionally this is done by screening broadcast media, but in recent years social media like the blogosphere have gained more and more importance. As manual screening of the blogosphere is a tedious task, automated knowledge discovery techniques for trend analysis and topic detection are needed.

Our system "Social Media Miner" supports professionals in these tasks. The system aggregates relevant blog articles in a specified domain from blog search services, analyzes their link structure and their importance, provides an overview of the most active topics and identifies general trends in the area. For every topic it gives the analyst access to the most relevant articles. Experiments show that our system achieves a high degree of sound automated processing.

1 Introduction

Besides the traditional media such as newspapers, radio, and television the World Wide Web (WWW) plays an increasing role as an information source for the shaping of public opinions. With the expansion of the Web 2.0 a shift in user behavior can be observed. More and more users are no longer only consumers but they become also producers of content. They contribute and comment pictures, videos, and bookmarks in resource sharing platforms, write reviews in online shops, and collaboratively collect information in wikis. In our article we focus on blogs as a medium that allows every user of the WWW to easily express her opinion about anything.

In this context, we collaborate with professional market researchers. For them it is essential to stay tuned about reviews and the acceptance of products, or about trends in the area of their interest. Traditionally, this is done by screening broadcast media, but in recent years, social media like the blogosphere have gained more and more importance for the evaluation of products and trends. A good indicator for this is Microsoft's PR action from December 2006, when they

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 490–504, 2010.

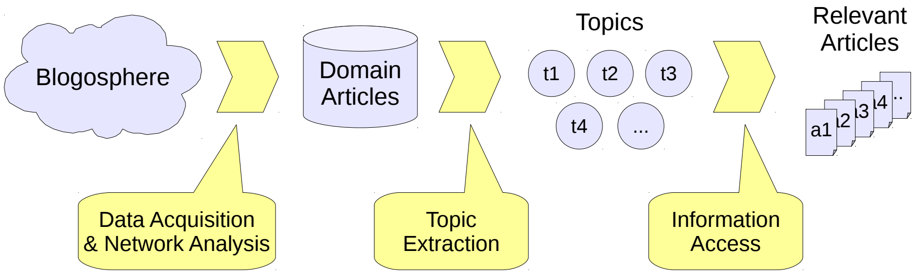


Fig. 1. System workflow

sent free Vista laptops to influential bloggers.[1] However, the large amount of available information sources, the problem to obtain a good overview of them, and the difficulty to rate their importance makes the monitoring of social media a tedious task if performed manually.

We developed a system named "Social Media Miner" to support professional market researchers in these tasks. There are three key actions that are automated by our system, which are illustrated in Figure 1. In the first action, the system aggregates relevant blog articles of a domain from different blog search services for maximal reach. In this data set, it analyzes the structure between the blogs and articles for ranking purposes. In the second action, we monitor the number of articles for each day. For periods of four days we extract the topics of the articles by applying textual data mining techniques. That way we provide an overview over the discussions in the domain. A user who is interested in a topic can select the most relevant terms from its label and in a third action, the system returns a ranked list of relevant articles as reading recommendations for exploring the selected topic.

The remainder of this article is structured as follows. In Section 2 we present related work in the fields of trend analysis and topic extraction. In Section 3 we describe how blog articles are aggregated from different search engines. Next, in Section 4 we briefly explain how networks are generated out of the data that allow us to derive social authority metrics for the articles. We depict the topic extraction process in Section 5 and the process of information access to articles of a topic in Section 6. Then we go on to present a first evaluation of our system in Section 7 before we conclude our findings and present our ideas for future work in Section 8.

2 Related Work

Research in the blogosphere can be roughly divided into two categories. The analysis of its structure and the analysis of its content. The structural analysis focuses on the ranking of blogs, the identification of communities and the

[1] http://apcmag.com/microsoft_sends_ferraris_to_bloggers.htm

dynamics of the blogosphere, mostly with Social Network Analysis (SNA) methods. The contentual analysis instead focuses on the blog articles and investigates what bloggers are writing about, usually with respect to a time dimension.

As explained in the introduction, we are following the second direction and investigate the current topics in the blogosphere. The analysis of its structure is however an essential tool for understanding the field and for ranking blog articles. Kumar et al. [1] describe how information evolves in the blogosphere, namely in "bursts" of increased activity. This increased activity does not necessarily imply that there is more real information around, but merely reflects its popularity.

In the area of contentual monitoring of the blogosphere, there exists the common idea of trend detection based on keyword frequencies, i. e., how often a certain keyword appears throughout all articles. The search service BlogPulse[2] implemented an automatic trend discovery for weblogs [2]. It is capable of identifying key persons or key phrases in the whole data set on a daily basis. This form of global monitoring is a very good indicator for the blogosphere as a whole, but it provides only insights for social researchers and curious individuals, hardly for market researchers interested in a specific domain, or any other focused observers. For these tasks, BlogPulse offers "trend search". Based on a query, it plots the number of matching articles per day over a certain time period. This gives a good impression of the general popularity of a domain or product, and periods of increased activity or buzz. You can also compare the trend lines of two different queries for a comparison of activity. This methods can provide deeper insights for specific keywords, but it fails to explain the curves any further.

The same trend lines are offered by other blog search services as well, be it commercial ones like Icerocket[3] or more research-oriented ones like BlogScope.[4] This method is currently well-established and state-of-the-art and practice.

To detect the topics in the corpus of blog postings our system uses algorithms from the domain of topic detection and tracking (TDT). TDT is concerned with finding and following new events in a stream of documents. In [3] the following TDT tasks have been identified: First is the segmentation task, i. e., segmenting a continuous stream of text into its several stories. Second, there is the detection task which comprises the retrospective analysis of a corpus to identify the discussed events and the identification of new events based on online streams of stories. Third is the tracking task where incoming stories are associated with events known in the system. In this work we focus on the detection of topics in a corpus of blog postings.

In [4] Schult and Spiliopoulou consider the problem of finding emerging and persistent themes in accumulating document collections which are organized in rigid categorization schemes such as taxonomies. They propose ThemeFinder, an algorithm for monitoring evolving themes from accumulating document collections. The algorithm works as follows: In the first period, it clusters all documents in the collection. In the following periods, it clusters the new documents with

[2] http://www.blogpulse.com/

[3] http://www.icerocket.com/

[4] http://www.blogscope.net/

the old feature space and compares the new clusters to the ones found in the previous period. If the clusters of two adjacent periods are similar with regard to their themes and if the quality of the clustering is not declining significantly, then the original feature space is kept. Otherwise a new feature space is build for the documents of the latest period and the next comparison. Thematic clusters are represented by a label, consisting of a set of terms that have a minimal support in the associated cluster. Thematic clusters that survived over several periods, despite re-clustering and changes of the feature space, will become part of the classification scheme. The authors put special emphasis on the evolution of topics over time. Our system deals with data from the Web 2.0, where many topics emerge in a short term and decay just as quickly. For that purpose we currently only focus on the topic detection task. Our approach combines statistical analysis (publication trend) with topic extraction techniques.

3 Data Acquisition

In order to find blog articles relevant to our domain, we define the appropriate keywords for a search query and regularly aggregate the search results from multiple blog search services. That way, we do not have to set up a complete search engine infrastructure by ourselves, and we can reach more articles than a single search service can offer, as our experiment will show.

3.1 Blog Search Service Analysis

In a preliminary step, we evaluate the quality and reach of five popular blog search services. These are Technorati,[5] Google Blogsearch,[6] Bloglines,[7] Icerocket and BlogPulse.

As the domain for this test, we have chosen the keyword "Henrietta Hughes", which unequivocally refers to an event on February 10th 2009, where this homeless person talked to US president Barrack Obama. The event had a major impact in broadcast media, as well as social media, especially the blogosphere.

Using this search query with the aforementioned services two weeks after this event, we aggregated and manually verified a total of 871 unique blog articles writing about this event. The most important finding concerns the percentage of articles each search service contributed to the aggregated article set. The best service, Icerocket, reached 51% here, while the other search services range between 21% and 35%. Thus, by aggregating results from multiple services, we can acquire a significantly larger data set as the basis for our analyses.

Concerning the validity of the search results, we discovered a number of unreachable sites, non-blog articles as well as presumably related pages that do not even mention the lady's name. Apart from Google Blogsearch's results, where only 51% were valid, i. e., blog articles on topic, the validity of the remaining services is between 84% and 93%.

[5] http://www.technorati.com/

[6] http://blogsearch.google.com/

[7] http://www.bloglines.com/

Consequently, we left Google's service out of the final aggregation component, and implemented a number of heuristics, based on the URL, meta-data and the site content, in order to filter out as much of the invalid results as possible.

3.2 The Aggregation Component

For our analyses, we need the URL of each blog article along with the date of publication, the title and the textual content. All search services allow to return the query results sorted by date, enabling us to exactly fetch the results in our individual time period of interest in the first step of the aggregation process. In a second step, each result is validated by the RSS entry on the blog site, fetching the accurate date, the full title and the available textual content.

Another important building block of our data is the link structure among these articles. We want to track all links, where the textual content of an article is referring to another current blog article in the domain. In most cases, the link targets will be articles already existing in the data set, but eventually, this will discover new relevant articles to be added to our data set. These links are used later as a social assessment of relevance and authority of articles, as widely known from PageRank [5] and similar methods. We impose some requirements on these article links, in order to include only expressive ones. First of all, links between articles on the same blog are ignored, since their expressiveness of authority is doubtful at best.

In a next step, we extract the underlying blog URLs out of the article URLs and gain a second type of data, the blogs. We then collect the blogroll links between these blogs, according to our method presented in [6]. These will serve as additional authority indicators in the subsequent network analysis.

4 Network Analysis

Our acquired data enables us to use SNA methods [7] to derive social authority values from the link structure.

Our data set can be represented in two networks that are linked with each other. We have a first directed network of articles, in which the nodes represent the blog articles and the edges represent the links between these articles. We also have a second directed network of blogs, in which the nodes represent the blogs in our data set, and the edges represent the blogroll links between them. These two networks are connected via a relation between the blog articles and their originating blog. These relations can be used to map metrics from one network to the other one.

We are interested in an authority value of blog articles for the reading recommendations on detected topics. First of all, we apply the PageRank algorithm [5] on the article network to obtain initial authority values for the articles. Alternatively, Kleinberg's HITS algorithm [8] can also be chosen for this task by the user. This authority is grounded on the fact, that often-cited articles are more likely to contain original and interesting content than less- or non-cited ones.

Second, the blog in which the article has been published is also a very strong indicator for the authority of an article, as the public usually trusts into a blog or blog author, not into a specific set of her articles. Therefore, we also calculate an authority value for each blog from the link structure of the blog network.

The final authority value of an article is the mean value of its initial authority from the article network and the authority value of its blog. That way, by the application of established network ranking algorithms, we can provide an importance indicator to the reader, when she has to choose articles from a given list. In our application this metric is used to sort the article list of a topic, so that the authoritative articles are listed first.

5 Topic Extraction

We identify the topics in our corpus of blog articles by applying textual data mining techniques. First we aggregate articles that were published within a time interval of four days. The time window is shifted by two days in each iteration of the algorithm, i. e., first time window from day 0 to day 3, second time window from day 2 to day 5, and so on. We will refer to these iterations of the algorithm as *runs* subsequently. The setting has been chosen as a typical use case, where a market researcher requests every two days an analysis of the domain for the last four days. However the size of the time window can be adapted to the needs and preferences of the user as well as to the publication volume in the respective domain.

The process steps of our topic extraction algorithm are depicted in Figure 2 and will be described in detail subsequently:

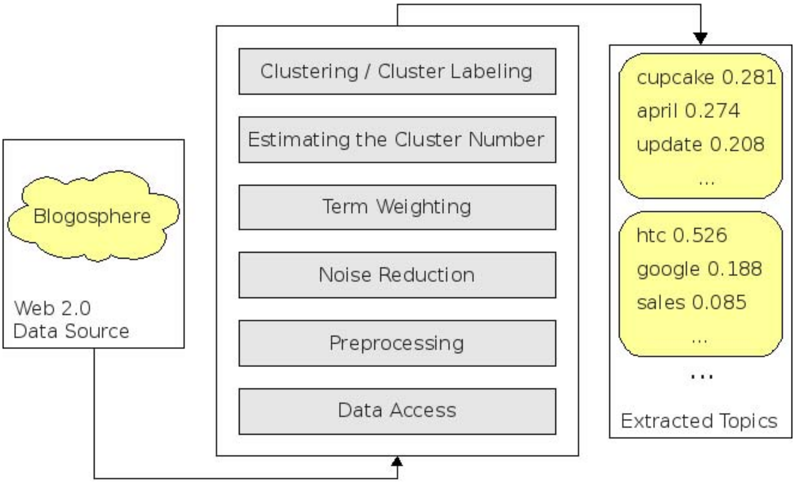


Fig. 2. Topic extraction process steps

Data Access. We use the titles of the blog articles as the input for the topic extraction algorithm, as they are considered to reflect the content of the associated articles appropriately in the majority of the cases. On account of the usually large number of articles per topic, a topic can still be detected reliably even if some titles do not perfectly describe the content of the respective articles. We also conducted experiments including the full text of the articles, however, the best clustering results were achieved when only the titles of the blog articles were used.

Preprocessing. We convert the terms contained in the titles to lower case characters, remove punctuation characters and stop words. Further stemming is applied to bring the terms to a normalized form. We use the Snowball stemmer[8] for this purpose. The normalized profiles of the blog articles are represented according to the "bag-of-words" model, i. e., they are represented as vectors where the features correspond to the terms in the corpus and the feature values are the counts of the words in the respective articles.

Noise Reduction. Very rare and very frequent terms are not considered helpful to characterize articles. As a consequence dimensions representing these terms are removed. To reduce the noise that is inherent in social metadata we experimented with dimensionality reduction based on Latent Semantic Analysis ([9]). However the positive impact of the application of this technique still has to be examined in greater depth.

Term Weighting. Terms that appear frequently in the data/metadata of one article but rarely in the whole corpus are likely to be good discriminators and should therefore obtain a higher weight. We use the TF-IDF measure ([10]) which is widely applied in information retrieval systems in order to achieve this goal.

Clustering and Cluster Labeling. To be able to cluster the blog articles we need to find a reasonable number of clusters in our data first. For this purpose we follow an approach which is based on the residual sum of squares (RSS) in a clustering result. For document clustering and cluster label extraction we apply non-negative matrix factorization.

We *estimate the number of clusters* in the data set as described in [11], page 365. First we define a range in which we expect to find the number of topics per run. We chose a range between 2 and 20 for our experiments, however the boarders are configurable in our algorithm. For each potential cluster size k ($2 \leq k \leq 20$) we run K-Means i-times (we chose $i = 10$), each time with a different initialization. We compute for each clustering the residual sum of squares (RSS) and the minimum RSS over all i clusterings (denoted by $\widehat{RSS}_{min}(k)$). Then we take a look at the values $\widehat{RSS}_{min}(k)$ and search for the points where successive decreases in $\widehat{RSS}_{min}$ become significantly smaller.[9] The first five such values $k-1$ are stored as reasonable cluster sizes. We store five values in order to enable

[8] http://snowball.tartarus.org/

[9] $RSS_{min}(k)$ is a monotonically decreasing function in k with minimum 0 for $k = N$ with N being the number of documents.

clusterings according to different granularities. If broad clustering granularity is desired we take the first reasonable number of clusters, for middle granularity the second, and so on.

Using non-negative matrix factorization (NMF) for *document clustering* has firstly been introduced by Xu et al. ([12]). The authors show that NMF-based document clustering is able to surpass latent semantic indexing and spectral clustering based approaches.

NMF finds the positive factorization of a given positive matrix. It is applied on the term-document matrix representation of the document corpus. In the latent semantic space which is derived by applying NMF, each axis represents the base topic of a document cluster. Every document is represented as an additive combination of these base topics. Associating a document with a cluster is done by choosing the base topic (axis) that has the highest projection value with the document. Formally NMF is described as follows:

Let $W = \{f_1, f_2, ..., f_m\}$ be the set of terms in the document corpus after our preprocessing steps. The weighted term vector X_i of a document is defined as

$$X_i = [x_{1i}, x_{2i}, ..., x_{mi}]^T \tag{1}$$

with x_{ij} being the TF-IDF weights of the terms f_i as described before.

We assume that our document corpus consists of k clusters. The goal of NMF is to factorize X into non-negative matrices U $(m \times k)$ and V^T $(k \times n)$ which

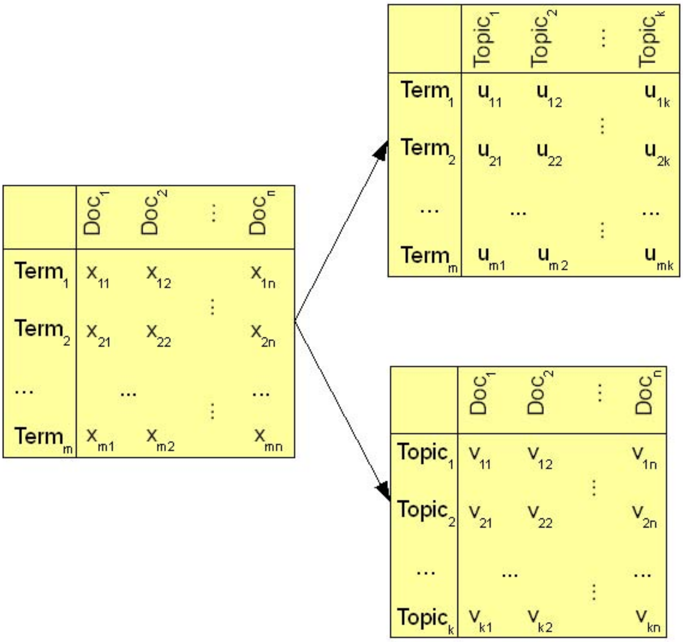


Fig. 3. Factorization of the term-document matrix by the NMF algorithm

minimize the following objective function:

$$J = \frac{1}{2} \parallel X - UV^T \parallel \tag{2}$$

$\parallel \cdot \parallel$ denotes the squared sum of all the elements in the matrix.

Each element u_{ij} of matrix U determines the degree to which the associated term f_i belongs to cluster j. For cluster labeling we simply choose for each cluster the ten terms with the highest degree of affiliation. Analogously each element v_{ij} of matrix V represents the degree to which document i is associated with cluster j. To cluster the documents, again we assign every document to the cluster with the highest degree of affiliation. If a document i clearly belongs to one cluster x then v_{ix} will have a high value compared to the rest of the values in the i'th row vector of V. The matrix factorization is depicted in Figure 3.

In our previous work, we used the X-means algorithm ([13]) to cluster our blog articles. For cluster label extraction, frequency-based cluster labeling as well as feature selection methods such as mutual information and the chi-square test ([11] pages 396-398) have been used. Our experiments showed that the results using NMF were more gratifying, in particular with respect to meaningful cluster labels.

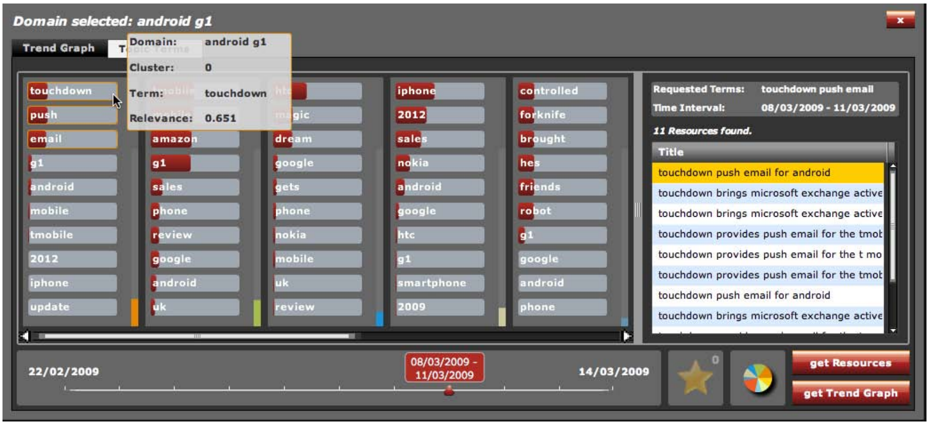
6 Information Access

For each run the detected cluster labels are displayed to the user. Every cluster is associated with a label consisting of at most ten terms and their respective relevance values. The presentation of topic terms together with their associated relevance values in the Social Media Miner web interface is shown in Figure 4. Our system offers three options to access the relevant postings of a topic:

1. Get all postings in a cluster.
2. Get postings in a cluster that match specified search terms.
3. Get postings of the current run that match specified search terms.

In order to provide access to all relevant postings in the current run and to avoid the presentation of wrongly clustered postings to the user we chose the third option for our GUI. The approach of deducing a cluster label first and then re-querying the input documents has been proposed in the literature before (e. g., [14]). In our system the user is currently required to select the relevant terms of a topic manually. However we plan to automate this step in the next version of the Social Media Miner.

The result set of this query can be evaluated with precision and recall values [11], whereby the precision value is the critical one. The query should ideally return only articles that belong to the topic of interest, i. e., the precision value should be close to 1.0. Given that the system achieves indeed a high precision, a high recall value is not overly important, since a few good articles, or even only one in certain topics, are enough to get all the facts and information the topic


Domain selected: android g1

Domain: android g1

Cluster: 0

Term: touchdown

Relevance: 0.651

Requested Terms: touchdown push email

Time Interval: 08/03/2009 - 11/03/2009

11 Resources found.

Title
touchdown push email for android
touchdown brings microsoft exchange active
touchdown brings microsoft exchange active
touchdown provides push email for the tmob
touchdown provides push email for the t mo
touchdown provides push email for the tmob
touchdown push email for android
touchdown brings microsoft exchange active

Fig. 4. Social Media Miner interface for the presentation of topics in a selected run

contains. However, from a user's intuitive point of view, it is desirable to obtain a longer list of relevant articles for a very active topic, than for a less active one. This expectation can be met with relatively constant recall values over the queries, if the precision is reliably high at the same time.

We rank the result list according to the authority value of the articles, as described in Section 4. Thus, the resulting reading recommendation gives the user an impression of the popularity and efficient access to the important and relevant articles of the topic, given that precision and recall values fulfill the conditions postulated before.

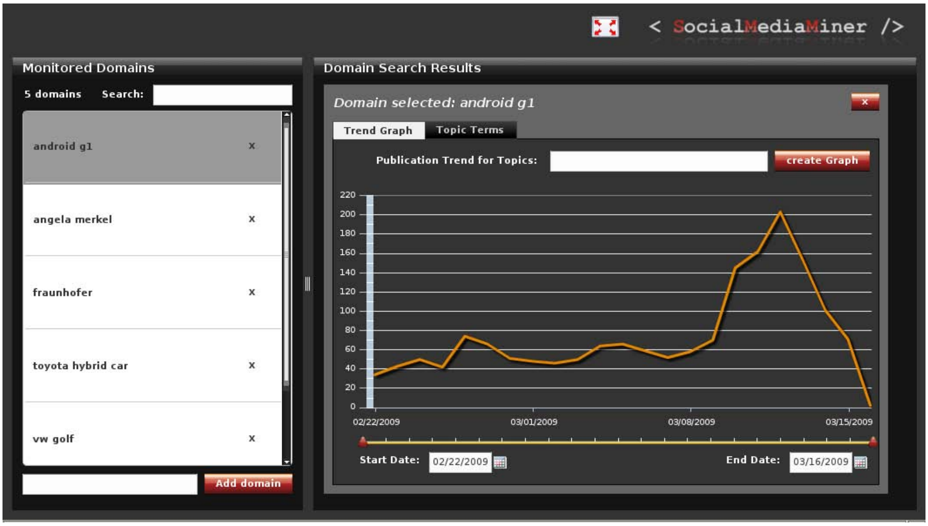
Besides providing access to the relevant blog postings of a topic in a run, we also want to support the users in identifying whether a topic of interest is of increasing or decreasing importance in the blogosphere. For that purpose, the user chooses a topic of interest, again selects the relevant terms of the topic and clicks the "get Trend Graph" button. A graph is generated and displayed that depicts the publication trend of articles matching the selected terms during the time the domain has been tracked. That way the user can easily determine the current relevance of the associated topic in the blogosphere.

7 Evaluation

7.1 Example Data Set

We evaluate our system by comparing its suggestions with manually categorized topics for the articles. As a test domain, we have chosen the relatively new and hyped mobile handset G1 launched by T-Mobile in 2008, with its Google Android software as the most important feature. This is a domain that is interesting for marketing professionals or market researchers in the mobile phone sector.

The resulting search query is "Android G1", and the data has been acquired as described in Section 3 at the end of March 15th 2009, with a time frame of



5 domains Search:

android g1 x
angela merkel x
fraunhofer x
toyota hybrid car x
vw golf x

Domain selected: android g1

Start Date: 02/22/2009

End Date: 03/16/2009

Fig. 5. Social Media Miner interface for the publication trend in a domain (i. e., number of blog articles per day)

the last 22 days, including February 22nd 2009 as day 0. The search services returned 2193 unique URLs, from which our heuristics validated 1710 as blog articles of the domain, for which a timestamp and the content were available. From the originally 693 links between these articles, only 350 adhered to our criteria, but are supposed to be expressive. The 1710 articles were published in 931 different blogs. Between these blogs, we detected 264 blogroll links.

Figure 5 shows the number of published articles by day in our test data set. This is how a traditional trend analysis would present the data, as outlined in Section 2. Obviously, this method detects three peaks of activity in our data set, leaving the observer alone to find out about the reasons and topics behind them. With almost 400 blog articles in the main peak on the days 17 and 18, this is a time-intensive task to perform for a human being.

7.2 Ground Truth

For the evaluation, we have looked through all of the articles and categorized them into topics. A topic is relating multiple articles by either a specific event in the domain, or by a common entity, which is not the domain itself of course. We found 57 different topics with at least three articles in the data set. 775 blog articles did not belong to any topic, e. g., reviews of an author's new G1 phone.

Figure 6 plots the Top 7 topics of the domain with their volume of articles per day. The ground truth reveals that there exist two different kinds of topics, which are very good to distinguish from each other. Event-based ones and entity-based ones. The articles of an event-based topic usually appear around a certain peak day, in a frame between three and five days, like the announcement of the

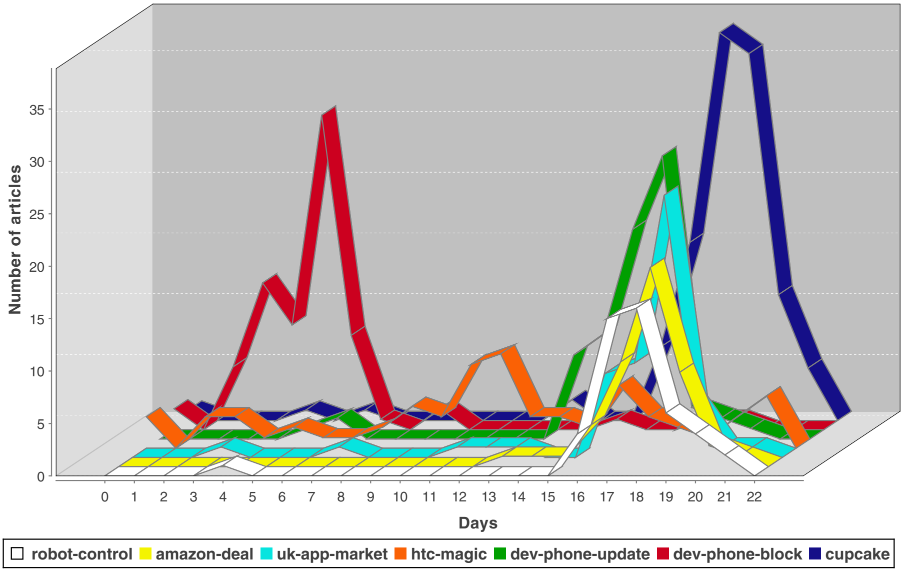


Fig. 6. Ground Truth: Topics in example domain

cupcake update for April. Entity-based topics on the other hand appear more or less intensive throughout the whole time period, e. g., the speculations and discussions about HTC's next android device codenamed "Magic".

Another observation is the composition of peaks in the overall publication trend. Figure 6 disguises that the publication peak at the end of the observed period (see Figure 5) is not expressive for the domain, but mostly a result of four concurrent large-volume events, which occur around the same time by coincidence. This illustrates very well the limits of "Trend Graphs" as they are used in search services today, and the need for a topic detection that can explain the publication trend and its composition in more detail.

7.3 Topic Detection

For the topic detection we aim to identify such topics for which at least 10 blog postings are available in a run. To evaluate the topic detection step, we assigned each cluster whose label indicated a certain topic to the respective topic in our ground truth data set. To objectify this manual step, we require that our system presents at least three relevant articles for the topic in the top ten list of recommended blog postings which corresponds to a precision of 0.3. However the average precision is much higher (> 0.8). Table 1 shows how many topics have been identified in each run. Altogether 29 of 37 topics (78.38%) could be detected with our approach.

Table 1. Detected topics in runs

Run	#Articles	#Topics to detect	#Topics detected
0 - 3	169	1	1
2 - 5	232	1	1
4 - 7	239	2	2
6 - 9	195	2	2
8 - 11	226	1	1
10 - 13	241	1	1
12 - 15	239	2	1
14 - 17	435	8	8
16 - 19	663	9	6
18 - 21	528	10	6

Table 2. Average precision and recall values for different term selection heuristics

	Top3	Top5	Top10
Precision	0.84	0.87	0.87
Recall	0.35	0.35	0.33

7.4 Information Relevance

To get access to the relevant postings of a topic the user selects the terms that best characterize the topic and clicks on a search button. We chose three heuristics for the term selection step. Top3 refers to the three most relevant terms in the label, Top5 and Top10 to the five and ten most relevant terms respectively. Additionally we require that the relevance of each selected term is not less than 50% of the relevance of the most relevant term. Terms whose relevance is less are dismissed.

Precision and recall values are calculated over the top 10 recommended blog articles for each topic. With the Top3 heuristic we achieved a precision of 0.84 and with the Top5 and Top10 terms a precision of 0.87 in averaged over all detected topics. The average recall values were 0.35 for the Top3 and Top5 approaches and 0.33 for the Top10 approach. The average precision and recall values for the different term selection approaches are summarized in Table 2. Concerning the recall values it has to be considered that the amount of postings on a topic is usually too high for the user to check all of them. For that purpose we aim at finding a smaller set of relevant postings for each topic. With an average precision of 0.87 for the Top5 and Top10 approaches we can in general present the user eight to nine relevant postings for each topic. In our future work we will examine how the amount of relevant postings can be restricted to a smaller set by exploiting the relevance values derived from the SNA algorithms thus making higher recall values possible.

8 Conclusion and Future Work

By combining methods from social network analysis and textual data mining, we set up a system for the semi-automatic analysis of topics and trends in selected domains in the blogosphere, therewith supporting the work of professionals in market research or public relations businesses.

In the next version of our system we plan to automate the connection of topic terms with relevant blog postings of the associated topic thus making the manual term selection step obsolete. Further we plan to integrate topic tracking algorithms that allow for a visualization of publication trends of specific topics that way improving the perceptibility of trends in an early stage.

A new insight revealed during this work is the fact that links between blog articles cannot only be used to measure article authority, but they also give strong hints for the topic clustering in the domain. We intend to integrate this network component information into the clustering algorithm and improve it further that way.

Acknowledgments. This research has been financed by the IBB Berlin in the project “Social Media Miner”, and co-financed by the EFRE fonds of the European Union. Special thanks to Fernanda Pimenta for graphical assistance.

References

1. Kumar, R., Novak, J., Raghavan, P., Tomkins, A.: On the bursty evolution of blogspace. World Wide Web 8(2), 159–178 (2005)
2. Glance, N., Hurst, M., Tomokiyo, T.: Blogpulse: Automated trend discovery for weblogs. In: WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics, ACM Press, New York (2004)
3. Allan, J., Carbonell, J., Doddington, G., Yamron, J., Yang, Y.: Topic detection and tracking pilot study. In: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, pp. 194–218 (1998)
4. Schult, R., Spiliopoulou, M.: Discovering emerging topics in unlabelled text collections. In: Manolopoulos, Y., Pokorný, J., Sellis, T.K. (eds.) ADBIS 2006. LNCS, vol. 4152, pp. 353–366. Springer, Heidelberg (2006)
5. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, Stanford (1998)
6. Obradovic, D., Baumann, S.: Identifying and analysing germany’s top blogs. In: Proceedings of the 31st German Conference on AI, pp. 111–118. Springer, Heidelberg (2008)
7. Wasserman, S., Faust, K., Iacobucci, D.: Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences). Cambridge University Press, Cambridge (1994)
8. Kleinberg, J.M.: Authoritative Sources in a Hyperlinked Environment. In: Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 668–677. AAAI Press, Menlo Park (1998)
9. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. Journal of the American Society of Information Science 41(6), 391–407 (1990)

10. Jones, K.S.: A statistical interpretation of term specificity and its application in retrieval. Journal of Documentation 28(1), 11–21 (1972)
11. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to Information Retrieval. Online edn. Cambridge University Press, Cambridge (April 2009)
12. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: SIGIR 2003: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp. 267–273. ACM, New York (2003)
13. Pelleg, D., Moore, A.W.: X-means: Extending k-means with efficient estimation of the number of clusters. In: ICML 2000: Proceedings of the Seventeenth International Conference on Machine Learning, pp. 727–734. Morgan Kaufmann Publishers Inc., San Francisco (2000)
14. Osinski, S., Stefanowski, J., Weiss, D.: Lingo: Search results clustering algorithm based on singular value decomposition. In: Intelligent Information Systems, pp. 359–368 (2004)

Combining Business Process and Data Discovery Techniques for Analyzing and Improving Integrated Care Pathways

Jonas Poelmans[1], Guido Dedene[1,4], Gerda Verheyden[5], Herman Van der Mussele[5],
Stijn Viaene[1,2], and Edward Peters[1,3]

[1] K.U.Leuven, Faculty of Business and Economics, Naamsestraat 69,
3000 Leuven, Belgium
[2] Vlerick Leuven Gent Management School, Vlamingenstraat 83,
3000 Leuven, Belgium
[3] OpenConnect Systems, 2711 LBJ Freeway Suite 700,
Dallas, TX 75234, United States of America
[4] Universiteit van Amsterdam Business School, Roetersstraat 11
1018 WB Amsterdam, The Netherlands
[5] Sint-Augustinus hospital, Oosterveldlaan 24,
2610 Wilrijk, Belgium
{Gerda.Verheyden,Herman.Vandermussele}@gza.be,
{Jonas.Poelmans,Stijn.Viaene,Guido.Dedene}@econ.kuleuven.be,
epeters@oc.com

Abstract. Hospitals increasingly use process models for structuring their care processes. Activities performed to patients are logged to a database but these data are rarely used for managing and improving the efficiency of care processes and quality of care. In this paper, we propose a synergy of process mining with data discovery techniques. In particular, we analyze a dataset consisting of the activities performed to 148 patients during hospitalization for breast cancer treatment in a hospital in Belgium. We expose multiple quality of care issues that will be resolved in the near future, discover process variations and best practices and we discover issues with the data registration system. For example, 25 % of patients receiving breast-conserving therapy did not receive the key intervention "revalidation". We found this was caused by lowering the length of stay in the hospital over the years without modifying the care process. Whereas the process representations offered by Hidden Markov Models are easier to use than those offered by Formal Concept Analysis, this data discovery technique has proven to be very useful for analyzing process anomalies and exceptions in detail.

Keywords: Breast cancer, process mining, data discovery, integrated care pathways.

1 Introduction

An increasingly competitive health care market forces hospitals to search for ways to improve their processes in order to deliver high quality of care while at the same time

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 505–517, 2010.

reducing costs [1]. According to [14], the solution to poor quality is not to increase the supply of physicians or specialists or hospital beds, but instead to improve health care systems and incentives to ensure that existing physicians and hospitals provide the best possible quality at the lowest cost. Integrated care pathways are structured multidisciplinary care plans which detail the essential steps in the care process of a population of patients with a certain clinical problem [3]. The aims to achieve with care pathways are improving quality and efficiency of care, to standardize the outcomes of the provided care, to facilitate communication between healthcare professionals and to allow for systematic continuing audit. Care pathways are business process models which describe the expected progress of the patient through the care process and try to model the most standard frequent care pathway, based on expert prior knowledge.

Till date, the continuous monitoring, analysis and improvement of the care pathway's performance was performed in an ad hoc, manual and labor-intensive way. This approach however has some limitations. Modifications to the care process are performed in an ad hoc way and their success can only be measured by the impact of these modifications on the Key Performance Indicators (KPIs). This retrospective impact analysis can only be done after several months, which is an unacceptable long time window in healthcare management. Moreover, this standard model does not capture process variations, nor process exceptions and the root causes for inefficiencies are not known. Moreover, in practice there is often a significant gap between what is prescribed or supposed to happen and what actually happens. Process mining is an interesting method for gaining insight into what happens in a healthcare process for a group of patients with the same diagnosis.

In [6] the applicability of process mining in the healthcare domain was investigated, using Petri-Nets. The idea of process mining [12] is to extract, monitor and improve real processes by extracting knowledge from event logs.

In this paper, we use a unique combination of process discovery techniques and data discovery techniques to gain a deeper understanding of an existing breast cancer care process and the actual activities performed on the working floor to discover process inefficiencies, exceptions and variations immediately and to search for the root causes of inefficiencies. We propose and use a new approach based on Hidden Markov Models to discover a process model from event sequences. Formal concept Analysis (FCA) is used to analyze the characteristics of the clusters of patients that emerged from this process discovery exercise and vice versa to find groups of patients to feed into the process discovery methods.

The remainder of this paper is composed as follows. In section 2 we introduce the essentials of business process discovery, Hidden Markov Models and the HMM-based techniques that are proposed for process discovery. In section 3, we elaborate on FCA as a data discovery technique. In section 4, we discuss the dataset used. Section 5 describes the methodology and the results of our discovery exercise. Finally, section 6 rounds up with conclusions.

2 Business Process Discovery

In contrast to process modeling, which is developing a top-down representation of a "to-be" process reality, process discovery is a bottom up approach that tries to gain an understanding of the as-in process realities that are existing at the operational work floor. Discovering irregularities, exceptions and variations by means of analytics is essential in developing process and workforce intelligence. Statistical techniques often consider exceptions as nuisance information and eliminate them as noise. According to [8], statistical techniques are able to capture the general process model rather than the process model containing exceptional paths. For discovering process exceptions, anomalies and variations, the combination of learning techniques, mining and clustering is required to gain sufficient insights in the processes. Most workflow mining methods use Petri-Net like models. In [7], simulated process logs of hospital-wide workflows, containing events like "blood test" or "surgery" were used to build Petri-Net like models. In [2] a statistical approach, using Hidden Markov Models (HMMs) is taken to model the workflow inside the Operation Room. These probabilistic models offer a greater degree of flexibility and are a better option for healthcare, where traditional process mining techniques do not work well [4].

2.1 Hidden Markov Models

A Hidden Markov Model (HMM) is a statistical technique that can be used to classify and generate time series. A HMM [13] can be described as a quintuplet $I = (A, B, T, N, M)$, where N is the number of hidden states and A defines the probabilities of making a transition from one hidden state to another. M is the number of observation symbols, which in our case are the activities that have been performed to the patients. B defines a probability distribution over all observation symbols for each state. T is the initial state distribution accounting for the probability of being in one state at time $t = 0$. For process discovery purposes, HMMs can be used with one observation symbol per state. Since the same symbol may appear in several states, the Markov model is indeed “hidden”.

We visualize HMMs by using a graph, where nodes represent the hidden states and the edges represent the transition probabilities. The nodes are labelled according to the observation symbol probability.

2.2 HMM-Based Process Discovery

There are multiple advantages of using HMMs for process discovery:

- A lot of (open source) algorithms have been published for analyzing and understanding HMMs (e.g. Expectation Maximization, Viterbi algorithm for most probable path for a given pattern of observations, etc.)
- Micro patterns of actor behavior (e.g. medical acts that belong together) can be easily aggregated into one single state in HMMs. Transitions of 100%. probability can be aggregated into one single state of activity.
- HMMs can be annotated with a variety of attributes, such as (risk and transition) probabilities, time duration, variances, etc.
- HMMs offer better possibilities to match the models obtained from process discovery with the training/learning datasets. In particular, parallel activities are filtered out in HMMs.

In this paper the standard HMM MATLAB toolbox developed by Kevin Murphy was used [9]. The patient data were transformed into sequences, and the Expectation Maximization (EM, also known as Baum-Welch) algorithm was used to produce the results for this paper. This algorithm combines both forward and backward learning techniques for training an HMM as a process model. The input data were organized according to the Event – Object – Actor standard for process mining input. In this case the input data were obtained from standard clinical patient reporting datasets, compatible with the Healthcare Level 7 record standard.

The only large scale commercial toolset for process discovery (including not only the process analytics, but also the automatic non-invasive gathering of input data) is provided by OpenConnect in its Comprehend product family.

3 Data Discovery with Formal Concept Analysis

Formal Concept Analysis [5] is a data analysis technique that supports the user in analyzing the data and discovering unknown dependencies between data elements. In particular, the visualization capabilities are of interest to the domain expert who wants to explore the information available, but at the same time has not much experience in mathematics or computer science. The details of FCA theory and how we used it for KDD can be found in [11].

Traditional FCA is mainly using data attributes for concept analysis. In this paper the process activities (events) are used as the attributes, whereas the patients are used as the objects in the cross-table that is used as input for FCA. In analogy with [11] where coherent data attributes were clustered to reduce the computational complexity of FCA, coherent events have been clustered in this study.

4 Dataset

Our dataset consists of 148 breast cancer patients that were hospitalized during the period from January 2008 till June 2008. They all followed the care trajectory determined by the clinical pathway Primary Operable Breast Cancer (POBC), which structures one of the most complex care processes in the hospital. The treatment of breast cancer consists of 4 phases in which 34 doctors, 52 nurses and 14 paramedics are involved. Fig. 1 contains a high-level summary of the breast cancer care process. Before the patient is hospitalized, she ambulatory receives a number of pre-operative investigative tests. During the surgery support phase she is prepared for the surgery she will receive, while being in the hospital. After surgery she remains hospitalized for a couple of days until she can safely go home. The post-operative activities are also performed in an ambulatory fashion. Every activity or treatment step performed to a patient is logged in a database and in the dataset we included all the activities performed during the surgery support phase to each of these patients.

Each activity has a unique identifier and we have 469 identifiers in total for the clinical path POBC. Using the timestamps assigned to the performed activities, we turned the data for each patient into a sequence of events. These sequences of events were used as input for the process discovery methods. We also clustered activities with a similar semantical meaning to reduce the complexity of the lattices and process models.

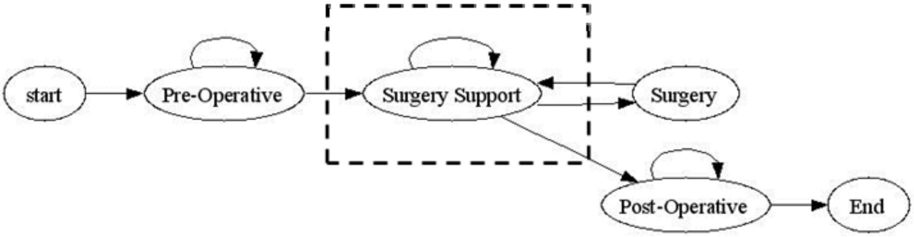


Fig. 1. Breast cancer care process

5 Analysis and Results

One of the most important tasks of the care process manager is to gain insight into what's happening on the working floor. The goal was to develop an approach that optimally supports this manager's role. The synergy of process and data discovery techniques in healthcare we propose, has some major advantages over the traditional way of working:

- Significantly reduces the workload for the care process manager who has to monitor over 42 care processes.
- Many unknown data dependencies are revealed that stay hidden for traditional statistical analysis techniques, which typically only look at one or two aspects of the process simultaneously.
- Provides a structured method for finding knowledge gaps, outliers, quality of care issues, process anomalies and inefficiencies.
- Much more information is provided to the process manager, much more quickly. This allows for better analysis and real-time anticipation on potential problems, whereas in the past, this could be done only after a yearly, very time-consuming and labor-intensive retrospective data analysis.
- The method allows the user to zoom in on different aspects of the provided care.

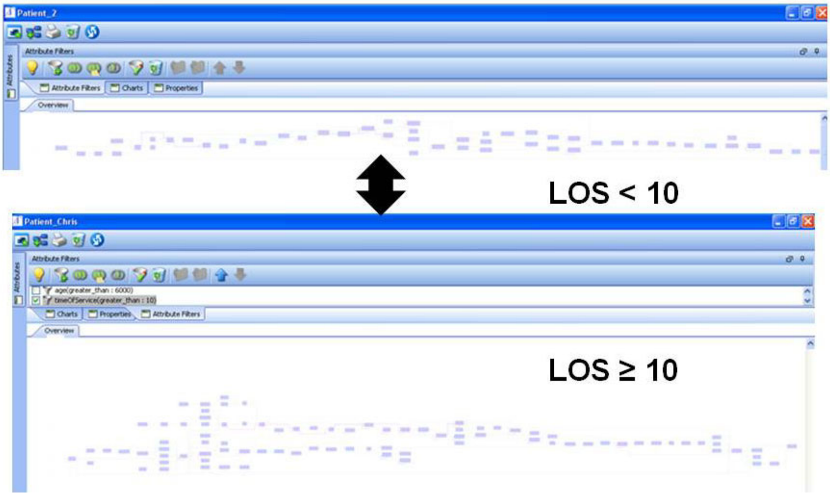
The process models allow for the extraction and visualization of the most frequent standard care pathway. While analyzing these models, we observed many anomalies and process exceptions that were hard to explain. Therefore, we used FCA to zoom in on and analyze these observations in detail.

5.1 Quality of Care Analysis

Our initial process model was built from the full dataset with 148 patients and 469 activity codes. We observed a relatively linear process for the group of patients with a length of stay in the hospital less than 10 days. However, there were 12 patients for which the process model was very complex. They all had in common that their length of stay in the hospital was longer than 9 days. Fig. 2 contains screenshots from the output produced by the Comprehend toolset. The upper part displays the obtained process map on the set of patients with a length of stay lower than or equal to 10 days in the hospital and the lower part displays the obtained map for the patients with a length of stay lower than 10 days.

We built an FCA lattice to explore their characteristics. This lattice gave us some first interesting insights in the problem. We will try to summarize the most important ones.

- One of our clinical indicators is the pain score which tells us at which days the pain experienced by patients reaches its highest level. We always saw peaks on 1 and day 4 of hospitalization however until now we had no idea why. The lattice gave us an interesting suggestion that this might be due to an overlooked connection between removal of the wound drains and insufficient pain medication. We were able to find that wound drains is probably the most contributing factor to an increased pain score experience by patients and that pain medication should be administered before removing the drains (= improving quality of care).
- We were able to find a quality problem in the care provided to these 12 patients. For 1 patient the history record (containing amongst others clinical, psychosocial information) was not consulted prior to the start of treatment. This may result in an inappropriate nursing care thereby potentially neglecting physical and psychosocial patient needs.
- Probably one of the main reasons of the increased length of stay we found to be the following: neurological/psychiatric problems, wound infection, subsequent bleeding. This makes the care process more complex and result in more investigative tests. Since these additional morbidities are probably one of the root causes for this increased length of stay, there treatment should be anticipated on and optimalized during the preoperative phase.



□ age(greater_than : 6000)
☑ timeOfService(greater_than : 10)

Fig. 2. Comprehend process map for patients with a length of stay smaller than 10 days (upper part) and process map for patients with a length of stay larger or equal than 10 days (lower part)

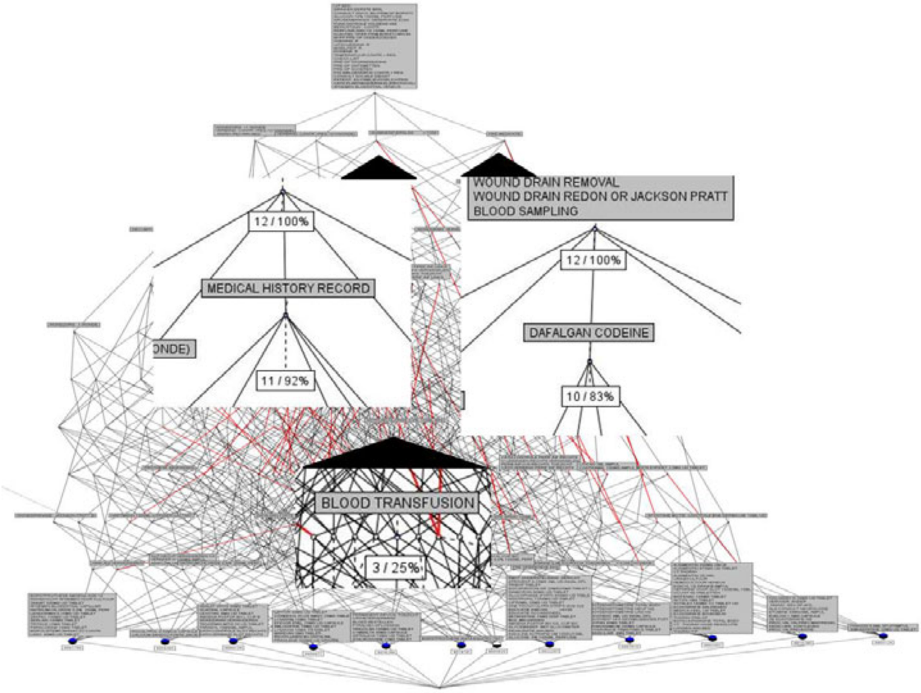


Fig. 3. Lattice containing 12 patients with length of stay larger or equal to 10 days

5.2 Process Variations

There are five types of breast cancer surgery: mastectomy, breast conserving surgery, lymph node removal and the combination of either mastectomy or breast conserving surgery with lymph node removal. For each of these surgery types, we extracted the corresponding patients in the dataset and constructed a process model and an FCA lattice for in-depth analysis of the characteristics of these groups.

Mastectomy surgery consists of completely removing the breast and during breast conserving surgery only the tumor is removed. The process models showed that the complexity of the care process is much larger for the mastectomy patients. Since mastectomy is a more complex surgery type, we expected that the FCA lattices would also be more complex than for breast conserving surgery. Surprisingly we found out that this was not true. The complexity of the lattice was larger for the breast conserving surgery patients and we found that this was due to the less uniform structure of this care process, in which for many patients some essential care interventions were missing. Fig. 4 contains the interventions performed to the 60 patients receiving breast-conserving surgery with lymph node removal. The lattice shows that 3 of these patients did not receive a consultation from the social support service. 15 patients did not have an appointment with a physiotherapist and did not receive revalidation therapy. 1 patient did not receive a pre-operative preparation and 2 patients were missing emotional support before and after surgery.

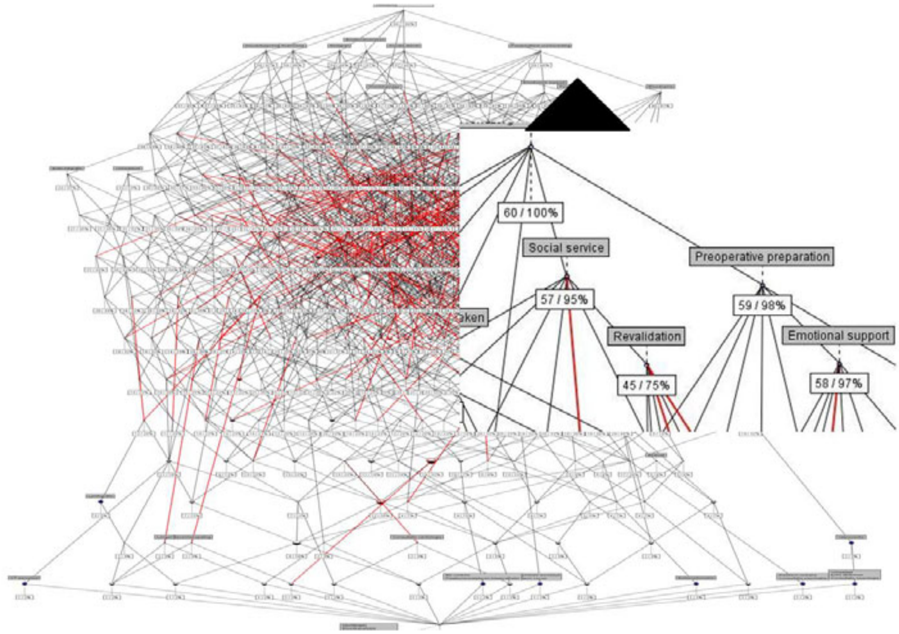


Fig. 4. Lattice containing 60 patients receiving breast-conserving surgery with lymph node removal

The originally developed breast-conserving surgery care pathway was written for a certain length of stay for the patients in the hospital. This length of stay was significantly reduced over the past years without modifying the care process model. As a consequence, we found it became impossible to execute the prescribed process model in practice and patients are receiving suboptimal care. The activities performed to the patients should be reorganized and a new care pathway, taking into account this time restriction, should be developed.

Fig. 5 shows the lattice for the 37 patients receiving mastectomy surgery with lymph node removal, which has a much less complex structure than the lattice for the breast conserving surgery with lymph node removal. For the mastectomy patients, we found that most patients received all key interventions prescribed in the clinical pathway. Only for two patients there was a quality of care issue, namely 1 patient did not receive emotional support and 1 patient did not receive a breast prosthesis. These shortcomings in the provided care however may have serious consequences for her psychological well-being.

5.3 Workforce Intelligence

We also made a lattice for each type of surgery in which we used as attributes the names of the surgeons and the length of stay of the patients in the hospital. We calculated the average length of stay of the patients and looked at how many patients stayed longer, equal or shorter than this average time of' stay. Fig. 6 contains the

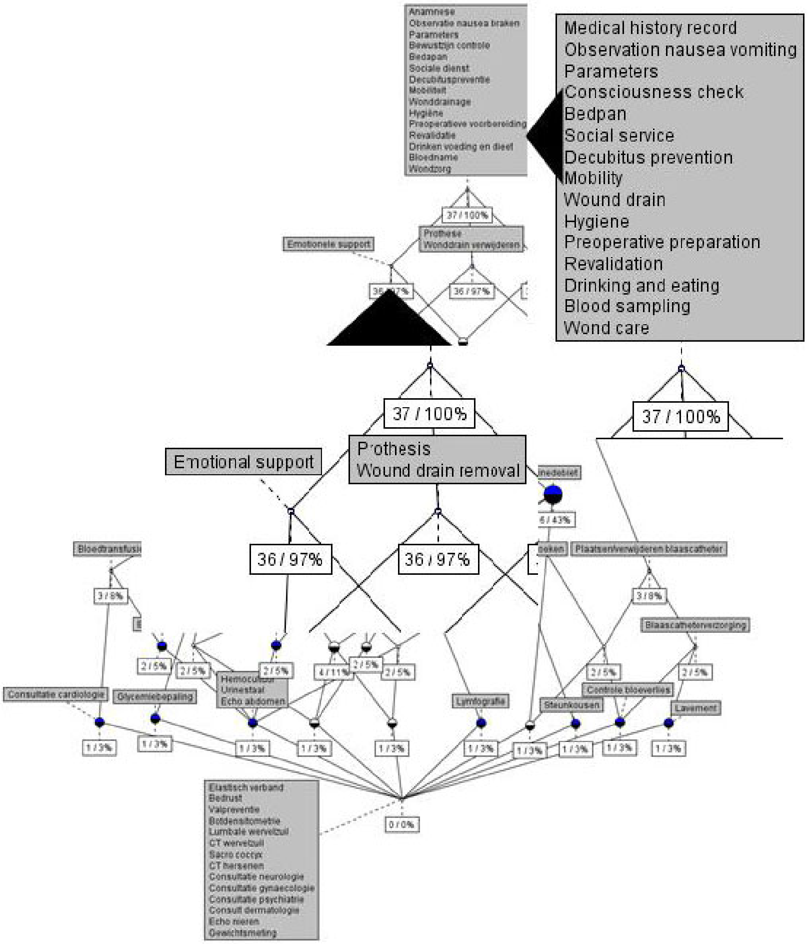


Fig. 5. Lattice containing 37 patients receiving mastectomy surgery with lymph node removal

lattice for the 60 patients receiving breast conserving surgery with attributes length of stay and doctor performing the operation.

We saw for the breast conserving surgery with lymph node removal that 25 patients with a length of stay smaller than 4 days were treated by "surgeon 9", whereas almost all patients treated by the other doctors had a longer length of stay.

We extracted these subsets of patients and constructed a process model for the groups of patients with a length of stay smaller than 4 days, equal to four days and larger than 4 days. This way, we were able to extract some best practices that could be used to improve the care provided to all patients. Fig. 7 contains the HMM process model extracted from the datasets with the 10 breast-conserving surgery patients with a length of stay in the hospital of 4 days (the average length of stay). This process model was chosen because of its simplicity in comparison with the other models and since it most closely resembles the standard care process as perceived by the domain experts.

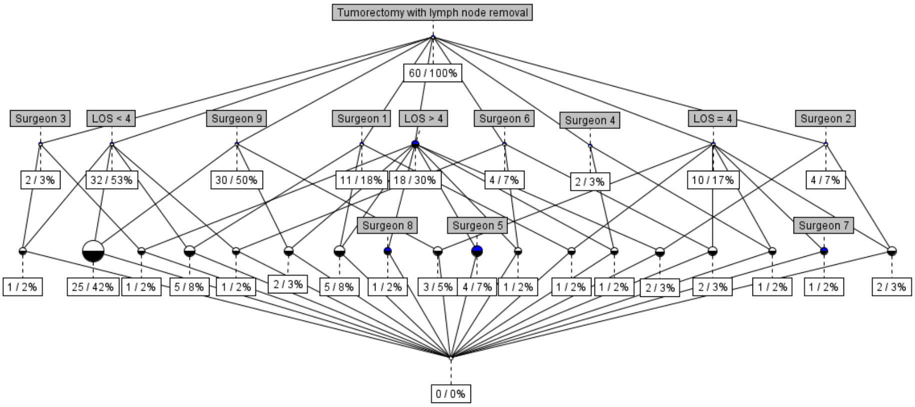


Fig. 6. lattice for 60 patients receiving breast conserving surgery

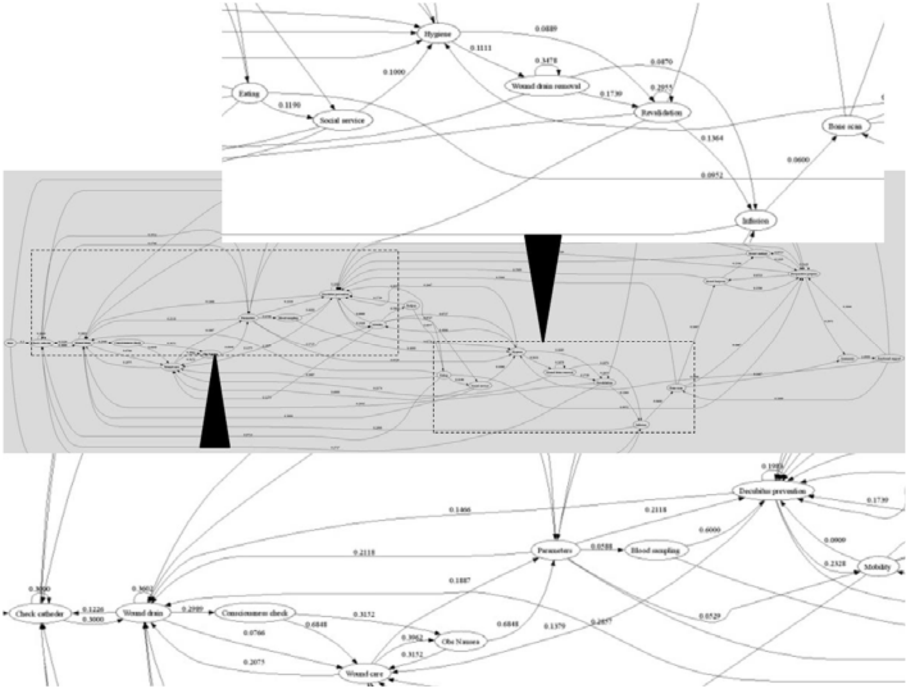


Fig. 7. Process model for 10 breast-conserving surgery patients with length of stay of 4 days

Table 1 contains some of the complexity measures for these process variations. For each surgery type and length of stay, the number of patients, the average number of activities and the number of unique activities performed to these patients is given. For visualizing the process maps, we laid a cutoff point at 5%, i.e. all transitions with a lower probability of occurrence were removed from the process representation. The table also contains the number of remaining unique activities and the number of connections after filtering. The structural complexity measure after filtering is the sum of these two measures.

Table 1. Complexity measures for the two process variations with the largest number of patients

SURGERY	\ LOS	LOW	AVG	HIGH
Breast Conserving Therapy with Lymph Node Removal	Length of stay	< 4 days	= 4 days	> 4 days
	# patients	32	10	18
	Avg. # activities	97	146	184
	# unique activities	32	23	35
	# unique act filtered	24	22	22
	# connections filtered	98	80	92
	Struct. Complex. filtered	122	102	114
Mastectomy with Lymph Node Removal	Length of stay	< 7 days	= 7 days	> 7 days
	# patients	17	4	16
	Avg. # activities	187	206	268
	# unique activities	27	21	36
	# unique act filtered	19	20	24
	# connections filtered	83	78	100
	Struct. Complex. filtered	102	98	124

5.4 Data Entrance Quality Issues

Using the process models, we also found some data entrance quality problems. For some patients, activities were registered after the day of discharge. We found that this was due to an error in the computer program combined with sloppy data entry by the nursing staff. We also found many semantically identical activities that had different activity numbers.

When we analyzed the process models, we found that some of the events typically were not ordered in the sequence that they are performed in real life. In other words, the timing of the events as can be found in the data does not always correspond to the timing at the real-life working floor. We found this is due to an error in the computer system which sometimes imposes a certain sequence of events and does not allow for a correct registration of activities.

There is a discrepancy between this built-in top-down developed model and the reality. This discrepancy is probably due to the insufficient insight into the reality of the working floor when the system was developed. The anomalies found during this process mining exercise will be used as input for the development of the new IT systems.

6 Discussion and Conclusions

Neither process nor data discovery techniques alone are sufficient for discovering knowledge gaps in particular domains such as healthcare. In this paper, we showed that the combination of both gives significant synergistic results. Whereas FCA does not provide easy to use process representations, it has proven to be very useful for process analysis, i.e. to analyze anomalies and exceptions in detail.

Initially, we thought FCA would only be useful for post-factum analysis of the results obtained through process discovery, but in this case we also found that FCA can play a significant role in the discovery process itself. In particular, concept lattices were used to improve the detection and understanding of outliers in the data. These exceptions are not noise, but are the activities performed to human beings, so every exception counts and must be understood. Concept lattices were also used to reduce the workspace, to cluster related events together in an objective manner.

Using this combination of techniques, we exposed multiple quality of care issues. We gained a better understanding of the process variations and better understood where we should take action to improve our healthcare processes. The impact of comorbidities of patients on the overall care process was found to be of importance and offers some opportunities for improving quality and efficiency of care. Further, reducing the length of stay of breast-conserving therapy patients was discovered to be the root cause for a suboptimal care, missing some key interventions, provided to patients. Finally, we found the length of stay for patients receiving breast-conserving surgery was significantly different for different surgeons. This situation may be improved by uniformization of discharge criteria.

Avenues for future research include the use of supervised clustering, mainly to obtain normalized process models, in which many-to-many transitions are eliminated (as argued in [10]). The normalized clusters will give the best views on process variations. Again, a posterior data discovery (FCA) can be used to understand the meaning of the different clusters.

Acknowledgements

We would like to express our special thanks to Chris Houck (OpenConnect) for his help in the construction of the process models. Jonas Poelmans is aspirant of the "Research Foundation - Flanders" or "Fonds voor Wetenschappelijk Onderzoek - Vlaanderen". Ed Peters is special guest professor for the K.U.Leuven OPENCONNECT Research Chair on Process Discovery.

References

1. Anyanwu, K., Sheth, A., Cardoso, J., Miller, J., Kochut, K.: Healthcare enterprise process development and integration. Journal of research and practice in information technology 35(2), 83–98 (2003)
2. Blum, T., Padoy, N., Feussner, H., Navab, N.: Workflow mining for visualization and analysis of surgery. International journal of computer assisted radiology and surgery 3 Suppl. 1, (June 2008)

3. Campbell, H., Hotchkiss, R., Bradshaw, N., Porteous, M.: Integrated care pathways. British Medical Journal 316, 133–137 (1998)
4. Ferreira, D., Zacarias, M., Malheiros, M., Ferreira, P.: Approaching process mining with sequence clustering: experiments and findings. In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 360–374. Springer, Heidelberg (2007)
5. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, Heidelberg (1999)
6. Mans, R.S., Schonenberg, M.H., Song, M., Aalst, W.M.P., Bakker, P.J.M.: Application of process Mining in health care - a case study in a Dutch hospital. In: Biomedical engineering systems and technologies, International Joint conference, BIOSTEC 2008, Funchal, Madeira, Portugal, January 28-31, Springer, Heidelberg (2008)
7. Maruster, L., Van der Aalst, W.M.P., Weijters, A.J.M.M., Van der Bosch, A., Daelemans, W.: Automated discovery of workflow models from hospital data. In: Proceedings of the ECAI workshop on knowledge discovery and spatial data, pp. 183–190 (2002)
8. Maruster, L., Weijters, A.J.M.M., Van der Aalst, W.M.P., Van den Bosch, A.: A rule-based approach for process discovery dealing with noise and imbalance in process logs. Data mining and knowledge discovery 13 (2006)
9. Murphy, K.: Hidden Markov Model (HMM) MATLAB toolbox (1998), http://people.cs.ubc.ca/~murphyk/Software/HMM/hmm.html
10. Peters, E.M., Dedene, G., Houck, C.: Business Process Discovery and Workflow Intelligence Techniques with healthcare applications. In: Proceedings of the INFORMS ORAHS 2009 Conference, Leuven (2009), http://www.econ.kuleuven.be/eng/tew/academic/prodbel/ORAHS2009/page5.htm
11. Poelmans, J., Elzinga, P., Viaene, S., Dedene, G.: A case of using formal concept analysis in combination with emergent self organizing maps for detecting domestic violence. In: Perner, P. (ed.) Advances in Data Mining. Applications and Theoretical Aspects. LNCS, vol. 5633, pp. 247–260. Springer, Heidelberg (2009)
12. Quaglini, S.: Process Mining in Healthcare: A Contribution to Change the Culture of Blame. In: Book Business Process Management Workshops. LNBIP, vol. 17, pp. 308–311. Springer, Heidelberg (2009)
13. Rabiner, L.R.: A tutorial on Hidden Markov Models and selected applications in speech recognition. Proceedings IEEE 77(2), 257–286 (1989)
14. Skinner, J., Chandra, A., Goodman, D., Elliot, S.F.: The elusive connection between health care spending and quality. Health Affairs - Web Exclusive w119-w123 (2008)

Interest-Determining Web Browser

Khaled Bashir Shaban[1], Joannes Chan[2], and Raymond Szeto[2]

[1] Department of Computer Science and Engineering, College of Engineering,
Qatar University, P.O. Box: 2713, Doha, Qatar
khaled.shaban@qu.edu.qa
[2] Department of Systems Design Engineering, Faculty of Engineering,
University of Waterloo, 200 University Avenue West, Ontario, N2L 3G1, Canada

Abstract. This paper investigates the application of data-mining techniques on a user's browsing history for the purpose of determining the user's interests. More specifically, a system is outlined that attempts to determine certain keywords that a user may or may not be interested in. This is done by first applying a term-frequency/inverse-document frequency filter to extract keywords from webpages in the user's history, after which a Self-Organizing Map (SOM) neural network is utilized to determine if these keywords are of interest to the user. Such a system could enable web-browsers to highlight areas of web pages that may be of higher interest to the user. It is found that while the system is indeed successful in identifying many keywords of user-interest, it also mis-classifies many uninteresting words boasting only a 62% accuracy rate.

Keywords: Web Mining, Machine Learning, Self-Organizing Map.

1 Introduction

According to a study conducted by Statistics Canada in 2007 [1], over two-thirds of all Canadian Internet home users log on daily, with half of these individuals reporting 5 or more hours of usage in a typical week. Furthermore, it was reported that behind checking one's email, general web-browsing was the second most popular task performed when surfing the web. With such a high frequency of usage, the act of web browsing inherently reveals much about the character of the user. Similar to how a person's preferences in literature, movies, and music can all provide clues as to their likes and dislikes, webpage preferences can also be a window into personal taste. Unlike the other aforementioned mediums however, the world-wide web presents a unique ability for a person's preferences to be analyzed and dissected as all web-traffic is ultimately recorded on a day-to-day basis by the user's browser.

The goal of this work is to design a system that, by analyzing the user's browsing history, could potentially endow a browser application with the ability to discern what the likes and dislikes of the user are. Once available, such information could be utilized for a variety of different applications. For example, the web browser could then highlight areas of new incoming web-pages that may be of interest to the user, or actively crawl the internet for interesting articles and suggest these to the user. Furthermore, since a user's web-browsing history continually builds over use, such a system should be able to adapt to the user's preferences over time as well.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 518–528, 2010.

The application of similar concepts can readily be seen in the usage of contextual advertisements on the web [8]. Such systems, like Google's Ad-Sense, parse the text of a webpage or email in order to determine what type of relevant advertisement should be placed on a particular page. The system proposed in this paper differs from techniques such as those in Google Ad-Sense as it attempts to collectively analyze all of the web pages a user visits and cross reference this data. For example, should a user visit a video website regularly to download/watch a particular television show, the system should highlight a headline pertaining to such a show the next time the user is on an entertainment news website. Stated another way, the proposed system is client-side based and thus privy to much more usage information, whereas the information available to Google's Ad-Sense is limited due to the fact that it is primarily server-side. Other server-side implementations include the Web-Lattice approach, which intelligently sorts server-side browsing data in order to determine an interest hierarchy [9].

Of higher relevance to the task at hand is a research project that was carried out at the Jozef Stefan Institute in Slovenia which attempted to reorganize browsing history based on subjects of interest [3]. In their article, Grcar et al., discuss how words of interest can be determined by extracting all of the content text from the HTML source of a webpage, converting this text into word vectors, and performing k-means clustering on these vectors. While this solution is somewhat adequate, it does not analyze usage statistics of the user but instead only focuses on the nature of word distribution.

The main challenge in designing such a system involves accurately separating the true “likes” of the user from what the user just happens to see repeatedly. For example, the system should realize that the appearance of a copyright line on the bottom of many web pages does not necessarily indicate that the user has an interest in copyright law. For the purposes of a prototype, it was decided that the problem would be scaled down to tackle one specific application. More specifically, a system was developed that would parse and analyze a simulated set of browser history in order to determine the user's interests. The system was then presented with a new web page separate from which it attempted to extract a subset of words which it believed to be "liked" by the user (which could then be used in future implementations for highlighting). It was decided that in order to tackle such a problem, a neural network would be utilized.

2 Approach

The problem of determining user interests based on web history can essentially be broken down into two separate subtasks. The first of these tasks requires that keywords be extracted from the web pages in the browser history. In this particular context, the term "keywords" refers to the words in an HTML file that are representative of the content on that particular webpage. For example, keywords for a webpage about the Beatles may include "Beatles", "music", "John Lennon", "guitar" and so forth. Once these words are determined, a set of usage-data about each word is recorded. The second phase of the problem then involves using this usage-data to determine the subset of words that are of interest to the user. More specific details on these to subtasks are in the follow subsections.

2.1 Determining Keywords

In order to generate the dataset necessary, the user's browsing history is first parsed in order to collect a list of web addresses that the user has visited in the past. For each URL in the history, the system downloads the HTML source for the page and discards all HTML tags and any scripting code (e.g. JavaScript) leaving only the text of the actual content behind. With the content text successfully extracted, the system performs a tally on each of the content words (within a single page) and saves this tally into a record. In this way, a record containing all of the word tallies on a web page is created for each URL. Once all of the URLs in the history are successfully visited, the system iterates through the records and extracts a subset of "keywords" using term-frequency/inverse-document frequency filtering (TF/IDF). This method, commonly used in data mining applications, assigns a weight to every word within every record by using the following equation [6]:

Weight = $(f_l / f_{\max}) * log(N/N_l)$, where
f_l is the frequency of the word within the local webpage
$f_{\max}$ is maximum frequency of any word on the same page
N is the total number of web pages in the history
N_l is the number of pages containing the word

Note the following characteristics of this equation:

- The weighting is proportional to the frequency of the word within the local webpage. That is, if the word appears frequently on a single page, it is considered to be more important.
- The frequency of the word on the local webpage is divided by the maximum frequency of any word on the same page as a means to normalize this value.
- The weighting decreases if the word appears over many pages in the history. That is, if the word is extremely common over all pages, then it is considered to be less important. This property is useful for filtering words such as articles (i.e. "a", "an", "the" etc.), prepositions ("on", "over", "in" etc.) and pronouns ("he", "she" etc.).

Once the weighting is calculated, the system makes a decision to keep or reject the candidate as a keyword by comparing the weighting to a threshold value. If the weight is greater than the threshold, the keyword is entered into an entirely new record system along with usage-data pertaining to the keyword. This usage-data is obtained by looking at the usage information of the keyword's associated web page. More specifically, after this filtering process the following attributes are stored along with each keyword:

1. Total number of web pages that contain the word as a keyword
2. Days since the page with the word was last accessed
3. Total minutes spent on the page with the word
4. Total number of web pages that contain the word (regardless of whether it is a keyword or not)

These inputs were partially decided based on past research work on user interest profiling [7]. This process is repeated for every single word over every single webpage in the history.

2.2 Determining User Interests

With usage data for all of the keywords in the browsing history now available, the second part of the problem involves feeding this data into a neural network in order to determine those words that are of actual interest to the user. Neural networks are suitable approach since they are inherently adaptive, and the ability to adapt is an asset since user-browsing patterns are likely to change variably over time. For this specific application, an unsupervised neural network is utilized since a supervised neural network would require the user to give constant feedback as to whether certain words were in fact liked. Such a task would be a significant nuisance to the user and thus extremely impractical in a real application. In contrast, an unsupervised neural network is capable of taking the usage data, and determining a pattern based on clustering in order to decide whether a keyword is of interest without any need for user input.

3 Implementation

For the purposes of a working prototype, the proposed system was simulated with the use of the Mozilla Firefox web browser, a Python script, and the Matlab neural network toolbox. The Python script was primarily responsible for parsing the browser history and determining content keywords, while Matlab was required in building neural net used to determine which of the keywords were of actual interest to the user. Finer implementation details are discussed in the following subsections.

3.1 Determining Keywords

In order to generate the required data, all Firefox history was cleared and a series of webpages were chosen and visited based on a certain user profile. In this particular instance, websites pertaining to jazz music were chosen in order to simulate a jazz music lover who often scouts for concerts and visit pages with musician information. Once completed, the Firefox history was extracted by looking at Firefox's history.dat file. Since this data is stored in a proprietary file format known as MORK, the MORK file was first converted into a space-delimited text file with the use of a free utility named DORK. Each row of a DORK-processed text file contains a single URL, along with several attributes including the number of repeated visits, the first visit date, and the last visit date. Since the Firefox session took place within the same day, the dates in the DORK-processed text file were then slightly modified manually in order to simulate a greater span of time. Furthermore, an extra column was added to represent "total minutes spent on the page". This was primarily motivated by the fact that past research has shown that pages viewed for longer periods of time tend to contain information of higher interest to the user [5]. Thus, if such a system were actually incorporated into Firefox, this additional attribute could be added to the history file via a Firefox plugin or extension.

Once the DORK-processed text file was made available, a Python script was written to visit each URL and extract the content text from the HTML source. This was done with the help of the Python SGMLParser library, which contains pre-written routines to help identify the different components within an HTML source file. With the content text extracted, each unique word was tallied and the results were used to calculate the term-frequency/inverse-document frequency weight of the word. Should the weighting exceed a threshold of 0.3, it was considered as a "keyword" of the page. The end result was a dictionary (a data structure similar to a Perl hash) with all of the unique words in the history as keys, and the desired attributes (listed in section 2.1) as values. For completeness, the total frequency of the word over all pages was also saved as an optional fifth attribute, but ultimately not used as an input to the neural network. Furthermore, for the purposes of this prototype, the data set was also cut down (to speed up the neural network) by eliminating any words which never appeared as keywords on any page. This data structure was then printed to a space-delimited text file, which was subsequently converted to an excel spreadsheet to facilitate easier importing into Matlab. This file represented the training data for the neural network.

With the training data determined, a simple testing set was also created by going to the Toronto Star music web page, and running the HTML source through both the SGMLParser to obtain the content, and then the term frequency/inverse-document frequency to obtain the keywords on the page. These keywords were then queried against the training set, and if they existed in the training set the attributes for each word were printed to a text file. Since the training set only contained words that appeared as keywords in at least one page in the history, this step also effectively filtered out all words that were not keywords in the past. It should also be noted that in essence, the testing data set is actually a subset of the training data set. However this is inherently necessary since the system should not update the existing data with new information from the current testing page. For example, the "last accessed" date should not be changed to the current time just because the user is accessing the test page now. Once determined, the testing set was also converted into an excel spreadsheet for use in Matlab.

3.2 Determining User Interests

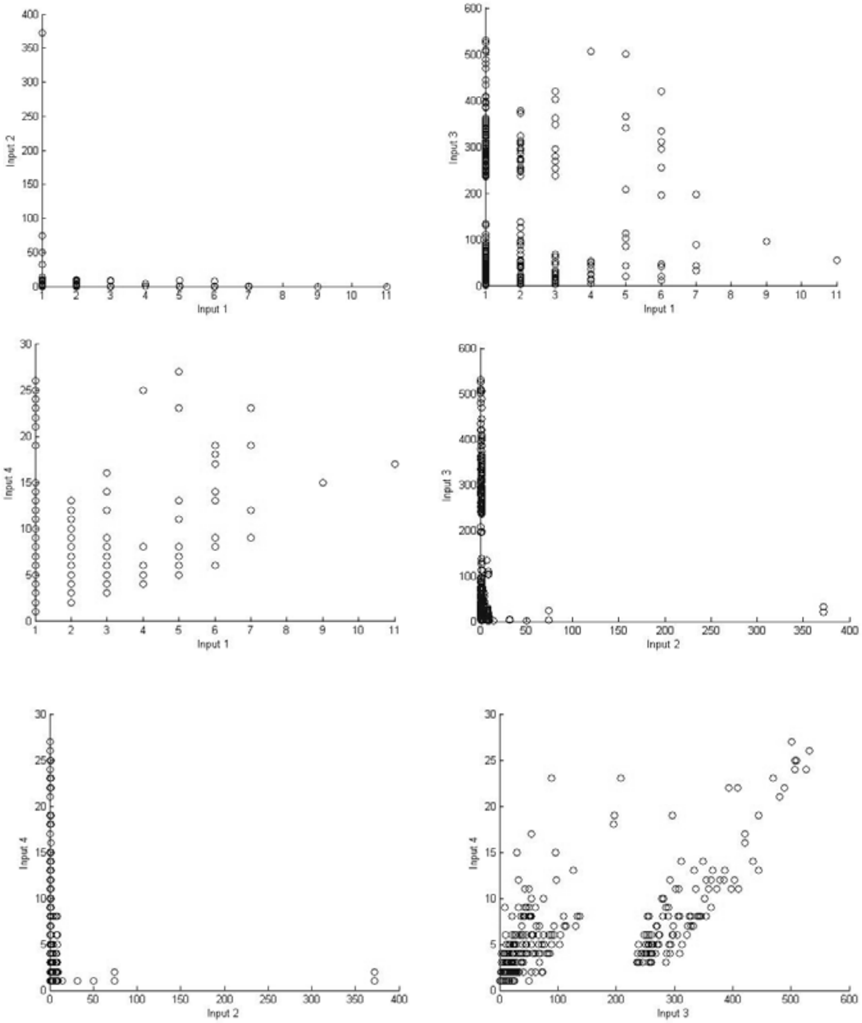
Since the system was expected to learn continuously over time, an unsupervised learning neural network was deployed in Matlab. To reiterate, for any given keyword the system was required to identify patterns utilizing the 4 inputs defined as follows:

1. Total number of web pages that have this word as a keyword
2. Days since the page with the word was last accessed
3. Total minutes spent on the page with the word
4. Total number of web pages that have this word (regardless of whether it is a keyword or not)

The actual inputs utilized for testing can be found by looking at Appendix A. The solution was implemented in the form of a Self-Organizing Map (SOM) neural network. It should be noted however, that an SOM takes only 2-dimensional data points to compute a neuron map. While there existed 4 data points for input, it was quickly realized through trial and error with various neural net configurations that the inputs did not form adequate clusters in n-4 space. Thus, the best 2 out of the 4 inputs were selected. In order to determine which 2 inputs were the most appropriate, the mechanics of the SOM first need be discussed.

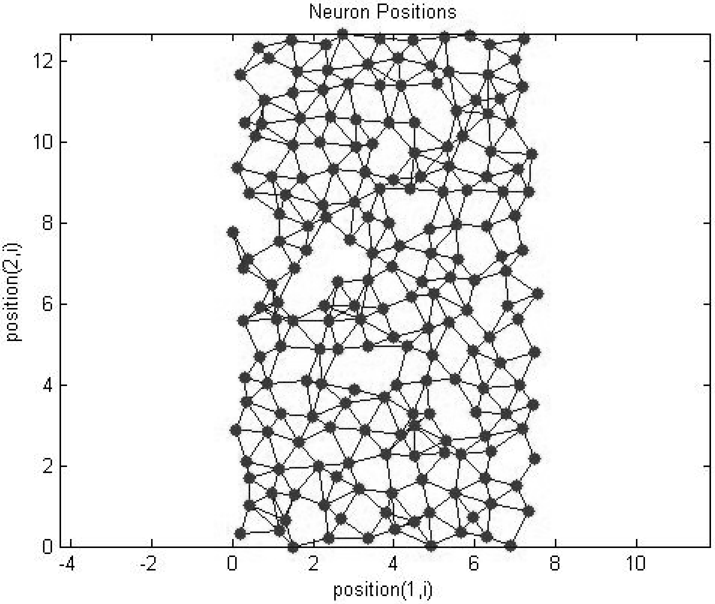
Given the spread and spatial organization of a set of data points, an SOM determines different classes in a manner similar to K-means clustering, a pattern recognition approach that separates data based on their Euclidian distances. Thus, when graphed against each other, an ideal set of inputs should present themselves with some distinct clustering patterns in Euclidian space that the SOM can recognize. Holland's paper [4] stresses the statistical importance of data to be used, which seconds this assertion.

In order to indentify the ideal input pair, the different pairs of input data for all 300 keywords in the training set were plotted against each other. The results were as follows:



The results clearly show that the patterns of input 1 vs. 2, input 1 vs. 4, input 2 vs. 3 and input 2 vs. 4 have poor clustering patterns for a two class case. In contrast, the plot of input 3 vs. 4 clusters much better into two classes. Thus, these two sets of data points were chosen to be the inputs fed into the SOM. It should be noted once more that while input 1 was not chosen to be inputted into the neural network, the information gleamed from input 1 was already used once to cut the data set down to only approximately 300 inputs. That is, the data set fed into the neural network contained only words that appeared as keywords on at least one page.

With the inputs determined the SOM was generated with the following parameters:
Neuron grid: 10 x 20, randomized shape
Epoch: 3000
Training Data Size: 300
The resultant SOM generated was graphed as follows:



When the network was simulated, the SOM determined the neuron closest to each of the input data. The closet neuron activated, and the input data can be understood as associated with the activated neuron. The neurons were numbered and ordered, with neuron 1 located at the bottom left of the map and neuron 200 located at the top right of the map.

4 Evaluation

The neural network was tested against 50 data points that were extracted from another web page (the Toronto Star music homepage). Higher neuron numbers refer to a higher chance of the word being a word of interest, thereby denoting the relative importance of the word. To evaluate the results based on the binary selection requirement, it can be understood that with an activation neuron number above 100, the word tends to be of interest, while a number below 100 means the word is not of interest. Furthermore, in order to test the accuracy of the results each word was qualitatively assigned as being truly “of interest” or ”not of interest” based on the simulated user profile. The 50 data points were evaluated based on this understanding with the results as follows:

Word	Neuron #	Correct?
Advanced	196	No
Advertising	101	No
Air	144	No
American	36	Yes
April	196	No
Archive	76	Yes
Back	144	No
Band	115	Yes
Barber	36	Yes
Beat	120	Yes
Blogs	81	Yes
Books	18	Yes
Can	196	No
Canadian	125	Yes
CD	196	Yes
Charles	195	Yes
David	138	Yes
DVD	177	No
Events	114	Yes
Full	126	No
Function	142	No
Having	144	No
He	196	No
Hot	20	Yes
Ian	125	Yes
Jazz	196	Yes
Julian	70	Yes
Kennedy	20	Yes
Last	147	No
Life	189	No
Live	186	Yes
Mike	127	Yes
Music	196	Yes
Musician	174	Yes
Must	175	No
News	92	Yes
Note	61	Yes
Out	196	No
Phoenix	69	Yes

Photos	83	Yes
Playing	196	Yes
Recent	198	No
Releases	133	Yes
S	81	Yes
Sensitive	9	Yes
Show	137	Yes
Singers	36	No
Site	113	No
Specials	10	Yes
Star	199	No

The items highlighted in grey are important words that are definitely of user interest.

Results:

Correct: 31 / 50
Incorrect: 19 / 50
Accuracy rate: 62%
Keywords identified: 16 / 17
Keyword identification accuracy rate: 94%

The overall test results do not demonstrate very accurate results, with an accuracy of 62%. Major keywords of interest however, were in fact identified correctly. Recalling that the user profile was preset to be a jazz music lover who often scouts for concerts and visit pages with musician information, important keywords such as “band”, “events” and “jazz” were correctly identified. Furthermore, with 17 pre-determined words of interest, the system was able to properly identify 16 of them. The problem however, was that the system also identified many uninteresting words as words of interest. In other words, the system tends to be too permissive in identifying the words of interest.

5 Discussion

One of the sources of error within the current system stems from the fact that while the SGMLParser class is extremely easy to use, it is not the most robust. More specifically, it was found early on that the SGMLParser class would actually misidentify snippets of Javascript and HTML as content text. In order to correct for this, a filter was implemented whereby only alphanumeric characters would pass. This greatly decreased the errors, however a few words still slipped through, as is evidenced by the abundance of the word "function" in the testing set. To correct for this in the future, the open source SGMLParser would need to be investigated and modified appropriately.

Another source of error came from the fact that the idealized threshold value for the term-frequency/inverse-document frequency was actually slightly different for each webpage. An attempt was made to normalize these values, as evidenced by the fact that in calculating the weight the equation divides by the maximum frequency for any word for any given page. However, while this normalization did help, the threshold values still seem to be somewhat disparate from each other. More work needs to be done in order to come up with a proper normalizing function over all pages.

Furthermore, while the term frequency/inverse document frequency technique is adequate and relatively inexpensive computationally, it is not entirely the most accurate. Since whether or not a word should be considered a "keyword" is rather subjective, no quantitative measure can be given as to the accuracy of the TF/IDF technique for this particular application. However, upon perusal of the output, it is fairly obvious that some words do not belong. Since usage data on these misidentified keywords is fed as the input into the neural network, they obviously introduce error into the neural network as well. Thus, any future system should investigate alternative methods to determining keywords. One such approach may be to forgo the step completely, and instead let the neural network do the work instead. Another approach would be to first extract all the nouns in a given web page since nouns are more likely to be keywords. This can be done by interfacing the system with a dictionary database, or by using a probabilistic approach to detect nouns [2].

In addition to flawed input data, the usage of an unsupervised network also limits the intelligence of the system. The core concept motivating an unsupervised network is that it is able to recognize a distinct pattern given the data. However, the inputs to the network may very well prove to not have a distinct pattern as indicated by the difficulty in creating a working neural-net using all four data points in n-4 space. Thus, it becomes questionable whether the user’s interest is determined solely by the patterns in the numerical factors presented in the user’s browsing history. That is, it might be difficult for the system to learn without a teacher. On the other hand, as mentioned previously the system should not expect a teacher as it becomes unrealistic for a user browsing the web to constantly train the system. It may be more realistic to develop a system that can take user-feedback in a way that is intermittent and less intrusive.

6 Conclusion

There are many uncertainties present in determining user interest based on past browsing history. Thus, it is not surprising that there does not exist a single algorithm or solution that can compute user interest based on the patterns found in browsing history. Nonetheless, the implemented system is able to reduce the giant list of potential keywords to a small list, and indicate the words of interest with decent accuracy. However, the system is also overly permissive in marking uninteresting words as words of interest to the user. To improve on the robustness and sensitivity of the system, a more complex neural network approach with some redundancy might be helpful. Furthermore, incorporating a supervised learning system that is unobtrusive to the user may also be beneficial.

References

1. Statistics Canada, Canadian Internet Use Survey (June 12, 2008), `http://www.statcan.gc.ca/daily-quotidien/080612/dq080612b-eng.htm`
2. Chen, K.-h., Chen, H.-H.: Extracting Noun Phrases from Large-Scale Texts: A Hybrid Approach and Its Automatic Evaluation. In: Proceedings of the 32nd annual meeting on Association for Computational Linguistics, pp. 234–241 (1994)

3. Grcar, M., Mladenic, D., Grobelnik, M.: User Profiling for Interest-Focused Browsing History. In: Proceedings of the Workshop on End User Aspects of the Semantic Web (in conjunction with the 2nd European Semantic Web Conference), May 29-June 1, pp. 99–109 (2005)
4. Holland, S., Ester, M., Kiebling, W.: Preference Mining: A Novel Approach on Mining User Preferences for Personalized Applications. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 204–216. Springer, Heidelberg (2003)
5. Liang, T.-P., Lai, H.-J.: Discovering User Interests From Web Browsing Behavior: An Application to Internet News Services. In: Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS 2002), vol. 7, p. 203 (2002)
6. Ramos, J.: Using TF-IDF to Determine Word Relevance in Document Queries. In: Proceeding of the 1st International Conference on Machine Learning (2003)
7. Stermsek, G., Strembeck, M., Neumann, G.: User Profile Refinement Using Explicit User Interest Modeling. In: Proceedings of 37th Jahrestagung der Gesellschaft für Informatik, GI (2007)
8. Yih, W.-t., Goodman, J., Carvalho, V.R.: Finding Advertising Keywords on Web Pages. In: Proceedings of the 15th international conference on World Wide Web, pp. 213–222 (2006)
9. Zhou, B., Hui, S.C., Fong, A.C.M.: A Web Usage Lattice Based Mining Approach for Intelligent Web Personalization. International Journal of Web Information Systems 1(3), 137–146 (2005)

Web-Site Boundary Detection

Ayesh Alshukri, Frans Coenen, and Michele Zito

Dept. of Computer Science, The University of Liverpool, Liverpool L69 3BX, UK
{a.alshukri,coenen,michele}@liverpool.ac.uk

Abstract. Defining the boundaries of a web-site, for (say) archiving or information retrieval purposes, is an important but complicated task. In this paper a web-page clustering approach to boundary detection is suggested. The principal issue is feature selection, hampered by the observation that there is no clear understanding of what a web-site is. This paper proposes a definition of a web-site, founded on the principle of *user intention*, directed at the boundary detection problem; and then reports on a sequence of experiments, using a number of clustering techniques, and a wide range of features and combinations of features to identify web-site boundaries. The preliminary results reported seem to indicate that, in general, a combination of features produces the most appropriate result.

Keywords: Web-site definition, Web-page Clustering, Web Archiving.

1 Introduction

As the World Wide Web has grown in size and importance as a medium for information storage and interchange, the problem of managing the information within it has assumed great significance. In particular, there has been a lot of interest, recently, in working with whole web-sites, and other compound web-objects rather than single web-pages [5,17,18]. The detection of web-site boundaries is an important aspect with respect to many applications such as web archiving, WWW information retrieval and web spam detection. The process of archiving web content is a non trivial task [6, page 82]. The target information may be contained in just a few HTML files, or a very complex web application [1]. Identifying the boundary of a web-site can automate the choice of pages to archive. Studying the world-wide web at web-site level rather than web-page level may also have useful applications [3].Documents can be represented by multiple pages on the web [5]. Thus, sometimes, it is not reasonable to study attributes like authorship at page level. A web-site entity may be reorganised at the site owners control, as pages and links appear/disappear on an infinite basis [14]. This characteristic implies the separate study of inter and intra site links. The accessibility of content on the web [4], assuming content is fully accessible from within a site (navigation between pages all of the site) can focus on connectivity between sites. Finally, the study of the web using statistical analysis of web-pages maybe skewed due to the simplicity of rapid and dynamic generation.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 529–543, 2010.

The identification of the boundaries of a web-site can be a relatively simple task for a human to achieve. When traversing the web, navigating from one web-page to another, the detection of a particular web-sites boundaries is done by a human recognising certain attributes from these pages or closely related content. The set of attributes from a page is usually common to each of the pages within a web-site, this is also true of the topics which are closely related or about one theme. The features that a user can recognise to determine similarity between pages can be the style and layout of the page, including; colours, borders, fonts, images and the positioning of these items. Also the content covered, including topic or topics displayed in various sections or sub section within the same page or spread across pages. Although this is all fairly obvious to humans, the boundary detection task is far from trivial for a machine. This paper tries to overcome such difficulties by proposing a data mining approach to the web-site boundary identification problem.

The identification process is hampered by the lack of a clear, general, and useful definition of what a web-site is [2,3]. The term is often used either informally (for instance when investigating the sociological impact of the web [10]), or in rather specific ways. The simplest option is to state that a web-site is defined by the machine it is hosted on. However, several web-sites may be hosted on the same machine (e.g. http://www.member.webspace.virginmedia.com has content by many authors), alternatively a single web-site may span several machines (for example the INRIA's web-site has content on domains www.inria.fr, www-rocq.inria.fr, osage.inria.fr, etc). A web-site may also comprise several sub web-sites. To apply data mining techniques to the web-site boundary detection problem, in the context of applications such as web archiving, requires some definition of a web-site. This is one of the issues addressed in this paper. The second issue has to do with the nature of the web-page features that should be included in a feature vector representation that permits the application of data mining techniques to identify web-site boundaries. From the above it is clear that URL alone is not sufficient. Intuitively content alone would also not be sufficient given that any web-site can be expected to link to other sites with similar content. In this paper we present a number of experiments investigating which features are the most appropriate to aid the identification of web-site boundaries.

Given a collection of web-pages, represented in terms of a set of features, we can attempt to identify boundaries either by processing the collection in a static manner or a dynamic manner (by "crawling" through it). The first option is considered in this paper. In the static context clustering techniques may be applied so as to distinguish between web-pages that belong to a given web-site and web-pages that do not belong to the site.

The contributions of this paper may thus be summarised as follows;

1. A definition of what constitutes a web-site in the context of web-site boundary identification.
2. A report on a sequence of preliminary experiments, conducted using a number of different web-page features, and a combination of features, to determine the most appropriate features for boundary identification.

3. A report on the use of a number of different clustering techniques to identify the most appropriate for web-site boundary identification.

Note that the most appropriate clustering technique and web-page model combination, as will be demonstrated, is that which most accurately generates the known clusters present in a number of "test" input data sets.

The rest of this paper is organised as follows. In Section 2 we present our definition of what a web-site is, and compares this to previous proposals. Section 3 then presents a discussion of the web-site boundary identification process, and discussion of the potential features that may be most appropriately used to identify such boundaries. The results of the evaluation of the different potential features, using a number of clustering techniques, is presented in Section 4. some conclusions are presented in Section 5.

2 Web-Site Definition

A number of proposals have been put forward over the years, to characterize the idea of a collection of strongly related web-pages. In the work by Senellart [19,20], the aim is to find web-pages that are contained in logically related groups using the link structure. Senellart emphasises the fact that there is no clear definition of what a web-site is, and defines a "logical" web-site as a collection of nodes that are significantly more connected than other "nodes". This definition abstracts from the physical notions described in the traditional definition (single server, single site) and makes a more subjective claim that concentrates on the similarity between pages.

Work has also been done in the area of detecting web *subsites* by, for example, Rodrigues et al. [16,17] and Neilsen [15]. The authors use the word subsite to refer to a collection of pages, contained within a main web-site, that fills the criteria of having a home page, and having distinct navigation and styling from the main pages of the site.

Research by Dmitriev [5,7] brings about the notion of *compound documents.* This is a set of web-pages that aggregate to a single coherent information entity. An example of a compound document is a news article that will be displayed over several pages within the news web-site, each with a unique URL. The authors intention is for the reader to absorb the article as a single piece of information [7]. Some points to note about compound documents is that they have an entry point (which can be non trivial to find) which is similar to the definition of a subsite above. Using the definition it challenges the synonymous notion of web node equals web-page.

It is suggested, in the context of boundary detection, that an appropriate definition must encompass several of the above concepts. The following definition is therefore proposed:

Definition 1. *A* web-site *is a collection of web-pages that:*

WS1 *have a common entry point, referred to as the web-site* home-page*, such that every page in the collection is reachable from this home-page through a sequence of directed hyperlinks;*
WS2 *have distinct navigation or styling features, and*
WS3 *have a focused content and intention.*

The first two elements in the statement above are syntactic in nature. They refer to clearly recognizable features of the given collection of web-pages. The third one is intended to capture the purposes of the creators of the given collection.

Considering the above definition in further detail it should be noted that the definition is couched in terms of the expected structure of a web-site, and that some of the elements of the definition build upon existing ideas found in the literature. Constraint **WS1** is probably the most obvious one, and its importance has been recognized previously (see for instance [12,13,15,21]). It is also natural to add a constraint like **WS2**; similar styling is a clear sign of authorship. Collections of web-pages that have the same styling tend to have been created by the same people. Minor differences may arise between pages in the same collection, however common themes will often be shared by all pages that are part of a single conceptual unit. Constraint **WS2** also refers to the possibility that many pages in the same site may have similar link patterns. The styling may be completely different, but the navigation of the pages may share some common links (for instance a back link to the web-site home-page). As to **WS3**, the idea of focused content and intention has never been explicitly included in a web-site definition, although it is implicitly present in other proposals (e.g. [2]). The idea reflects the situation where an author has control over a collection of pages so that the pages can thus be said to be related by the author's intention.

It is perhaps also important to stress that we move away from the popular graphical vision associated to the web (see e.g. [4]). Web-sites are collections of related web-pages, but their hyper-link structure is only one of the many possible features that one should consider when grouping related pages. It will become apparent that hyper-links (directed out-going links) from a page are important, but, for instance, "popular-pages" [11] (a notion derived from the analysis of in-going links) seem to be less relevant with respect to web-site boundary definition.

The above definition (in the context of boundary detection) offers a number of advantages:

Generality: This is more general than previous proposals. Constraint **WS1** clearly relates to the notion of *seed pages* that has been used in the past as a means of clustering content-related web-pages.**WS2** encompasses the approaches based on the study of the URL's and the link structure of the given set of pages.

Flexibility: The definition is flexible. It is argued that any sensible definition must contain a semantic element referring to the authors' intentions. Such an element cannot be defined prescriptively, and is application dependent. Adding such element to the definition (constraint **WS3**) makes it suitable to describe a wide range of boundary detection scenarios.

Effectiveness: The proposed definition is effective because, as will be demonstrated, it can be used to identify web-site boundaries using data mining techniques.

3 The Web-Site Boundary Identification Process and Feature Selection

We now turn to the description of the proposed approach to the problem of web-site boundary identification.

As noted above, the process of identifying web-site boundaries adopted in this paper is a static one (as opposed to a dynamic one). The process commences with a crawl whose aim is to collect a set of web-pages that will represent the domain of investigation in the subsequent boundary detection process. The start point for the crawl is the home page of the target site. The search then proceeds in a breadth-first fashion with a crawling that is not limited to URL domain or file size. Thus, for example, if an external link (e.g. `google.co.uk`) was found, it would be followed and included in the dataset. Once a sufficiently large collection of pages has been gathered, feature vectors are constructed, one for each page, and a clustering algorithm applied to distinguish the target site from the "noise" pages. To complete the description of our approach we need to specify what features (attributes) to include in the feature vector and what clustering technique is the most appropriate. A tentative answer to the latter is provided in Section 4.4, here we address the former.

The space of features that may be used to describe a given web-page is massive. The features selected in the study described here include: hyper links, image links, *Mailto* links, page anchor links, resource links, script links, title tags and URLs. We contend that web-pages grouped based on such features and arbitrary combinations therein can be considered part of the same web-site, based on the definition given in Section 2.

The hyper-link based features were constructed by extracting all of the hyperlinks from each of the pages. Each textual hyperlink, representing a pointer to another web-page was stored as a single string (the only processing that was done was that the text was cast into lower-case, to facilitate comparison). The values associated with each of the features in the hyper-link group was the number of potential occurrences (frequency count) of each identified hyper link. The theory behind the use of hyper-links is that pages that are related may share many of the same hyper-links. The shared links may be other pages in the same website (e.g. the web-site home page) or significant external pages (e.g. most pages from a Department with a University web-site may point to the main University portal).

The image links feature sub-vector was built by extracting all of the links to images ($< img >$) from each of the given pages in W. The image links were processed in a similar fashion to the hyper-links, as described above. Pages that link to the same images were deemed to be related; for example the same set of logos or navigation images.

Another feature sub-vector was constructed by extracting $Mailto$ links from the pages in W. The idea is that a group of related pages may contain a $Mailto$ link to a single email contact (for example the *web master*). The links are extracted from the HTML code using the same method as described above, but looking for the $Mailto$ tags.

The page anchor links sub-vector was constructed by extracting all of the page anchors from each of the pages. Page anchors are used to navigate to certain places on the same page, these can be helpful for a user and can very often have meaningful names. It is conjectured that if the same or related names are used on a set of pages it could imply related content. The Page Anchor Links group of attributes were extracted by parsing the HTML code as above and identifying the number of possible occurrences (the values for the individual attributes).

The Resource Links feature sub-vector was constructed by extracting all of the resource links from the given pages. This commonly includes CSS (Cascading Style Sheet) links. The motivation is that the styling of a page is often controlled by a common CSS which could imply that a collection of pages that use the same style sheet are related. In this case the feature space is built by extracting only the resource links from the HTML.

The script links sub-vector was constructed by extracting all of the script links from each of the pages in W. This commonly included Java script links. The observation here is that some functions that are used on web-pages can be written and used from a common script file; if pages have common script links then they could be related. This feature sub-vector was built by extracting this information.

It is conjectured that the titles used in a collection of web-pages belonging to a common web-site are a good indicator of relatedness. The title group of features was constructed by extracting the title from each of the given pages. The individual words in each title were then processed to produce a "bag of words" (a common representation used in text mining). Note that when the textual information was extracted from the *title* tag non-textual characters were removed, along with words contained in a standard "stop list". This produced a group of feature's comprising only what were deemed to be the most significant title words.

The textual content was extracted from each page in the dataset by using a html text parser/extractor (http://htmlparser.sourceforge.net/). This gave only the text that would be rendered by a web browser. This is deemed to be the same text that a user would use to judge a pages topic/subject. Stop words (same list as used to identify title text above) were then removed and a bag of words model produced as in the case if the title sub-vector.

Finally the URL feature sub-vector was constructed by collating the URL's from each of the pages. The motivation is that the URL is likely to be an important factor in established whether subsets of web-pages are related or not. As noted above URL should not be considered to be a unique identifier for every web-page in the given collection. The URL of each page was split into "words" using the delimiters found in URL's. For example the URL http://news.bbc.co.uk would produce the attributes news, bbc, co and uk. Non textual characters were removed (no stop word removal was undertaken). The process constructed an attribute group that would have a high frequency count for common URL elements (words).

4 Evaluation

This section describes the results of the sequence of experiments conducted to identify the most appropriate set of features, considering a number of clustering algorithms, in the context of web-site boundary identification and with respect to the web-site definition given in Section 2. The clustering algorithms used are briefly reviewed in Sub-section 4.1. The test data is described in Sub-section 4.2. The evaluation strategy adopted is introduced in Sub-section 4.3. The results are presented and discussed in Sub-section 4.4.

4.1 Clustering Algorithms

Four different clustering algorithms were selected to evaluate the proposed web-site boundary identification process: two variants of the well-known k-means process (k-means and Bisecting k-means), k-nearest neighbour, and the DBSCAN. A brief overview of each is given below:

k-means: The *k-means* algorithm is an example of an iterative partitional algorithm [8]. It operates on the actual feature space of the items. Items are allocated to a user specified number of k clusters. Only "spherical" shaped cluster are found, and the process has the disadvantage that results can be influenced by outliers.

Bisecting k-means: The *bisecting k-means* clustering algorithm is a partitional clustering algorithm that works by computing a user specified k number of clusters as a sequence of repeated bisections of the feature space. A k-way partitioning via repeated bisections is obtained by recursively computing 2-way clusterings. At each stage one cluster is selected and a bisection is made[22].

k-nearest neighbour: The *k-nearest neighbour algorithm* is an iterative agglomerate clustering algorithm [8]. Items are iteratively merged into existing clusters that are "closest", within a user specified threshold value. If items exceed the threshold, they start a new cluster. The algorithm has the ability to find arbitrary shaped clusters in the feature space.

DBSCAN: The *DBSCAN (Density-Based Spatial Clustering of Applications with Noise)* algorithm creates clusters that have small size and density [9]. Density is defined as the number of points within a certain distance of one another. Note that the number of clusters, k is not input, but it is determined by the algorithm.

The selection of candidate clustering algorithms was made according to the distinctiveness of their operation. To remove the dependence on the number of clusters of some of the algorithms, in each case the cluster containing the start page of the crawl in each data-set was designated as the *target* cluster K_T (ideally, such cluster would include all pages belonging to the web-site to be archived). All other clusters were then identified as *noise* cluster K_N.

4.2 Test Data

For the purposes of the experiments a collections of web-pages was obtained by crawling the University of Liverpool's WWW site. For the evaluation four sets of web-pages were obtained, comprising 500 pages each, and describing the activity of a number of University Departments, namely: (i) Chemistry, (ii) Mathematics, (iii) History, and (iv) Archaeology, Classics and Egyptology. The data sets were identified as: $LivChem500$, $LivMaths500$, $LivHistory500$ and $LiveSace500$.

4.3 Evaluation Strategy

For evaluation purposes the four clustering algorithms identified above were applied to the data collection several times to identify each of the four University Departments web-sites, each time using different groups of features to characterize the given web-pages. The objective on each occasion was to correctly identify all pages describing a particular department, and group in a generic "noise" cluster all other pages. For each experiment one of the data sets was identified as the target class, C_T, and the remainder as noise. The results are presented in the following section.

Two measures were used to evaluate the quality of the resulting cluster configuration: (i) accuracy and (ii) entropy. The accuracy was calculated as the sum of the correctly classified target class web-pages within K_T plus the sum of the number of "noise" web-pages correctly allotted outside K_T divided by the total number of web-pages. Thus:

$$accuracy = \frac{correctClass(K_T) + \sum_{i=1}^{i=n} correctClass(K_i)}{|W|} \tag{1}$$

Where the function *correctClass* returns the number of correctly classified items in its argument, which must be a cluster, K_T is the target cluster, K_1 to K_n are the remaining clusters and W is the input set.

Similarly, denoting by m_{TT} (resp. m_{TN}) the number of pages from (resp. not in) the given web-site (according to the human classification) that land in K_T, the entropy for K_T is defined as:

$$e_T = -\frac{m_{TT}}{|K_T|}\log\frac{m_{TT}}{|K_T|} - \frac{m_{TN}}{|K_T|}\log\frac{m_{TN}}{|K_T|} \tag{2}$$

(here, clearly, the size of cluster K_T satisfies $|K_T| = m_{TT} + m_{TN}$). Therefore, the total entropy of the resulting set of clusters, is calculated as:

$$\frac{|K_T|e_T + (500 - |K_T|)e_N}{500} \tag{3}$$

(where e_N is defined in a similar way to e_T with respect to K_N).

4.4 Results and Discussion

The results of the experiments are presented in this section. Table 1 describes results using the Chemistry data-set. The table presents a comparison of the effectiveness of the proposed web-site boundary identification process using: (i) different web-page features and (ii) different clustering algorithms. The first column lists the feature of interest. The *Composite* feature (row 1) combines all features except the textual content feature (row 2). The second column gives the clustering algorithm used, and the third the value of any required parameters. The clustering algorithm, with respect to each feature, are ordered according to the accuracy value (column 5) obtained in each case. The fourth column gives the entropy value obtained using each of the 10 identified features with respect to each of the clustering algorithms.

Similar experiments were conducted with respect to the other data sets. Table 2 summarises the entire set of experiments. The column headings are the same as for Table 1. For each target class the best two performing features (according to the accuracy measure) were selected and reported in Table 2.

Discussion of Feature Selection. The first observation that can be made from Table 1 is that the entropy and accuracy measure corroborate each other. The second observation is that *Resource Links*, *Image Links*, *Mailto Links* and *Page Anchors*, when used in isolation, are poor discriminators. With respect to accuracy the best discriminators are (in order): *Composite*, *URL*, *Hyper Links* and *ScriptLinks*. In terms of maximising the entropy the best features are (in order): *ScriptLinks*, *URL*, Hyperlinks and Composite. Putting these results together we can observe that there are clear candidates for the most appropriate features to use for boundary identification. There is two possible reasons for the poor performance of the trailing features. One reason could relate to the absence of a feature, this could be a consequence of a specific design choice or function of a web-page. In terms of page anchors and mailto links, these feature will only be present if the specific function is needed/used for a certain page, mailto link may not be provided, or page anchors might not be used. The second reason might be because of the common presence of the feature amongst all pages in the dataset. The pages collected in the dataset that are classed as irrelevant (i.e not in the target class CT) still come from various divisions of the Liverpool University. If many pages use many common images, scripts or resource links, distinguishing between pages may prove quite difficult if the pages only vary by a small degree. Finally, it is perhaps worth noticing that the composite feature acts as a boost in terms of dissimilarity between pages. As described above, if the difference in the pages using a single feature are very small, then combining features will increase this small distinction, to provide a more detectable difference in the inter page dissimilarity between groups. It also copes well with missing features, as the composite feature provides other items that can be present to correctly classify data items.

Inspection of Table 2 indicates that the best discriminators, across the data sets, are: *Composite*, *URL*, *Hyperlinks* and *Textual.* The composite feature

Table 1. Clustering accuracy and entropy results obtained using *LivChem*500, different features and using different clustering algorithms. (Results ordered by clustering algorithm with respect to best average performing feature, according to accuracy).

Chemistry Department 500 (LivChem500)				
Feature	**Algorithm**	**Params (optimal)**	**Entropy (%)**	**Accuracy (%)**
Composite	Bisecting Kmeans	k=4	87.09%	98.2%
	Kmeans	k=4	86.99%	98%
	DBSCAN	minPoints=1, eps=250	62.82%	91.8%
	KNN	Threshold=20	46.07%	13.2%
Textual	Kmeans	k=5	81.09%	96.8%
	Bisecting Kmeans	k=4	66.18%	91.6%
	DBSCAN	minPoints=1, eps=999	48.99%	88.6%
	KNN	Threshold=25	64.02%	23.4%
URL	Bisecting Kmeans	k=4	87.42%	98.2%
	Kmeans	k=4	85.86%	98.1%
	DBSCAN	minPoints=1, eps=5	58.72%	91.6%
	KNN	Threshold=5	45.92%	12.4%
Hyperlinks	Kmeans	k=5	87.28%	98.2%
	Bisecting Kmeans	k=5	65.78%	93%
	DBSCAN	minPoints=1, eps=250	55.87%	90.6%
	KNN	Threshold=30	45.96%	12.6%
Title	Kmeans	k=6	83.98%	97%
	DBSCAN	minPoints=3, eps=5	54.41%	90.4%
	Bisecting Kmeans	k=4	60.14%	84.6%
	KNN	Threshold=5	45.92%	16.8%
ScriptLinks	Kmeans	k=4	88.25%	97.8%
	Bisecting Kmeans	k=3	64.08%	91.8%
	DBSCAN	minPoints=3, eps=5	47.73%	46.6%
	KNN	Threshold=1	45.92%	12.4%
ResourceLinks	DBSCAN	minPoints=3, eps=5	63.23%	91.8%
	Bisecting Kmeans	k=5	52.85%	63%
	Kmeans	k=5	55.29%	61.6%
	KNN	Threshold=5	45.92%	12.4%
MailtoLinks	Bisecting Kmeans	k=6	48.00%	74.2%
	Kmeans	k=7	46.00%	12.8%
	DBSCAN	minPoints=1, eps=200	45.92%	12.4%
	KNN	Threshold=5	48.98%	11.4%
ImagesLinks	Bisecting Kmeans	k=6	46.05%	34.4%
	Kmeans	k=8	48.95%	27%
	DBSCAN	minPoints=1, eps=250	46.19%	13.8%
	KNN	Threshold=15	46.11%	13.4%
PageAnchors	Bisecting Kmeans	k=9	45.92%	12.4%
	Kmeans	k=9	45.92%	12.4%
	KNN	Threshold=5	45.92%	12.4%
	DBSCAN	minPoints=1, eps=1	45.92%	12.4%

Table 2. Best results for all four test set combinations (Results ordered by clustering algorithm with respect to best average performing feature, according to accuracy)

Departments from University Of Liverpool				
Best performing feature	**Best performing Algorithm**	**Params (optimal)**	**Entropy (%)**	**Accuracy (%)**
Chemistry Department (LivChem500)				
Composite	Bisecting Kmeans	k=4	87.09%	98.2%
	Kmeans	k=4	86.99%	98%
URL	Bisecting Kmeans	k=4	87.42%	98.2%
	Kmeans	k=4	85.86%	98.1%
Mathematics Department (LivMaths500)				
Textual	Bisecting Kmeans	k=8	76.3%	96%
	Kmeans	K=7	75.35%	95.8%
Hyperlink	DBSCAN	minPoints=3, eps=5	69.72%	94.4%
	KNN	Threshold=90	44.16%	86.8%
History Department (LivHistory500)				
Composite	Bisecting Kmeans	k=3	77.28%	96%
	Kmeans	k=6	72.83%	95.2%
Hyperlinks	Bisecting Kmeans	k=3	75.06%	92.6%
	Kmeans	k=5	72.25%	95.2%
School of Archaeology, Classics and Egyptology (LivSace500)				
Composite	Bisecting Kmeans	k=5	82.33%	89.2%
	Kmeans	k=9	85.03%	93.2%
Hyperlinks	Bisecting Kmeans	k=4	72.84%	79%
	Kmeans	k=3	74.36%	70%

performs the best in three out the four cases and can thus be argued to have the best performance overall. It is conjectured that this is because it is the most robust comprehensive representation, and thus can operate better with respect to missing or irrelevant values in the vector space (compared to using features in isolation). For example, title seems to be a good indicator of pages in the same web-site, but if a title tag is missing then the page will be missed completely. Using a composite set of features boosts the performance, and helps find pages that span across multiple domains and services within the input data. There are some cases were the Textual (content) works well. However, content tends to be dynamic and is subject to change; it is suggested that the composite feature representation would be able to deal effectively with such changes.

In general, it can be said that the features considered in the composite feature include attributes of a web-page that are more representative of authors' overall intentions, rather than the authors means of conveying an idea. The composite feature representation will model a page using: the URL which can be considered as the place in the web structure it resides, the title provides a round-up of the overall message the page conveys, the hyperlinks consider the position it is in the website site structure (home page, leaf node etc); while the resource, script, mail, image and page anchors links provide a consistent representation of the skeleton structure of the page. All features in combination perform better than in isolation. The performance is better than only textual content, which can be thought of as a representation of the target information an author is trying to convey at a specific point. This is subject to change as events/schedules or activities change. The main skeleton structure will remain fairly consistent, and thus, in the experiments conducted in this paper, prove to be a better model for website boundary identification according to our definition.

Discussion of Clustering Algorithms. The best overall clustering algorithms tend to be Bisecting Kmeans and Kmeans. It is worth noting that the feature space that is created from each of the web-page models is quite dense, with low ranges of values with occasional outliers, and with very high frequency of certain features. Consequently the KNN and DBSCAN algorithms tend to produce clustering results that merge almost all items into a single cluster, or they overfit, and produce a single cluster for each data item (note that these clustering algorithms do not work with a predetermined number of clusters). The items in the feature space are densely packed so even using low threshold values cannot produce distinctions between related and non related items. This observation is also reflected with respect to the Kmeans and bisecting kmeans algorithms when a low initial cluster value (k) is used; in this case it can also be seen that the majority of items are grouped together, this is contrary to what we might expect to be produced, i.e. a cluster containing items from the ideal class and another cluster containing the remaining items.

In the early stages of the investigation it was thought that a cluster value of $K = 2$ for Bisecting Kmeans and Kmeans would be the most appropriate to distinguish between desired web-pages from the target class (C_T), and web-pages that are irrelevant (noise included in the crawl). However, from test results, it quickly became apparent that using $K = 2$ did not provide any useful distinction in the data sets. This was because the clusters produced by Bisecting Kmeans and Kmeans are Hyper spheres, i.e with equal radii in all n dimensions. Any change in the cluster radius in any specific dimension impacted on all dimensions which meant that in some cases, given a low number of clusters (i.e. $K = 2$), some "short" dimensions was entirely encompassed by a single cluster. By increasing the value of K much better results were produced as clusters were not able to grow in the same manner as with low values of K. Thus a high initial cluster value (K) was eventually used so as to distinguish between items in the densely packed feature space. The effect of this was to force the generation of many cluster centroids (in the case of kmeans) or many bisections (in the case of bisecting kmeans), This method of using high initial cluster values was re-enforced by the adverse results obtained using DBSCAN and KNN which do not operate with an initial number of cluster parameters, and instead tried to adapt to the feature space. DBSCAN and KNN either produced single clusters containing most items, or they "over-fitted" and generated a large number of clusters each containing very few items.

It can be argued, out of the clustering algorithms that were tested, that the Bisecting kmeans seemed to produce the overall best performance. The reason for this is that it suffered much less with initialisation issues; and that the feature space is bisected on each iteration which produced clusters that were not limited by *centroid distance*, as in the case of Kmeans (and others).

The method of using a high initial clustering value proved to have very good results when combined with the composite web-page representation. The features in isolation were out performed by the more robust composite feature, which is also true for the content (textual) representation. The composite feature representation using high initial cluster value for the Bisecting Kmeans algorithm produced a better more consistent performing result that fits our selective archiving application.

5 Conclusions

An approach to the clustering of web-pages for the purpose of web-site boundary detection has been described. The reported study focuses firstly on the identification of the most appropriate WWW features to be used for this purpose, and secondly on the nature of the clustering algorithm to be used. The evaluation indicated that web-page clustering can be used to group related pages for the purpose of web-site boundary detection. The most appropriate features, identified from the experimentation were *Composite*, *URL*, *Hyper Links* and *ScriptLinks*. These *Composite* features can be argued to be the most appropriate because it appears to be the least sensitive to noise because it provided a much more comprehensive representation (although it required more computation time to process). The most appropriate clustering algorithms, from the four evaluated, were found to be Bisecting Kmeans and Kmeans.

There are many applications that may benefit from the work. described Examples include: (i) WWW spam detection, (iii) creation of WWW directories, (iii) Search Engine Optimisation (SEO) and (iv) the generation of site maps. In future work the research team are interested in conducting experiments using much bigger data sets, including some currently popular web-sites.

References

1. Antoniol, G., et al.: Web site: files, programs or databases? In: Proceedings of WSE 1999: 1st International Workshop on Web Site Evolution (October 1999)
2. Asano, Y., Imai, H., Toyoda, M., Kitsuregawa, M.: Applying the site information to the information retrieval from the web. In: Ling, T.W., Dayal, U., Bertino, E., Ng, W.K., Goh, A. (eds.) WISE, pp. 83–92. IEEE Computer Society, Los Alamitos (2002)

3. Bharat, K., Chang, B.w., Henzinger, M., Ruhl, M.: Who links to whom: Mining linkage between web sites. In: Cercone, N., Lin, T.Y., Wu, X. (eds.) ICDM, pp. 51–58. IEEE, Los Alamitos (2001)
4. Broder, A.Z., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomikns, A., Wiener, J.: Graph structure in the Web. In: Proceedings of the Ninth Internation World Wide Web Conference (WWW9)/Computer Networks, vol. 33, pp. 1–6. Elsevier, Amsterdam (2000)
5. Dmitriev, P.: As we may perceive: finding the boundaries of compound documents on the web. In: Huai, J., Chen, R., Hon, H.-W., Liu, Y., Ma, W.-Y., Tomkins, A., Zhang, X. (eds.) WWW, pp. 1029–1030. ACM Press, New York (2008)
6. Deegan, M., Tanner, S. (eds.): Digital Preservation. Digital futures series (2006)
7. Dmitriev, P., Lagoze, C., Suchkov, B.: Finding the boundaries of information resources on the web. In: Ellis, A., Hagino, T. (eds.) WWW (Special interest tracks and posters), pp. 1124–1125. ACM, New York (2005)
8. Dunham, M.H.: Data Mining: Introductory and Advanced Topics. Prentice-Hall, PTR, Upper Saddle River (2002)
9. Ester, M., Kriegel, H.-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Simoudis, E., Han, J., Fayyad, U.M. (eds.) Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996), pp. 226–231. ACM, New York (1996)
10. Hine, C.: Virtual methods: issues in social research on the Internet. Berg (2005)
11. Kleinberg, J.M.: Authoratitive sources in a hyperlinked environment. Journal of the ACM 46(5), 604–632 (1999)
12. Kumar, R., Punera, K., Tomkins, A.: Hierarchical topic segmentation of websites. In: Eliassi-Rad, T., Ungar, L.H., Craven, M., Gunopulos, D. (eds.) Proceedings of the Twelfth International Conference on Knowledge Discovery and Data Mining (KDD 2006), pp. 257–266. ACM, New York (2006)
13. Li, W.-S., Kolak, O., Vu, Q., Takano, H.: Defining logical domains in a web site. In: Hypertext, pp. 123–132 (2000)
14. Liu, B.: Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data. Springer, Heidelberg (2007)
15. Nielsen, J.: The rise of the subsite. useit.com Alertbox for September 1996 (September 1996)
16. Rodrigues, E.M., Milic-Frayling, N., Hicks, M., Smyth, G.: Link structure graph for representing and analyzing web sites. Technical report, Microsoft Research. Technical Report MSR-TR-2006-94, June 26 (2006)
17. Rodrigues, E.M., Milic-Frayling, N., Fortuna, B.: Detection of web subsites: Concepts, algorithms, and evaluation issues. In: Web Intelligence, pp. 66–73. IEEE Computer Society, Los Alamitos (2007)
18. Schneider, S.M., Foot, K., Kimpton, M., Jones, G.: Building thematic web collections: challenges and experiences from the september 11 web archive and the election 2002 web archive. In: Masanès, J., Rauber, A., Cobena, G. (eds.) 3rd Workshop on Web Archives (In conjunction with the 7thEuropean Conference on Research and Advanced Technologies for Digital Libraries, ECDL 2003), pp. 77–94 (2003)
19. Senellart, P.: Website identification. Technical report, DEA Internship Report (September 2003)

20. Senellart, P.: Identifying websites with flow simulation. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 124–129. Springer, Heidelberg (2005)
21. Xi, W., Fox, E.A., Tan, R.P., Shu, J.: Machine learning approach for homepage finding task. In: Laender, A.H.F., Oliveira, A.L. (eds.) SPIRE 2002. LNCS, vol. 2476, pp. 145–159. Springer, Heidelberg (2002)
22. Zhao, Y., Karypis, G.: Clustering in life sciences. In: Brownstein, M., Khodursky, A., Conniffe, D. (eds.) Functional Genomics: Methods and Protocols (2003)

An Application of Element Oriented Analysis Based Credit Scoring

Yihao Zhang[1], Mehmet A. Orgun[1], Rohan Baxter[2], Weiqiang Lin[2]

[1] Department of Computing, Macquarie University Sydney, NSW 2109, Australia
{yihao,mehmet}@ics.mq.edu.au
[2] Australian Taxation Office, Canberra ACT 2601, Australia
{rohan.baxter,wei.lin}@ato.gov.au

Abstract. In this paper, we present an application of an Element Oriented Analysis (EOA) credit scoring model used as a classifier for assessing the bad risk records. The model building methodology we used is the Element Oriented Analysis. The objectives in this study are: 1) to develop a stratified model based on EOA to classify the risk for the Brazilian credit card data; 2) to investigate if this model is a satisfactory classifier for this application; 3) to compare the characteristics of our model to the conventional credit scoring models in this specific domain. Classifier performance is measured using the Area under Receiver Operating Characteristic curve (AUC) and overall error rate in out-of-sample tests.

Keywords: Credit Scoring, Classifiers, Element Oriented Analysis, AUC.

1 Introduction

Credit scoring models have been widely used in industry and studied in various academic disciplines. We make two broad observations of the existing literature.

On the one hand, the existing credit scoring models are sensitive to the specific domain and available dataset. As a result, they might change significantly over variant domains or datasets. For example, Neural Networks performed significantly superior to LDA in predicting bad load [4]. Whereas Yobas et al [14] reported that the latter outperformed the former in bank credit account assessment. Similar problems also appeared among the applications of Logistic Regression model, *k*NN model and Decision Tree model [1]. On the other hand, domain knowledge and experience are the important aspects to influence the risk assessment. This point has also been demonstrated by many reported works, such as [2], [8] and [10].

This paper makes the following three contributions. First, we develop a stratified credit scoring model based on Element Oriented Analysis (EOA) methodology. Second, we apply the model to the Brazilian credit card data. Third, we evaluate the performance of the EOA model relative to some conventional models on this particular dataset.

The empirical results in this paper are focussed on the Brazilian credit dataset from the PAKDD09 data mining competition. This provides a good basis for comparison of the EOA model with other models used in that competition [2], [8], [9] and [10].

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 544–557, 2010.

Realistic available credit datasets are rare. The Brazilian credit data was available (with reasonable PAKDD09 competition restrictions). The widely used German credit data can criticised for being unrealistic because it is small, has few variables, and has been pre-cleaned.

The rest of the paper is organised as follows. Section 2 introduces the development of our stratified model, including a comprehensive explanation of EOA and a complete framework for the EOA model. We discuss related credit scoring models in Section 3. Then we describe the application data and procedure in Section 4, followed by the experimental results. We also discuss the related work from PAKDD09 competition in Section 5. Finally, Section 6 concludes with a summary of the paper's contributions and proposes possible directions for future work.

2 Credit Scoring Model Based on Element Oriented Analysis

In this section, we firstly review some key concepts of EOA. Then we describe how a credit scoring model is developed based on EOA.

2.1 Element Oriented Analysis

EOA is a methodology for developing predictive models and is not an algorithm. The EOA methodology involves the design of new features or attributes based on segmentation. EOA has been used to predict corporate bankruptcy by Zhang et al [15]. In that application, the data was segmented and the segment characteristics used to add new informative features. To better understand our model, it is necessary to briefly introduce the idea of EOA with the following definitions.

Definition 1. Let $D = (d_1, d_2, \ldots, d_m)$be a dataset and $d_i (1 \le i \le m)$be the i^{th} observation with d_i possibly being a vector. If $\forall d_i$ there exists k common intrinsic properties, and they can be represented by the following function (1).

$$d_i = s_{i1} \otimes, \ldots, \otimes\, s_{ik} \; (1 \le i \le m) \tag{1}$$

where s_{ik} the common intrinsic property and $\otimes$ represents the only relationship between the different properties. s_{ik} is defined to be an Element of data point d_i.

Example. Suppose that a dataset consists of ten observations with one binary target attribute (*Y*) and one numeric explanatory attribute (*X*) shown in below

Y: 0, 1, 1, 0, 1, 1, 0, 0, 1, 0
X: 7, 3, 4, 6, 3, 4, 7, 6, 4, 7

Suppose that we need to find the relationship explaining what kind of X is more likely to cause $Y = 1$. According to Definition 1, some intrinsic properties are extracted into the new informative features. One of the simple ways to define an informative feature is as follows. Consider an Element s_1 to represent the probability of an explanatory variable given the occurrence of the target variable Y. Then we might find $P(x = 6\ or\ 7|y = 1) = 0, P(x = 3|y = 1) = 0.4$ and $P(x = 4|y = 1) = 0.6$.

We may define another Element s_2 to depict the dataset from the viewpoint of the observations across all attributes. As a result, we choose s_2 to be the probability of belonging to an overall segment for the observation between $Y = 1$ and $⟨ = 0$. For example, we can use a clustering algorithm to calculate the probability of a data item being in a particular segment. In this example, we obtain $P'(x = 7|y = 1) = 0.11$, $P'(x = 3|y = 1) = 0.86$ $P'(x = 4|y = 1) = 0.87$ and $P'(x = 6|y = 1) = 0.2$ (Please note that more details of obtaining the Element are stated in Section 2.3). Now we generate two Elements to replace the original explanatory variables as shown in Table 1.

Table 1. Dataset Based on the New Elements

Obs.	Y	X	s_1	s_2
1	0	7	0.6	0.11
2	1	3	0	0.86
3	1	4	0	0.87
4	0	6	0.4	0.2
5	1	3	0	0.86
6	1	4	0	0.87
7	0	7	0.6	0.11
8	0	6	0.4	0.2
9	1	4	0	0.87
10	0	7	0.6	0.11

These two Elements are designed to represent the latent structure between the target variable (Y) and the original explanatory variable (X). Therefore, they fit the definition of a Structure Element in the following Definition 2.

Definition 2. The Element is defined as a Structure Element if it satisfies two conditions:

1. It contains the same number of observations as the original dataset.
2. It states the original data in terms of either the view of attributes or the whole dataset.

The defined Elements are chosen based on insights and knowledge of the intended application. The two Elements defined above use segments or strata. We now need describe how the Elements are used to do the predictions. Generally, EOA has two steps.

Definition 3. The Local Level (LL) calculates the Elements as defined for an application. The Global Level (GL) does the calculation to use the Elements to meet the modelling objective.

In the current application, the modelling objective is a classifier predicting whether a credit card holder is a good or bad. EOA has been applied to other applications such as short-term time series prediction where the modelling objective is the prediction of the next k-values in a time series.

Definition 4. Element Oriented Analysis (EOA) methodology has the following components:

1. New Elements representing the informative features are generated by segmenting the original dataset.
2. The resulting model uses the new Elements (and optionally the original data)
3. The resulting model is multi-level using a Local-Global hierarchy resulting from the use of new Elements based on segments and original data.

The same dataset could be segmented into the Elements in many different ways, which depend on the data domain and intended applications of the model. An important part of the EOA model application is the discovery and design of the Elements.

2.2 EOA Credit Scoring Model

Suppose that a target variable Y can be predicted by a linear function of the explanatory variables X_i *(i=1...p)*. Some parameters such as the intercept β_0 and the slope β_i are estimated in function (2)

$$Y_i = \beta_0 + \sum_{i=1}^{p} \beta_i X_i + \epsilon \tag{2}$$

where ϵ is the unobserved random variable. In the usual cases of credit risk assessment, the target variable is assumed to be the binary value using 0 for good risk and 1 for bad risk. Hence, there exists a conditional probability of the binary variable Y given the value of X_i as function (3)

$$P\,r\left(Y = 1 \middle| X_1 \dots X_p\right) = \frac{1}{1+e^{-z}} \tag{3}$$

and the probability of the contrary event is function (4)

$$Pr\left(Y = 0 \middle| X_1 \dots X_p\right) = \frac{e^{-z}}{1+e^{-z}} \tag{4}$$

where $z = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p + \epsilon$. In terms of Definition 2, we now define two Structural Elements (SE) s_1 and $s_2$to explain the observations. The s_1 element puts observations into attribute (or column) based segments while the s_2 element puts observations into record (or row) based segments. According to function (1), we now have a transferring function (5)

$$X_i = s_{1i} \otimes s_2 \; (1 \le i \le m) \tag{5}$$

As a result, regression function (2) is converted to (6)

$$z' = \beta_0' + \sum_{i=1}^{p} \beta_i' \, s_{1i} + \beta_{i+1}' s_2 \tag{6}$$

And the credit scoring model is built by following function (7)

$$\log \frac{Pr(Y=0|s_{1i}, s_2)}{Pr(Y=1|s_{1i}, s_2)} = \beta_0' + \sum_{i=1}^{p} \beta_i' \, s_{1i} + \beta_{i+1}' s_2 \tag{7}$$

According to Definition 3, our model is designed with LL and GL. The comprehensive development framework for our model is shown in Figure 1. We need to clarify that Logistic Regression is tentatively applied in GL in this application. However, our model is not limited to this technique only.

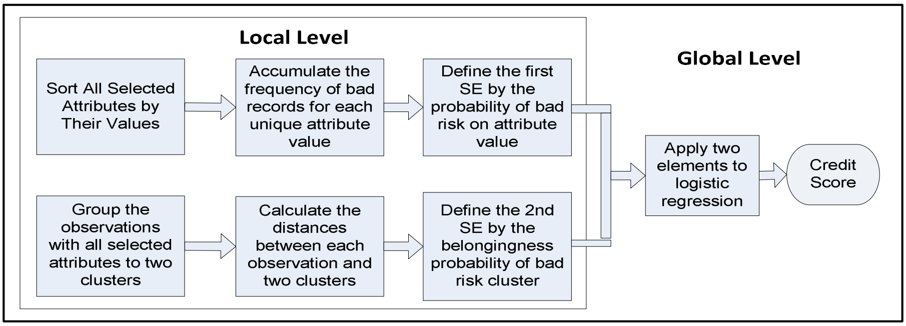

Local Level
Sort All Selected Attributes by Their Values
Accumulate the frequency of bad records for each unique attribute value
Define the first SE by the probability of bad risk on attribute value
Group the observations with all selected attributes to two clusters
Calculate the distances between each observation and two clusters
Define the 2nd SE by the belongingness probability of bad risk cluster
Global Level
Apply two elements to logistic regression
Credit Score

Fig. 1. The Framework of EOA Model in Credit Scoring

2.3 Finding the Structural Element in Local Level

To simplify the explanation, we first assume a sample of *n* independent observations (*x, y*), where *y* denotes the observed value of a binary variable (e.g. good/bad credit risk) and *x* denotes the explanatory variable. Following the EOA framework, we define two SEs to understand the dataset accurately. The first SE s_1 is given by function (8) to discover the probability of explanatory variable *x* given the occurrence of target variable *y*.

$$s_1 = P(x|y=1) \; or \; s_1 = P(x|y=0) \tag{8}$$

To provide a different view of a data observation, we define SE s_2 from the viewpoint of the observations across the attribute. In the other words, this SE should state the probability of an observation belonging to a segment of good credit risk or bad credit risk. To find s_2, we firstly generate two clusters through the distance function (9).

$$d_i = (x-c)^T A(x-c_i), (i=1,2) \tag{9}$$

Where A is the distance norm matrix, c_i is the cluster. Then the s_2, which represents probability of an observation belonging to a segment, which is calculated by the membership functions (10).

$$聽_2 = \frac{1}{\sum_{j=1}^{2}\frac{d_1}{d_i}} \; or \; s_2 = \frac{1}{\sum_{j=1}^{2}\frac{d_2}{d_i}} \tag{10}$$

2.4 Estimate Credit Score in Global Level

Based on the function (7), we fit a Logistic Regression model with the elements as inputs. The logistic regression model parameters are estimated by minimising the log-likelihood (11)

$$L = \sum_{i=1}^{m}[z_i log p_i + (1-z_i) log(1-p_i)] \tag{11}$$

where $p_i = 1/(1+e^{-z_i})$ and $z = \beta_o + \sum_{i=1}^{p} \beta_i \, s_{1i} + \beta_{i+1} s_2$. L uses the maximum likelihood method that depends on the estimation of $\beta_o, \ldots, \beta_{i+1}$. The estimated probability of target variable is calculated when our model is fitted in the training data. Then our model starts to group the observations based on the following rule (12)

$$Y = \begin{cases} 0, if\ Pr(Y = 0|s_1, s_2) \geq 0.5 \\ 1, if\ Pr\ (Y = 0|s_1, s_2) < 0.5 \end{cases} \tag{12}$$

3 Related Work

Reichert et al. [11] first proposed to assess credit risk by Linear Discriminate Analysis (LDA). They assigned the observations to the "good applicant" or "bad applicant" class by the posterior probability, which is calculated through Bayes' theorem. As a result, their model showed good performance in the normal distributed data. In other words, LDA can make a good linear classification if the covariance matrix can indicate the class boundary clearly. Henley [5] applied logistic regression to the credit scoring applications in his dissertation. Both LDA and logistic regression are parametric models.

Along with the development of parametric models, some non-parametric models have also been used to assess the credit risk. For example, the kernel density based techniques achieved good results for a two dimensional dataset [12]. The *k*NN (*k*-Nearest Neighbour) classifier has been used for a credit scoring model [6]. The basic idea of *k*NN is to classify the observations into the good or bad groups by their *k* most similar neighbours. Some experiments showed *k*NN model outperformed kernel based models in multidimensional datasets [6] [12].

Decision Tree models have also been developed within data mining community [3]. As a rule inductive algorithm, decision tree models first learn the rule information from the samples iteratively. The observations are then distinguished by those learnt rules. Recently, neural networks have also been applied to support credit risk analysis [7] [13]. Their non-linear function learning systems are recognised to be effective on real world data, such as credit card data and home loan data [7].

4 An Application on the Brazilian Credit Card

In this section, we apply our model to the Brazilian credit card dataset from the PAKDD09 competition. The dataset is available from the website http://sede.neurotech.com.br:443/PAKDD2009/. The competition datasets are derived from one continuous dataset originally. The organisers divided it into three datasets. One labelled dataset is provided for training purpose, and the other two datasets are unlabelled, which are set to test competitors' results. Due to the fact that the competition has been closed and we cannot get the unlabelled results, we adopt the labelled data in this application only.

4.1 Data Description and Pre-processing

The selected dataset consists of 50000 observations with 32 attributes. The ratio of dataset between risky observations and safe ones is 20% to 80%. According to the preliminary knowledge from the competition, the attribute labelled as "TARGET_LABEL_BAD_1" is the target variable with Boolean value that 0 denotes safe observation whereas 1 denotes risky one. However, the dataset encounters the issue of incompletion and diversity. Hence we pre-process the data in the beginning of the experiment. The pre-processing of the rest of 31 attributes together with the simple data descriptions are shown in appendix 1.

4.2 The Results

After pre-processing, we get a new dataset with 18 explanatory variables, an identification variable and a target variable. 49907 observations (93 observations are deleted as their values of target variable are null) consist of 40049 safe observations and 9858 risky ones. The training/test data are generated using a 50% random split.

Our experiment has two stages. In the first stage, we demonstrate the characteristics of our model on the training and test datasets. To apply the EOA model, the following calculations are done: calculate the two SEs, fit the SEs to the Logistic Regression. Then test performance of the resulting model on out of sample data. In the second stage we investigate the robustness of our model by testing out of sample data on the different runs and comparing our model to some conventional credit scoring models on the different split of training/test data. We apply to measure both training and test results using Area under the Curve (AUC) derived from the ROC and the overall error rate.

4.2.1 Stage One – Modelling and Testing

Following the EOA framework, we define 18 first SE giving the probability of bad risk for the original 18 selected explanatory variables to replace the original ones (A1-A18) in the new dataset. And a second type of SE defining the probability of membership of a bad risk cluster is added. The selection of the suitable explanatory variables from the new dataset depends on coefficients with significance level or 0.005. Table 2 shows the training results for the selected explanatory variables.

Table 2. The Results of Multivariate Testing

P.	Est.	Std. Error	Wald Chi-Sq.	Pr > Chi-Sq.	Select?
Int	0.0310	0.7797	0.0016	0.9683	N
s_1	-1.5666	0.3461	20.4865	<.0001	Y
s_2	-0.7079	0.7086	0.9981	0.3178	N
s_3	-2.3897	0.6545	13.3301	0.0003	Y
s_4	-0.8214	0.2248	13.3485	0.0003	Y
s_5	-2.3923	0.1689	200.5355	<.0001	Y
s_6	-2.0592	0.2644	60.6754	<.0001	Y
s_7	-2.5196	0.3485	52.2728	<.0001	Y
s_8	-6.4221	1.3099	24.0357	<.0001	Y
s_9	-1.4159	0.3390	17.4409	<.0001	Y
s_{10}	-1.9306	0.2008	92.4867	<.0001	Y
s_{11}	-2.5967	0.1484	306.2052	<.0001	Y
s_{12}	-1.1876	0.4341	7.4835	0.0062	N
s_{13}	-0.3539	0.4457	0.6304	0.4272	N
s_{14}	-1.8058	2.0924	0.7448	0.3881	N
s_{15}	-2.7061	0.6110	19.6157	<.0001	Y
s_{16}	-2.1595	2.6541	0.6620	0.4158	N
s_{17}	27.6272	1.8639	219.7100	<.0001	Y
s_{18}	-3.4889	0.3492	99.8379	<.0001	Y
m	1.3798	0.5634	5.9975	0.0021	Y

(*Note that s_1-s_{18} is the first type SE for the original 18 selected explanatory variables (A1-A18), m is the second type SE*)

The resulting model is used to make an out-of-sample test on another 50% observations. The result is shown in the following Figure 2(a). Also the precision results are concluded in the Figure 2(b).

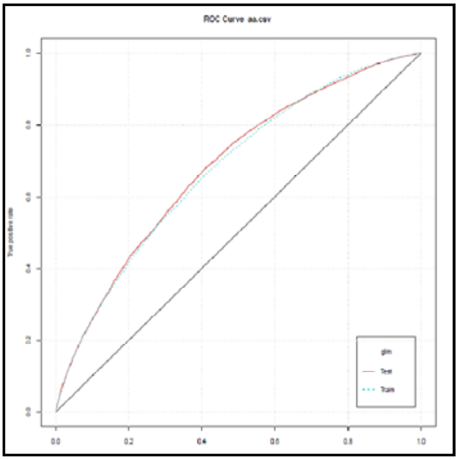


Fig. 2(a). ROC Curve for Our Model Estimating the Probability of Bad Credit Risk

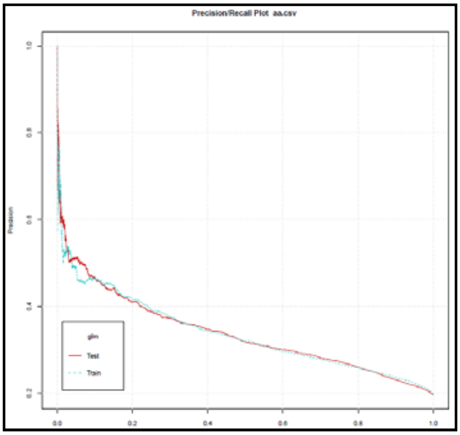


Fig. 2(b). Precision Curve for Our Model Estimating the Probability of Bad Credit Risk

(*The curves are based on a single training and out of sample test. The red curve indicates the training result while the blue one shows test.*)

The AUC results and the overall error rate are shown in the following Table 3. Comparing the EOA model results with those from the competition, we find the results are similar (Dasgupta et al, 2009; Kannan and Balakrishnan, 2009; Pfahringer, 2009 and Linhart et al, 2009).

Table 3. The Results of AUC from Our Model

Training		Out of Sample Test	
AUC	Overall Error	AUC	Overall Error
0.6843	0.1969831	0.6718	0.198273

4.2.2 Stage Two – Robustness Validation

To check the robustness of the EOA model, we run the different random training and test dataset ten times. The results are shown in Table 4.

After ten times out of sample testing, the average AUC we obtained from EOA model is 0.666, with standard deviation of 0.0072. The overall error rate is 0.199, these two measures showed the stability and robustness of EOA model.

Table 4. The Robust Validation of Our Model

No.	Training		Out of Sample Test	
	ROC	Overall Error	ROC	Overall Error
1	0.6843	0.1970	0.6718	0.1983
2	0.6891	0.1958	0.661	0.2022
3	0.6868	0.1965	0.6649	0.1995
4	0.6887	0.1973	0.6573	0.2
5	0.6849	0.1969	0.6717	0.1979
6	0.6865	0.1984	0.6621	0.1959
7	0.6901	0.1948	0.6605	0.2024
8	0.6827	0.1970	0.6765	0.1952
9	0.6827	0.1956	0.6781	0.2022
10	0.6875	0.1954	0.665	0.2012
$\bar{M}$	0.6863	0.1965	0.6669	0.1995
Std	0.0026	0.0011	0.0072	0.0026

4.2.3 The Results of Comparison

We provide a comparison between the performance of the EOA model and some alternative models over different training/test dataset split, from 10/90, 30/70, 50/50, 70/30 to 90/10. We use the R packages to test the performance of these models. The details of these models are listed in Table 5.

Table 5. The Benchmark Models

Benchmark	R Package	Model Setting
Decision Tree	rpart	Min Split=20 Max Depth=30 Complexity=0.01
AdaBoost	ada	Min Split=20 Max Depth=30 Complexity=0.01
RandomForest	randomForest	Trees=500
LogisticRegression	glm	Family=binomial Link=logit
Neural Networks	nnet	HiddenNodes=15

The comparison results are shown in Table 6 and Figure 3 below. We use AUC to measure the performance [16].

From the results of comparison, we observe that AUC of our model increases from 0.64 of 10/90 split to 0.67 of 90/10 split. Hence, our model can perform better if it has the higher ratio of the training dataset to the test dataset. In addition, our model starts to outperform all the benchmark models from the 30/70 split. Therefore, we conclude that our model is competitive in this application.

According to the latest publication by Hand [16] who reports that incoherence of AUC might mislead the results through using different misclassification cost distribution for different classifier, we also extend our experiment a bit further to the comparisons of Area Under the convex hull of a ROC curve (AUCH) for 50/50 split. The publically available R package is used in this comparison as well. The results are shown in Table 7 below.

Table 6. The Comparison between Our Model and Some Benchmarks

Benchmark	10/90	30/70	50/50	70/30	90/10
Our Model	0.6426	0.6592	0.6669	0.6671	0.6729
AdaBoost	0.659	0.655	0.6553	0.6543	0.6475
RandForest	0.6411	0.6473	0.6475	0.646	0.6409
Logit	0.6561	0.6575	0.657	0.6593	0.6545
DecisionTree	0.5	0.5	0.5	0.5	0.5
NNs	0.507	0.5077	0.5085	0.5064	0.5088

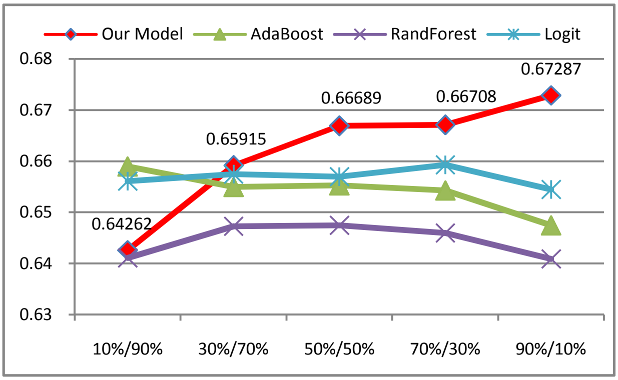


Fig. 3. The Comparison between Our Model and Some Conventional Models over the Different Training/Test Split

Table 7. The Comparisons of AUCH between Three Models

Benchmark	Our Model	AdaBoost	Logit
AUCH (50/50)	0.6687	0.6664	0.6652

In this comparison, we mainly compare the performance of AUCH between AdaBoost, Logit and Our EOA model, which have the leading performances in AUC comparison. Our model again outperforms the other two models although the superiority is very margin. The results also validate the robustness of our model from another view of points.

5 Discussion of the Related Works

Many credit assessment models were built for the Brazilian credit card dataset as part of the PAKDD09 data mining competition. In the following, we discuss work of different models from some top ranked teams (Note that the ranking is in terms of the results on the unlabelled dataset, to which we do not have access).

Equinox team from ANZ bank [2] demonstrated their stratified model in this competition. They trained two Logistic Regression models for the segmentation with AGE>30 and AGE<=30. They explained the reason for selecting AGE to partition data is that AGE has stable performance with respect to PSI (Population Stability Indices). And then the score one from the Logistic Regression model is brought to the score two by applying Naïve Bayesian Model with the Weight of Evidence of Quant_other_Bank_Accounts, Education Status and Cod_Application_Booth. The final credit score is then generated by the average of the score one and the score two. Similarly, Latentview team [8] used the joint score from the weighted average combining the prediction from Logistic Regress, TreeNet, Adaptive Logistic Regression & CART models.

Pfahringer [10] performed a detailed pre-processing of the dataset in his experiment. The processed data is run in four models, which are Logistic Regression, Neural Network, Bagged Boosted Decision Stumps and Ensembles of Boosted Random Rules. He claimed that Ensembles of Boosted Random Rules with 2.5 million rules gains the best performance in the final submission of competition.

Logit team from Tel-Avis University [9] showed their ensemble of models through combining the parametric model and nonparametric model. They also made a major step of data pre-processing before their attempts on Logistic Regression model and *k*NN model respectively. Eventually, they applied the Logistic Regression model to the results from *k*NN model. Their accurate results are validated by a 5-fold cross validation approach.

6 Conclusion

In this paper, we develop a stratified model based on EOA to solve the credit risk problem on the Brazilian credit card dataset. Our approach is different from the related works, most of which have domain knowledge assistance. The main insight in our model is to find two valuable Structural Elements discovering the latent regularities from the attributes and observations respectively. And then these two Structural Elements are regarded as explanatory variables for input into a Logistic Regression model. As the results demonstrated in Section 3, our model is applicable in credit risk assessment in the context of the accuracy measures from the existing works on the same dataset and the benchmark models.

In our future work, our framework will be extended to the standard UCI datasets. This will allow us to better understand the strengths and weaknesses of our approach.

References

1. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society 54, 627–635 (2003)
2. Dasgupta, S., Ismail, Z., Nair, K.C., Gurur, H., Kumar, N. : Team: Equinox (2009), `http://sede.neurotech.com.br:443/PAKDD2009/files` (viewed July 15, 2009)

3. Davis, R.H., Edelman, D.B., Gammerman, A.J.: Machine learning algorithms for credit-card applications. IMA Journal of Management Mathematics 4, 43–51 (1992)
4. Desai, V.S., Crook, J.N., Overstreet Jr., G.A.: A comparison of neural networks and linear scoring models in the credit union environment. European Journal of Operational Research 95(1), 24–37 (1996)
5. Henley, W.E.: Statistical aspects of credit scoring. Dissertation. The Open University, Milton Keynes, UK (1995)
6. Henley, W.E., Hand, D.J.: A k-nearest neighbour classifier for assessing consumer credit risk. The Statistician 45(1), 77–95, 1–34 (1996)
7. Huang, Z., Chen, H., Hsu, C., Chen, W., Wu, S.: Credit rating analysis with support vector machines and neural networks: a market comparative study. Decision Support Systems 37, 543–558 (2004)
8. Kannan, S., Balakrishnan, P.: Multi-stage Modeling for Predicting Payment Delinquency (2009), http://sede.neurotech.com.br:443/PAKDD2009/files (viewed July 15, 2009)
9. Linhart, C., Abramovich, S., Harari, G., Buchris, A.: PAKDD 2009, Data Mining Competition (2009), http://sede.neurotech.com.br:443/PAKDD2009/files (view July 15, 2009)
10. Pfahringer, B.: PAKDD 2009 competition: a Weka-based Solution (2009), http://sede.neurotech.com.br:443/PAKDD2009/files (viewed July 15, 2009)
11. Reichert, A.K., Cho, C.C., Wagner, G.M.: An examination of the conceptual issues involved in developing credit-scoring models. Journal of Business and Economic Statistics 1(2), 101–114 (1983)
12. Terrell, G.R., Scott, D.W.: Variable kernel density estimation. Ann. Statist. 20, 1236–1265 (1992)
13. West, D.: Neural network credit scoring models. Computers & Operations Research 27, 1131–1152 (2000)
14. Yobas, M.B., Crook, J.N., Ross, P.: Credit scoring using neural and evolutionary techniques. IMA Journal of Management Mathematics 11, 11–125 (2000)
15. Zhang, Y., Orgun, M.A., Lin, W.Q., Baxter, R.: Mining multidimensional data via element oriented analysis. In: Proceeding of Pacific Rim International Conference on Artificial Intelligence (2008)
16. Hand, D.J.: Measuring classifier performance: a coherent alternative to the area under the ROC curve. Journal of Machine Learning 77(1) (2009)

Appendix 1. Data Description and Pre-processing

Attribute	Description	Pre-processing
ID_CLIENT:	Sequential number for the applicant	Set as the identification of the dataset
ID_SHOP	Shop code where the application has been made	Drop it as no consistent values
SEX **(A1)**	M=Male, F=Female	M=1, F=0
MARITAL_STATUS **(A2)**	S=Single C=Married D=Divorced V=Widow O=Other	S=1 C=2 D=3 V=4 O=5
AGE **(A3)**	Applicant's age	Range it from 15 to 83
QUANT_DEPENDANTS	Quantity of applicant's dependants	Drop it as most of values are zeros
EDUCATION	Applicant's education level	Drop it as all values are null
FLAG_RESIDENCIAL_PHONE **(A4)**	Y=Yes, N=No; If the applicant possesses a residential phone	Y=0, N=1
AREA_CODE_RESIDENCIAL_PHONE **(A5)**	Modified residential phone area code	Keep as the original format
PAYMENT_DAY **(A6)**	Fixed month day selected for the eventual monthly payment	Categorize by 3 days interval.
SHOP_RANK	Company's rating for the shop in commercial terms	Drop it as no consistent values
RESIDENCE_TYPE **(A7)**	P=Owned, A=Rented, C=Parents' House, O=Other	P=1, A=2, C=3, O=4
MONTHS_IN_RESIDENCE **(A8)**	Time in the current residence in months	Keep as the original format
FLAG_MOTHERS_NAME **(A9)**	Y=Yes, N=No; If the applicant had filled the mother's name in the form	Y=0, N=1
FLAG_FATHERS_NAME **(A10)**	Y=Yes, N=No; If the applicant had filled the father's name in the form	Y=0, N=1
FLAG_RESIDENCE_TOWN_WORKING_TOWN **(A11)**	Y=Yes, N=No; If the applicant works in the same town where lives	Y=0, N=1
FLAG_RESIDENCE_STATE_WORKING_STATE **(A12)**	Y=Yes, N=No; If the applicant works in the same state where lives	Y=0, N=1
MONTHS_IN_THE_JOB **(A13)**	Time in the current job in months	Keep as the original format
PROFESSION_CODE **(A14)**	Applicant's profession code	Keep as the original format
MATE_INCOME **(A15)**	Applicant's mate monthly net income in Brazilian currency (R$)	All nonzero values are flagged as 1
FLAG_RESIDENCIAL_ADDRESS POSTAL_ADDRESS **(A16)**	Y=Yes, N=No; If the applicant receives the post in the same address where lives	Y=0, N=1
FLAG_OTHER_CARD	Y=Yes, N=No; If the applicant possesses another credit or private label card	Drop it as most of values are zeros

QUANT_BANKING_ACCOUNTS	Quantity of applicant’s banking accounts	Drop it as all values are zeros
PERSONAL_REFERENCE_#1	First name of the personal reference #1 (in Portuguese)	Drop it as it is difficult to categorize
PERSONAL_REFERENCE_#2	First name of the personal reference #1 (in Portuguese)	Drop it as it is difficult to categorize
FLAG_MOBILE_PHONE	Y=Yes, N=No; If the applicant possesses a mobile phone	Drop it as all values are N
FLAG_CONTACT_PHONE	Y=Yes, N=No; If the applicant possesses a contact phone	Drop it as most of values are N
PERSONAL_NET_INCOME **(A17)**	Applicant's personal monthly net income in Brazilian currency (R$)	Categorize by R$100 interval
COD_APPLICATION_BOOTH	Booth code where application was handed in	Drop it as most of values indicate zeros
QUANT_ADDITIONAL_CARDS_IN_THE_APPLICATION **(A18)**	Quantity of additional cards asked for in the same application form	All nonzero values are flagged as 1
FLAG_CARD_INSURANCE_OPTION	Y=Yes, N=No; If the applicant asked for card insurance service	Drop it as most of values are N

A Semi-supervised Approach for Reject Inference in Credit Scoring Using SVMs

Sebastián Maldonado and Gonzalo Paredes

Department of Industrial Engineering, University of Chile
{semaldon,goparede}@ing.uchile.cl

Abstract. This paper presents a novel semi-supervised approach that determines a linear predictor using Support Vector Machines (SVMs) and incorporates information on rejected loans, assuming that the labeled data (accepted applicants) and unlabeled data (rejected applicants) are not drawn from the same distribution. We use a self-training algorithm in order to predict how likely a rejected applicant would have repaid had the applicant received credit. A modification to the self-training algorithm based on Platt's probabilistic output for SVMs is introduced. Experiments with two toy data sets; one well-known benchmark Credit Scoring data set, and one project performed for a Chilean financial institution demonstrate that our approach accomplishes the best classification performance compared to well-known reject inference alternatives and another state-of-the-art semi-supervised method for SVMs (Transductive SVM).

Keywords: Semi-supervised learning, Credit scoring, Support vector machines, Reject inference.

1 Introduction

Credit scoring needs sophisticated models to assess the risk of providing a loan to a person or a business, rejecting those who are considered too risky. Credit scoring models are used by all major banks and financial institutions because of their advantages: Their use significantly reduces loan processing costs and diminishes aggregate default costs [2].

Credit scoring models are usually developed from granted loans (*known good/ bad sample*), because complete data are only available for those accepted. However, a representative sample should be drawn from the population which applies for credit. Using a model based on only previously approved applicants can be inaccurate [22]. If the previous accept/decline decisions were made systematically, the set of accepted loans is a biased sample and not representative of the rejects (*sample bias*). A method is needed to account for cases in which the behavior is unknown. Reject inference is therefore used to infer the status of applicants who have been rejected. [7].

The logit model is considered the main classification model in Credit Scoring [23]. However, several data mining approaches have been proposed for this task [8]. The main objective of this work is to incorporate data mining techniques

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 558–571, 2010.

such as semi-supervised learning and SVMs to credit scoring in order to improve classification performance, using a biased sample of the applicants. Another objective is to compare the performance of different reject inference approaches mentioned in the literature, with semi-supervised methods such as self-learning and transductive learning.

This paper is organized as follows. In Section 2 we briefly introduce semi-supervised learning for classification. Section 3 addresses the issue of non-random sample selection in credit scoring and the advantage of reject inference. Section 4 introduces the proposed semi-supervised method based on SVMs. Experimental results using two artificial and two real-world data sets are given in Section 5. Section 6 summarizes this paper by providing its main conclusions and addresses possible future developments.

2 Semi-supervised Learning

Semi-Supervised Learning (SSL) is a technique that lies "between" supervised and unsupervised learning and promises to make a better classification by including unlabeled data. Based on the fact that obtaining labeled data is expensive or difficult, making unlabeled data cheaper to obtain in many applications [6], SSL attempts to achieve better classification performance using both labeled and unlabeled data. One of the first algorithms proposed for using unlabeled data is the self-training method [1,21]. Two other important approaches are co-training [3] and the Transductive SVM or S^3VM [14].

In the late sixties, transductive inference combined with combinatorial optimization was applied by Hartley and Rao [12] in order to maximize the likelihood of their model. In the early seventies, semi-supervised learning appeared as a solution for Fisher linear discriminant with unlabeled data. Semi-supervised learning had also been applied to more theoretical analyses in the eighties and nineties, for example, determining learning rates in an approximately correct framework (PAC) by Valiant [24] and an identifiable combination in which Castelli and Cover [4] showed that with finite unlabeled points the probability of error has an exponential convergence to the Bayes risk.

In the nineties the interest in SSL increased thanks to text classification tasks. Currently semi-supervised learning is particularly important in machine learning areas such as speech recognition, web mining and three-dimensional protein sequence problems [6]. The main algorithms of semi-supervised learning will be reviewed in the following sections.

2.1 Self-training

Self-training, also known as self-labeling or decision-directed learning, is the most common and simple SSL method. This wrapper algorithm uses the prediction of a supervised learning method to label the unlabeled data. In other words, the classifier uses its own prediction to teach itself. It starts training a separating hyperplane only with labeled data. In each step the algorithm selects a fraction of the unlabeled examples for labeling, according to a target or a decision function. Then the method adds these objects to the training set. Finally the classifier retrains itself and the process is repeated.

The self-learning algorithm is very simple and can be used as a meta-learning algorithm. Nevertheless, it relies on the goodness- of-fit of the classifier obtained, considering that mistakes reinforce themselves. Another disadvantage of self-learning is the difficulty of analyzing it in general, however there have been some studies of convergence for specific base learners. [10,11]. Self-training will be one of the semi-supervised strategies that we will use in order to improve credit scoring models.

2.2 Co-training

Co-training methods are based on three assumptions. First, there should be a natural split of variables in two subsets. Second, each subset should be sufficiently large in order to train a good classifier. Finally, the method assumes that both subsets are conditionally independent given the class.

The approach trains two different classifiers, one for each subset, using only the labeled data. Then each classification function classifies part of the unlabeled data and teaches the other classifier. Both classifiers are retrained with this new labeled data given by the other classifier (cross information) in an iterative way.

Nigam and Ghani [19] compare co-training with generative models and an EM algorithm. Their results show that co-training performs well, when the assumption of conditional independence is held. They also demonstrated that it is better to perform probabilistic labeling of the entire universe rather than considering only the most confident unlabeled data. This work also states that if there is no natural feature split of the set, an artificial split could be created by randomly dividing the feature set in two. Although this artificial split helps, the results are not as good as in the case where the split is natural.

2.3 Transductive Support Vector Machine (TSVM) or S^3VM

Transductive Support Vector Machine is an extension of standard SVM, in which only labeled data are used. The goal of TSVM is to use both labeled and unlabeled data in order to obtain the maximum margin in the linear boundary of the Reproducing Kernel Hilbert Space. Finding the exact TSVM solution is NP-hard, so great effort has been made on approximation algorithms. One of the first widely used softwares for solving this problem is SVMlight, TSVM implementation by Joachims [14].

Considering L labeled examples $\left\{(\mathbf{x_i^l}, \mathbf{y_i^l})\right\} \in \mathscr{L}$, $y_i^l \in \{-1, +1\}$, and U unlabeled examples $\{\mathbf{x_i^u}\} \in \mathscr{U}$, where $N = L + U$ and $\mathscr{L}$ and $\mathscr{U}$ represent the sets of labeled and unlabeled examples respectively. Assuming a linear model $f(\mathbf{x}) = \mathbf{w^T} \cdot \mathbf{x^l} + \mathbf{b}$ and using hinge loss for the unlabeled data, TSVM has the following formulation [27] (C_1 and C_2 are penalty parameters):

$$\underset{f}{\text{Min}} \qquad \sum_{i \in \mathscr{L}} \left(1 - y_i f(\mathbf{x_i^l})\right)_+ + C_1 \|\mathbf{w}\|^2 + C_2 \sum_{i \in \mathscr{U}} \left(1 - |f(\mathbf{x_i^u})|\right)_+ \tag{1}$$

The last term of the equation takes into account the unlabeled data. The loss function $(1 - |f(\mathbf{x_i^u})|)_+$ has a non-convex hat shape, which is the source of difficulty in the optimization problem. Some researchers proposed to solve the optimization problem using Gaussian function as an approximation of the hat loss function [15]. Other approaches attempt to solve an easier problem and then gradually transform it into the TSVM objective. Collobert et al. [5] optimized the hard TSVM directly using an approximate optimization procedure called concave-convex procedure (CCCP). As a result the authors report improvements in speed for the training of the TSVM. A global optimal solution of TSVM was proposed using Branch and Bound, where excellent accuracy for small data sets is found. Although Branch and Bound probably will not be useful for large data sets, this result shows the potential of TSVM with better approximation algorithms.

3 Reject Inference for Credit Scoring

Credit scoring models are developed to predict the behavior of all applicants but using a model based only on approved clients can be inappropriate. This is a major issue when the accept/decline decisions are made systematically and not randomly. In this case the accepted population is not representative for the rejected loans and a method is needed for cases where the behavior is unknown. Reject inference is a process that forecasts the behavior of rejected applicants based on the analysis and performance of previously rejected ones. The main reason for performing reject inference is the sample bias issue.

Reject inference can neutralize some distortions in decision-making. For example, if credit is given to a group of applicants who have historic delinquency, and they respond as good applicants, a credit scoring model (without reject inference) would probably classify a new applicant who has historic delinquency as good, based on the result of the first group. This kind of distortion could be corrected with reject inference. It is also useful to estimate level of risk in a specific unknown situation, allowing estimation of bad rates by the score of those who were previously rejected, helping decision-making processes. However, reject inference involves predicting the unknown, and will always have a degree of uncertainty. Uncertainty can be reduced using better techniques but it will never be eliminated.

Depending on the application acceptance rate and the level of confidence in previous credit-granting criteria, reject inference could have great impact on the credit scoring models. For example, with a very high level of confidence and a high approval rate, reject inference is less important. In this case all rejected can be seen as bad with a high level of confidence. If the level of confidence in the credit-granting criteria is very low, near random adjudication can be assumed and again the reject inference is not relevant. In cases with low or medium approval rates and low bad rates, reject inference helps to identify opportunities to increase market share with risk-adjusted strategies. Reject inference will also have an important impact in cases where the accept/decline decision process performs well.

Several strategies for reject inference in credit scoring have been proposed [22]. Subsection 3.1 summarizes different approaches for traditional reject inference. With the advances in data mining made in the last decade, some strategies have been developed for credit scoring and reject inference using data mining techniques, which we present in subsection 3.2.

3.1 Traditional Reject Inference Strategies

There are various techniques used to perform reject inference. Some of these are presented in the following paragraphs.

Assign All rejects to bads. This approach is not adequate in most cases, because we know that an important fraction of the rejects would have been good. The only situation where this assumption could be suitable is when the approval rates are very high and the cost of default is very high as well.

Assign rejects in the same proportion of goods to bads as reflected in the accepted loans. We can use this assignment with confidence in two situations. If there is no consistency in the current selection system or if the decisions have been made randomly.

Ignore the rejects altogether. This is the most common method. The scoring system is developed only with accepted applicants and the sample bias issue is present. Ignoring the rejects is an ineffective and inefficient alternative.

Approve All Applications for a time period. This is the only method that allows finding out the actual or real performance of rejected accounts. It is necessary to approve all applications for a specific time period, which could be a very expensive decision in terms of credit risk. The approved applications should be representative of all score ranges, so it is not acceptable to understate or overstate the bad rate of the rejects.

Use Data mining techniques to classify rejects. Based on the idea that rejected and approved applicants have different distributions, we can improve the classification performance by applying data mining techniques in order to incorporate the complete information of the applicants for prediction.

3.2 Reject Inference Using Data Mining Techniques

Some approaches have been developed for reject inference for Credit scoring. This issue has been mainly addressed within the context of logistic regression. Chen [7] proposed a maximum likelihood approach for reject inference, which is limited to logistic regression. More general approaches such as Heckman's bivariate two-stage model and the augmentation method have also been proposed. Unfortunately, empirical research of these models reveals little promise [7].

The issue of a non-random sample for the unlabeled data in semi-supervised learning has been addressed in a more general context (see, for example [27]) and in different applications, such as spam filtering [26]. An interesting approach based on Bayesian networks is proposed for biased labeling in semi-supervised learning.

Transductive SVM also allows performing reject inference with an additional parameter p, which represents the fraction of unlabeled examples to be classified into the positive class [14]. This parameter allows a correction to the classifier but does not perform reject inference since this method does not classify the unlabeled data.

4 A Semi-supervised Algorithm for Reject Inference in Credit Scoring

We propose a self-training algorithm with a modification in order to incorporate the assumption that the unlabeled data (rejected loans) have a higher risk in terms of good/bad proportion. The main idea is to train a SVM classifier using the labeled data (accepted applicants) and to estimate the probability of default for rejected loans by using a logit link function as proposed by Platt [20]. The next step is to adjust the cut-off threshold using a parameter λ and computing the confidence of each unlabeled object. We incorporate the unlabeled observations iteratively with higher confidence to the labeled data until all unlabeled data are labeled. The final classifier considers all applicants for credit and allows an unbiased prediction for the behavior of new applicants in terms of risk.

The intention behind this approach is that we can adjust the classifier by penalizing the unlabeled objects with less confidence (closer to the hyperplane), forcing some of the rejected loans labeled as "good" in the self-training process to be "bad". According to the principles of self-training, which seems to be one of the most natural strategies to follow [27], unlabeled objects with high confidence are more likely to be consistent with the classifier, so we focus on modifying the separating hyperplane using the unlabeled objects with less confidence in order to incorporate the higher risk of rejected loans and to construct a classifier based on an unbiased sample of the real population of applicants.

Formally, given training vectors $\mathbf{x}_i \in \Re^M$, $i = 1, ..., N$, which consists of L labeled examples $\left\{(\mathbf{x^l_i}, \mathbf{y^l_i})\right\} \in \mathscr{L}$, $y^l_i \in \{-1, +1\}$, and U unlabeled examples $\{\mathbf{x^u_i}\} \in \mathscr{U}$. For binary classification, SVM provides the optimal hyperplane $f(\mathbf{x}) = \mathbf{w^T} \cdot \mathbf{x^l} + \mathbf{b}$ that aims to separate the training patterns. In the case of linearly separable classes this hyperplane maximizes the sum of the distances to the closest positive and negative training patterns. This sum is called *margin*. To construct the maximum margin or optimal separating hyperplane, we need to classify correctly the vectors $\mathbf{x}^l_i$ of the training set into two different classes y^l_i, using the smallest norm of coefficients $\mathbf{w}$ [25].

If we look for a linear hyperplane in the case of linearly non-separable classes, a set of slack variables is introduced for each training vector. C is a penalty parameter on the training error. The SVM procedure aims at solving the following optimization problem:

$$\underset{\mathbf{w}, b, \xi}{\text{Min}} \qquad \frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\sum_{i=1}^{L}\xi_i \tag{2}$$

subject to

$$y_i^l \cdot (\mathbf{w}^T \cdot \mathbf{x}_i^l + b) \geq 1 - \xi_i \qquad i = 1, ..., L,$$

$$\xi_i \geq 0 \qquad i = 1, ..., L.$$

Notice that the examples that are farthest from the separating hyperplane have higher confidence and are more likely to belong to their corresponding class. The self-training algorithm for reject inference follows:

Algorithm 1. Self-Training for Reject Inference

```
while($\mathscr{U} \neq \emptyset$) do
    train a SVM classifier $f$ using (1) with all data in $\mathscr{L}$;
    use $f$ to classify all unlabeled examples in $\mathscr{U}$;
    transform $f(\mathbf{x_i^u})$ to a probabilistic outcome using Platt's logit link function:
        $P(y = 1|f(\mathbf{x_i^u})) = \frac{\mathbf{1}}{\mathbf{1 + exp(Af(x_i^u) + B)}}$    (3)
    adjust the cut-off threshold by incorporating a higher risk in unlabeled data:
        $f_\lambda(\mathbf{x_i^u}) = \mathbf{P(y = 1|f(x_i^u)) - 0.5} + \lambda$    (4)
    select $\mathbf{x}^* \in \mathscr{U}$ with highest confidence $|f_\lambda(x^*)|$;
    $\mathscr{L}$.add(($\mathbf{x}^*, \mathbf{sign}(\mathbf{f}_\lambda(\mathbf{x}^*))$));
    $\mathscr{U}$.remove($\mathbf{x}^*$);
    end while;
```

The parameters A and B in the link function (3) are fit using maximum likelihood estimation from the training set [20]. Parameter $\lambda \in [-0.5, 0.5]$ represents the relative risk of the rejected applicants in terms of the accepted examples. $\lambda = 0$ performs standard self-training, while a value of λ equal to one of the bounds will assume that all rejected examples belong to a unique class. For example, given the class +1 the defaulted loans, $P(y = 1|f(\mathbf{x_i^u}))$ represents the probability of default for each rejected loan. A negative value of λ considers a higher risk in the rejected loans, and allows rejected examples with a probability of default smaller but close to 0,5 to belong to the positive class (defaulters).

We suggest the following procedure to estimate λ: we define u^- (u^+) as the number of negative (positive) examples in the unlabeled training data, which at this point we assume to be known. We consider $\frac{u^+}{U}$ the ratio of customers that belong to the positive class in the unlabeled data. We propose to obtain λ by correcting the expected value of $P(y = 1|f(\mathbf{x_i^u}))$ using the estimated proportion of bad customers in the subset of rejected loans:

$$\lambda = \frac{u^+}{U} - \frac{\sum_{i=1}^{U} P(y = 1|f(\mathbf{x_i^u}))}{U}, \tag{5}$$

which is positive when there is a higher proportion of good customers in the labeled class (given by the expected value of the probability of default trained using labeled data) than in the unlabeled class.

The parameter λ assumes knowing the probability of being a good customer for the rejected loans, which is unknown but can be estimated by understanding the process of credit assessment, and at the end represents a strategic decision. In our experiments the real proportion of good customers in the unlabeled data will be known and we will use this information to obtain λ.

5 Experimental Results

The proposed approach has been applied to two toy data sets, one real-world credit data set from the UCI data repository [13], and one data set from a Chilean financial institution [17]. Next, these data sets are briefly described and the classification results using different reject inference methods are provided.

5.1 Experiments with Toy Data Sets

A two-dimensional data set was constructed including 3 subsets: 100 examples in a training subset with 80 negative instances (good accepted loans, represented by squares in Figure 1) and 20 positive instances (bad accepted loans, represented by diamonds in Figure 1); 100 examples in a second training subset which we consider unlabeled in order to emulate a rejected subset (circles in Figure 1), with 60 negative instances (good rejected loans, shown by squares in Figure 1) and 40 positive instances (bad rejected loans, diamonds in Figure 1). Finally we consider a test subset with 100 examples (70 negative instances and 30 positive). The training data set (labeled and unlabeled) and the test data set are drawn from the same distribution. Both variables are generated in order to be useful for the classification task. Figure 1 represents a plot of the training subset of this toy data set.

A second toy data set is obtained using the same data: Using the same 200 examples for training and the remaining 100 examples for testing, we split the training data according to one simple rule: 100 examples above a threshold using one of the variables are considered unlabeled. In this case we do not have overlapping labeled/unlabeled data sets and we also have a higher risk (12 positive instances in the labeled-training subset and 48 positive instances in the unlabeled-training set). Figure 2 represents a plotted view of the training subset of the second toy data set.

We tested our approach in both data sets and obtained the following solutions for reject inference:

- SVM using only the labeled data set and omitting the unlabeled data set (SVM).
- SVM considering all rejected loans as good loans (SVM+g).
- SVM considering all rejected loans as bad loans (SVM+b).
- SVM using standard self-training (SVM+st), which is equivalent to $\lambda = 0$.

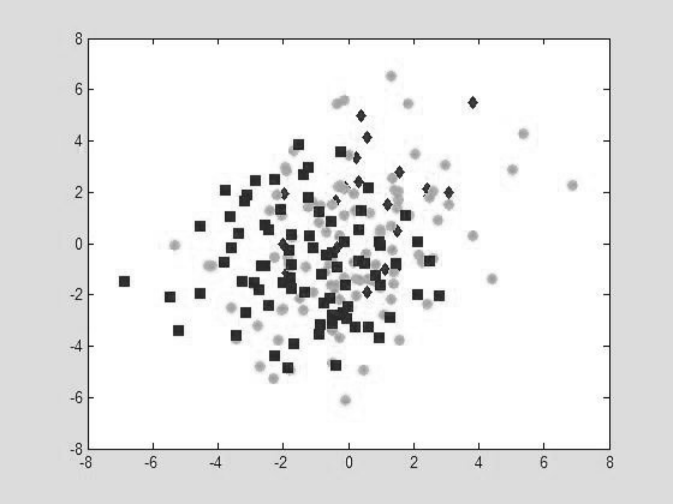


Fig. 1. Plot of the data set "Toy 1"

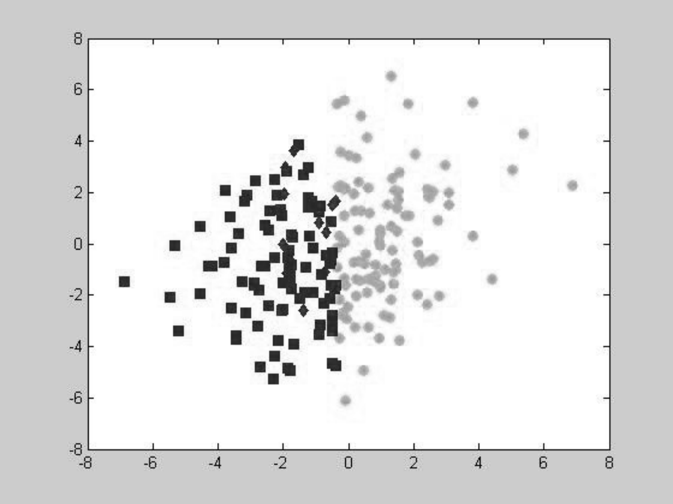


Fig. 2. Plot of the data set "Toy 2"

- SVM using the proposed modification for self-training (λ-SVM).
- SVM using Transductive SVM assuming the same proportion of classes in the labeled and unlabeled subset (TSVM).
- SVM using Transductive SVM considering the real proportion of classes in the unlabeled subset (TSVM+p).

Table 1. Classification accuracy for two "toy" data sets

	Test Toy 1	Test Toy 2
SVM	79%	70%
SVM+g	70%	70%
SVM+b	59%	72%
SVM+st	80%	70%
λ-SVM	**82**%	**76**%
TSVM	80%	75%
TSVM+p	80%	**76**%

Table 1 shows the classification performance (percentage of correctly classified examples over total examples) in the test subset for both data sets:

From these experiments we observed better classification performance with our approach. Transductive SVM performs as well as our approach in the second toy subset but it is much more time-consuming (about 3 hours versus a few seconds). We can also speed up our method to 10 iterations by incorporating 10 examples ($\frac{U}{10}$) to the labeled data set at each iteration, achieving the same classification performance.

From these results we observed that classification performance on the training set using self-training is much better than any other method, which can be tricky since the method assumes that the function used to label the rejected examples is classifying all data correctly. It is important to analyze the results only in the test subset and to avoid biased conclusions. Self-training without any inference of the unlabeled data just reinforces the classifier obtained with the labeled data and does not improve the classification.

5.2 Experiments with a Real-World Benchmark Data Set

In order to validate the results obtained with toy examples, we considered the real-world data set German Credit, which consists of 800 examples in a training subset and 200 examples in a test subset, both with approximately 70% good loans. We split the training subset into a training-labeled subset and a training-unlabeled subset using stratified sampling and selecting approximately 600 instances for the labeled subset and the remaining 200 instances for the unlabeled subset. We perform 4 different splits of the training subset in order to consider different levels of risk. Table 2 contains the information for the 4 different training subsets:

Table 2. Proportion good loans for the German Credit Data

	Gcredit1	Gcredit2	Gcredit3	Gcredit4
good loans labeled subset	70,1%	69,5%	67,6%	66,8%
good loans unlabeled subset	70,1%	70,3%	70,9%	71,2%
total good loans for training	70,1%	70,1%	70,1%	70,1%

Table 3. Classification accuracy for the German Credit Data

	Gcredit1	Gcredit2	Gcredit3	Gcredit4
SVM	78%	76%	76%	75%
SVM+g	70%	70%	73%	70%
SVM+b	58%	66%	67%	65%
SVM+st	78%	76%	77%	75%
λ-SVM	78%	77%	78%	77%
TSVM	77%	74%	76%	71%
TSVM+g	77%	74%	76%	69%

We ran the approaches described above for all 4 training subsets. Table 3 shows the classification performance in the test subset for both data sets:

Again our approach outperforms other methods in all 4 training subsets. Notice that our method behaves better in comparison with others in the training sets with a higher difference in terms of risk. When the risk is similar, all methods have similar classification performance.

5.3 Experiments with the INDAP Data Set

The INDAP data set stems from a credit scoring project performed for this Chilean organization. INDAP is the main service provided by the Chilean government that aims at supporting small agricultural enterprises; see www.indap.cl. It was founded in 1962 and has more than 100 offices all over Chile serving its more than 100,000 customers [17].

Following a feature selection step using the wrapper method HO-SVM [17], the data set is based on 21 variables describing 1,100 observations (767 good and 333 bad customers). We split the whole data set into a training data set with 770 examples and a test data set with 330 examples (both with approximately 70% good customers) using stratified sampling. From the training data set we obtained two subsets: one labeled-training subset emulating accepted loans with 539 observations and 71.6% good customers and one unlabeled-training subset with 231 examples and 65.4% good customers, which represent a sample of rejected loans and were used for reject inference. Table 4 shows the results in terms of accuracy for this data set, considering the experiments mentioned above.

Table 4. Classification accuracy for the INDAP data set

	INDAP
SVM	75%
SVM+g	70%
SVM+b	70%
SVM+st	75%
λ-SVM	**76**%
TSVM	**76**%
TSVM+g	75%

For this data set the proposed approach and TSVM perform slightly better than SVM and standard self-training, while traditional reject inference affects negatively in the performance of the classifier.

6 Conclusions

We presented a novel semi-supervised learning approach for reject inference in credit scoring using SVM. The intention behind this method is that we can correct the sample bias by labeling the rejected loans using self-learning. Although traditional self- learning focuses on the unlabeled examples with higher confidence, the important examples in our approach are the less confident ones (rejected loans which the classifier can not conclude with certainty if they would have been "bad" or "good" loans): these examples are more likely to be considered "bad" in the adequate proportion in order to adjust the classifier to the real good/bad proportion.

A comparison with other semi-supervised techniques and reject inference strategies for credit scoring shows the advantages of our approach:

- It outperforms other reject inference strategies for classification, based on its ability to reproduce the expected risk of the real credit scoring problem (the "through the door" population).
- Unlike TSVM, this approach represents an iterative algorithm based on standard SVM, avoiding a complex non-linear optimization problem and ensuring a global optimum solution.
- It can be used with any suitable Kernel function, allowing non-linear classifiers.
- It can be easily generalized to other classification methods, such as logistic regression.

The experiments performed show that the strategy of considering all rejected loans as "bad" or "good" proposed in [22] can negatively affect classification accuracy, and should be used only in very special cases. On the other hand, reject inference based on data mining strategies, such as self-training and transductive algorithms, can improve the classification task by adjusting the expected risk of the unbiased sample of loans. The significance of this improvement is shown by the consistency of the current credit system: if accepted and rejected loans differ significantly in terms of good/bad proportion, reject inference using the proposed method helps to obtain a better solution by incorporating all available data.

Our algorithm relies on an iterative optimization problem, which is computationally treatable but expensive if the number of input features is large. We could improve its performance by applying filter methods for feature selection before running the algorithm [17]. In this way we can identify and remove irrelevant features at low cost. In several credit scoring projects we have performed for Chilean financial institutions we used univariate analysis (Chi-Square Test for categorical features and the Kolmogorov-Smirnov Test for continuous ones) as a first filter for features selection with excellent results [17].

Future work has to be done in various directions. First, it would be interesting to improve the proposed technique by incorporating the information of the rules that generated the current scoring model in order to improve the inference. It would be possible to adjust the original score model by moving the rules according to the performance of the classification and incorporating reject inference based on semi-supervised learning. If the original model is not built on the basis of a set of rules, we can extract them using different rule extraction techniques for classification methods [18]. Also interesting is the application of this approach in the domain of spam filtering, where many semi-supervised approaches have been developed in order to improve the classification performance, considering that labeled cases are previously defined by spam filters.

Acknowledgments. Support from the Chilean "Instituto Sistemas Complejos de Ingeniería" (ICM: P-05-004-F, CONICYT: FBO16) is greatly acknowledged (www.sistemasdeingenieria.cl). The first author also acknowledges a grant provided by CONICYT for a grant it provided for his Ph.D. studies in Engineering Systems.

References

1. Agrawala, A.K.: Learning with a probabilistic teacher. IEEE Transactions on Information Theory 16, 373–379 (1970)
2. Berger, A.N., Frame, W.S., Miller, N.H.: Credit scoring and the availability, price, and risk of small business credit. Journal of Money, Credit and Banking 37(2), 191–222 (2005)
3. Blum, M.T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, pp. 92–100 (1998)
4. Castelli, V., Cover, T.M.: On the exponential value of labeled samples. Pattern Recognition Letters 16, 105–111 (1995)
5. Chapelle, O., Zien, A.: Semi-supervised classification by low density separation. In: Proceeding of the Tenth International Workshop on Artificial Intelligence and Statistic (AISTAT 2005) (2005)
6. Chapelle, O., Scholkopf, B., Zien, A.: Semi-Supervised Learning. MIT Press, Cambridge (2005)
7. Chen, G., Astebro, T.: A Maximum Likelihood Approach for Reject Inference in Credit scoring. Rotman School of Management Working Paper No. 07-05 (2006)
8. Chye, K.H., Chin, T.W., Peng, G.C.: Credit scoring using data mining techniques. Singapore Management Review 26(2), 25(23) (2004)
9. Collobert, R., Weston, J., Bottou, L.: Trading convexity for scalability. In: ICML 2006, 23rd International Conference on Machine Learning, Pittsburgh, USA (2006)
10. Culp, M., Michailidis, G.: An iterative algorithm for extending learners to a semisupervised setting. In: The 2007 Joint Statistical Meetings (2007)
11. Haffari, G., Sarkar, A.: Analysis of semi-supervised learning with the Yarowsky algorithm. In: 23rd Conference on Uncertainty in Artificial Intelligence (2007)
12. Hartley, H.O., Rao, J.N.K.: Classification and estimation in analysis of variance problems. Review of the International Statistical Institute 36, 141–147 (1968)

13. Hettich, S., Bay, S.D.: The UCI KDD Archive. University of California, Department of Information and Computer Science, Irvine, CA (1999), http://kdd.ics.uci.edu
14. Joachims, T.: Transductive Inference for Text Classification using Support Vector Machines. In: International Conference on Machine Learning, pp. 200–209 (1999)
15. Johnson, R., Zhang, T.: Two-view feature generation model for semi-supervised learning. In: The 24th International Conference on Machine Learning, pp. 25–27 (2007)
16. Maeireizo, B., Litman, D., Hwa, R.: Co-training for predicting emotions with spoken dialogue dat. In: The Companion Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics, ACL (2004)
17. Maldonado, S., Weber, R.: A wrapper method for feature selection using Support Vector Machines. Information Sciences 179(13), 2208–2217 (2009)
18. Martens, D., Baesens, B., Van Gestel, T., Vanthienen, J.: Comprehensible credit scoring models using rule extraction from Support Vector Machines. European Journal of Operational Research 183(3), 1466–1476 (2007)
19. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Ninth International Conference on Information and Knowledge Management, pp. 86–93 (2000)
20. Platt, J.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In: Advances in Large Margin Classifiers, pp. 61–74. MIT Press, Cambridge (1999)
21. Scudder, H.J.: Probability of error of some adaptive pattern-recognition machines. IEEE Transactions on Information Theory 11, 363–371 (1965)
22. Siddiqi, N.: Credit Risk Scorecards, Developing and Implementing Intelligent Credit scoring, 1st edn. Wiley & Sons, Chichester (2005)
23. Thomas, L.C.: A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. International Journal of Forecasting 16(2), 149–162 (2002)
24. Valiant, L.G.: A theory of the learnable. Commun. ACM 27(11), 1134–1142 (1984)
25. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
26. Xu, J.-M., Fumera, G., Roli, F., Zhou, Z.-H.: Training SpamAssassin with active semi-supervised learning. In: Proceedings of the 6th Conference on Email and Anti-Spam (CEAS 2009), Mountain View, CA (2009)
27. Zhu, X.: Semi-Supervised Learning Literature Survey. Computer Sciences TR 1530, University of Wisconsin, Madison (2007)

Data Mining with Neural Networks and Support Vector Machines Using the R/rminer Tool*

Paulo Cortez

Department of Information Systems/R&D Centre Algoritmi,
University of Minho, 4800-058 Guimarães, Portugal
pcortez@dsi.uminho.pt
http://www3.dsi.uminho.pt/pcortez

Abstract. We present rminer, our open source library for the R tool that facilitates the use of data mining (DM) algorithms, such as neural Networks (NNs) and support vector machines (SVMs), in classification and regression tasks. Tutorial examples with real-world problems (i.e. satellite image analysis and prediction of car prices) were used to demonstrate the rminer capabilities and NN/SVM advantages. Additional experiments were also held to test the rminer predictive capabilities, revealing competitive performances.

Keywords: Classification, Regression, Sensitivity Analysis, Neural Networks, Support Vector Machines.

1 Introduction

The fields of data mining (DM)/business intelligence (BI) arose due to the advances of information technology (IT), leading to an exponential growth of business and scientific databases. The aim of DM/BI is to analyze raw data and extract high-level knowledge for the domain user or decision-maker [16].

Due to its importance, there is a wide range of commercial and free DM/BI tools [7]. The R environment [12] is an open source, multiple platform (e.g. *Windows*, *Linux*, *Mac OS*) and high-level matrix programming language for statistical and data analysis. Although not specifically oriented for DM/BI, the R tool includes a high variety of DM algorithms and it is currently used by a large number of DM/BI analysts. For example, the 2008 DM survey [13] reported an increase in the R usage, with 36% of the responses [13]. Also, the 2009 KDnuggets pool, regarding DM tools used for a real project, ranked R as the second most used open source tool and sixth one overall [10]. When compared with commercial tools (e.g. offered by SAS: http://www.sas.com/technologies/bi/) or even open source environments (e.g. WEKA [18]), R presents the advantage of being more flexible and extensible by design, thus integration of statistics, programming and graphics is more natural. Also, due to its open source availability and users' activity, novel DM methods are in general more quickly encoded into

* This work is supported by FCT grant PTDC/EIA/64541/2006.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 572–583, 2010.

R than into commercial tools. The R community is very active and new packages are being continuously created, with more than 2321 packages available at http://www.r-project.org/. Thus, R can be viewed as worldwide gateway for sharing computational algorithms.

DM software suites often present friendly graphical user interfaces (GUI). In contrast, the most common usage of R is under a console command interface, which may require a higher learning curve from the user. Yet, after mastering the R environment, the user achieves a better control (e.g. adaptation to a specific application) and understanding of what is being executed (in contrast with several "black-box" DM GUI products). Nevertheless, for those interested in graphical DM suites for R, there is the Rattle tool [17].

In this work, we present our rminer library, which is an integrated framework that uses a console based approach and that facilitates the use of DM algorithms in R. In particular, it addresses two important and common goals [16]:

classification – labeling a data item into one of several predefined classes; and
regression – estimate a real-value (the dependent variable) from several (independent) input attributes.

While several DM algorithms are available for these tasks, the library is particularly suited for using neural networks (NNs) and support vector machines (SVMs). Both are flexible models that can cope with complex nonlinear mappings, potentially leading to more accurate predictions [8]. Also, it is possible to extract knowledge from NNs and SVMs, given in terms of input relevance [4]. When compared to Rattle, rminer can be viewed as a lightweight command based alternative, since it is easier to install and requires much less R packages. Moreover, rminer presents more NN and SVM capabilities (e.g. in Rattle version 2.5.26, SVM cannot be used for regression tasks). While adopting R packages for the DM algorithms, rminer provides new features:

i) it simplifies the use of DM algorithms (e.g. NNs and SVMs) in classification and regression tasks by presenting a short and coherent set of functions (as shown in Section 3.1);
ii) it performs an automatic model selection (i.e. tuning of NN/SVM);
iii) it computes several classification/regression metrics and graphics, including the sensitivity analysis procedure for input relevance extraction.

The rminer/R tool has been used by both IT and non-IT specialists (e.g. managers, biologists or civil engineers), with applications in distinct domains, such as civil engineering [15], wine quality [4] or spam email detection [5]. In this paper, we address several real-world problems from the UCI repository [1] to show the rminer capabilities.

2 Data Mining

DM is an iterative process that consists of several steps. The CRISP-DM [2], a tool-neutral methodology supported by the industry (e.g. SPSS, DaimlerChryslyer), partitions a DM project into 6 phases (Fig. 1): 1 - business understanding;

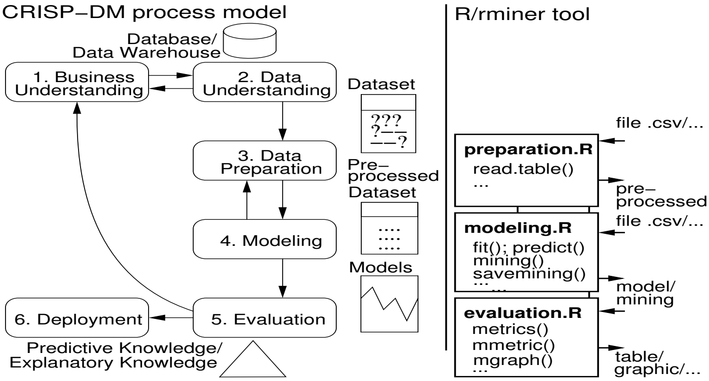


Fig. 1. The CRISP-DM and proposed R/rminer tool use

2 - data understanding; 3 - data preparation; 4 - modeling; 5 - evaluation; and 6 - deployment.

This work addresses steps 4 and 5, with an emphasis on the use of NNs and SVMs to solve classification and regression goals. Both tasks require a supervised learning, where a model is adjusted to a dataset of examples that map I inputs into a given target. The rminer models output a probability $p(c)$ for each possible class c, such that $\sum_{c=1}^{N_c} p(c)$ =1 (if classification) or a numeric value (for regression). For assigning a target class c, one option is to set a decision threshold $D \in [0, 1]$ and then output c if $p(c) > D$, otherwise return $\neg c$. This method is used to build the receiver operating characteristic (ROC) curves. Another option is to output the class with the highest probability and this method allows the definition of a multi-class confusion matrix.

To evaluate a model, common metrics are [18]: ROC area (AUC), confusion matrix, accuracy (ACC), true positive/negative rates (TPR/TNR), for classification; and mean absolute deviation (MAD), relative absolute error (RAE), root mean squared (RMSE), root relative squared error (RRSE) and regression error characteristic (REC) curve, for regression. A classifier should present high values of ACC, TPR, TNR and AUC, while a regressor should present low predictive errors and an high REC area. The model's generalization performance is often estimated by the holdout validation (i.e. train/test split) or the more robust k-fold cross-validation [8]. The latter is more robust but requires around k times more computation, since k models are fitted.

Before fitting the DM models, the data needs to be preprocessed. This includes operations such as selecting the data (e.g. attributes or examples) or dealing with missing values. Since functional models (e.g. NN or SVM) only deal with numeric values, discrete variables need to be transformed. In R/rminer, the nominal attributes (with N_c = 3 or more non-ordered values) are encoded with the common 1-of-N_c transform, leading to N_c binary variables. Also, all attributes are standardized to a zero mean and one standard deviation [8].

For NN, we adopt the popular multilayer perceptron, as coded in the R **nnet** package. This network includes one hidden layer of H neurons with logistic functions (Fig 2). The overall model is given in the form:

$$y_i = f_i(w_{i,0} + \sum_{j=I+1}^{I+H} f_j(\sum_{n=1}^{I} x_n w_{m,n} + w_{m,0}) w_{i,n}) \tag{1}$$

where y_i is the output of the network for node i, $w_{i,j}$ is the weight of the connection from node j to i and f_j is the activation function for node j. For a binary classification ($N_c = 2$), there is one output neuron with a logistic function. Under multi-class tasks ($N_c > 2$), there are N_c linear output neurons and the softmax function is used to transform these outputs into class probabilities:

$$p(i) = \frac{exp(y_i)}{\sum_{c=1}^{N_c} exp(y_c)} \tag{2}$$

where $p(i)$ is the predicted probability and y_i is the NN output for class i. In regression, the output neuron uses a linear function. The training (BFGS algorithm) is stopped when the error slope approaches zero or after a maximum of M_e epochs. For regression tasks, the algorithm minimizes the squared error, while for classification it maximizes the likelihood [8]. Since NN training is not optimal, the final solution is dependent of the choice of starting weights. To solve this issue, the solution adopted is to train N_r different networks and then select the NN with the lowest error or use an ensemble of all NNs and output the average of the individual predictions [8]. In rminer, the former option is set using `model="mlp"`, while the latter is called using `model="mlpe"`. In general, ensembles are better than individual learners [14]. The final NN performance depends crucially on the number of hidden nodes. The simplest NN has $H = 0$, while more complex NNs use a high H value.

When compared with NNs, SVMs present theoretical advantages, such as the absence of local minima in the learning phase [8]. The basic idea is transform the input $\mathbf{x} \in \Re^I$ into a high m-dimensional feature space by using a nonlinear mapping. Then, the SVM finds the best linear separating hyperplane, related to a set of support vector points, in the feature space (Fig. 2). The transformation ($\phi(\mathbf{x})$) depends of a kernel function. In rminer, we use the **kernlab** package, which uses the sequential minimal optimization (SMO) learning algorithm. We also adopt the popular gaussian kernel, which presents less parameters than other kernels (e.g. polynomial): $K(\mathbf{x}, \mathbf{x}') = exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$, $\gamma > 0$.

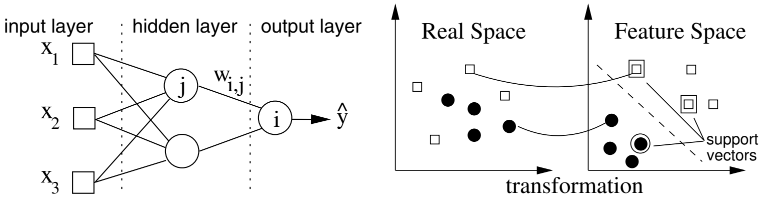


Fig. 2. Example of a multilayer perceptron (left) and SVM transformation (right)

The classification performance is affected by two hyperparameters: γ, the parameter of the kernel, and C, a penalty parameter. The probabilistic SVM output is given by [19]:

$$\begin{aligned} f(\mathbf{x}_i) &= \textstyle\sum_{j=1}^{m} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \\ p(i) &= 1/(1 + exp(Af(\mathbf{x}_i) + B)) \end{aligned} \tag{3}$$

where m is the number of support vectors, $y_i \in \{-1, 1\}$ is the output for a binary classification, b and α_j are coefficients of the model, and A and B are determined by solving a regularized maximum likelihood problem. When $N_c > 2$, the one-against-one approach is used, which trains $N_c(N_c-1)/2$ binary classifiers and the output is given by a pairwise coupling [19]. For regression there is an additional hyperparameter ϵ, used to set an ϵ-insensitive tube around the residuals, being the tiny errors within this tube discarded. The SVM algorithm finds the best linear separating hyperplane:

$$y_j = w_0 + \sum_{i=1}^{m} w_i \phi_i(x) \tag{4}$$

Since the search space for these parameters is high, we adopt by default the heuristics [3]: $C = 3$ (for a standardized output) and $\epsilon = 3\sigma_y\sqrt{\log(N)/N}$, where σ_y denotes the standard deviation of the predictions of given by a 3-nearest neighbor and N is the dataset size.

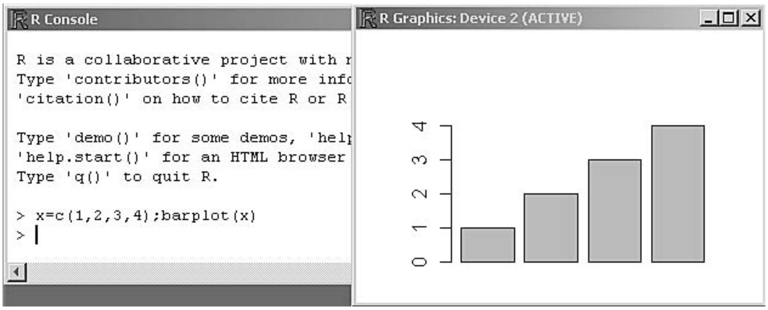
In rminer, the NN and SVM hyperparameters (e.g. H, γ) are optimized using a grid search. To avoid overfitting, the training data is further divided into training and validation sets (holdout) or an internal k-fold is used. After selecting the best parameter, the model is retrained with all training data.

The sensitivity analysis is a simple procedure that is applied after the training procedure and analyzes the model responses when a given input is changed. Let $y_{a,j}$ denote the output obtained by holding all input variables at their average values except x_a, which varies through its entire range ($x_{a,j}$, with $j \in \{1, \ldots, L\}$ levels). We use the variance (V_a) of $y_{a,j}$ as a measure of input relevance [9]. If $N_c > 2$ (multi-class), we set it as the sum of the variances for each output class probability ($(p(c)_{a,j}$). A high variance (V_a) suggests a high x_a relevance, thus the input relative importance (R_a) is given by $R_a = V_a / \sum_{i=1}^{I} V_i \times 100$ (%). For a more detailed analysis, we propose the variable effect characteristic (VEC) curve [6], which plots the $x_{a,j}$ values (x-axis) versus the $y_{a,j}$ predictions (y-axis).

3 Data Mining Using R/rminer

3.1 The R/rminer Tool

R works under a console interface (Fig. 3). Commands are typed after the prompt (>). An extensive help system is included (`help.start()` calls the full tutorial in an HTML browser). R instructions can be separated using the ; or newline



```
R is a collaborative project with r
Type 'contributors()' for more info
'citation()' on how to cite R or R

Type 'demo()' for some demos, 'hel
'help.start()' for an HTML browser
Type 'q()' to quit R.

> x=c(1,2,3,4);barplot(x)
> |
```

Fig. 3. Example of the R tool in *Windows*

character. Everything that appears after the # character in a line is a comment. R commands can be edited in a file[1] and loaded with the `source` command.

Data is stored in objects and the = operator can be used to assign an object to a variable. Atomic objects include the **character** (e.g. `"day"`) and **numeric** (e.g. `0.2`) types. There are also several containers, such as: **vector**, **factor**, **matrix**, **data.frame** and **list**. **Vectors** and **matrices** are indexed objects of atoms. A **factor** is a special vector that contains discrete values. A **data.frame** is a special matrix where the columns (vectors or factors) have names. Finally, a **list** is a collection of distinct objects (called components). Since R uses an object oriented language, there are important functions that can be applied to any of these objects (e.g. `summary` or `plot`).

Our rminer library[2] project started in 2006. All code is written in R and only a few packages need to be installed (e.g. **kernlab**). In this work, the rminer functions are underlined. The main functions are: fit – create and adjust a given DM model using a dataset (i.e. **data.frame**); predict – returns the predictions for new data; mining – a powerful function that trains and tests a particular model under several runs; mgraph, metrics and mmetric– which return several mining graphs (e.g. ROC) or metrics (e.g. ACC). All experiments were tested in *Windows*, *Linux* and *Mac OS*. The results reported here were conducted within a *Mac OS* Intel Core 2 Duo processor.

3.2 Classification Example

The satellite data was generated using **Landsat** multi-spectral images. The aim is to classify a tiny scene based on 36 numeric features that correspond to pixels from four spectral bands. In the original data, a numeric value was given to the output variable (`V37`). Also, the training and test sets are already divided into two files: `sat.trn` (with 4435 samples) and `sat.tst` (2000 cases).

We propose that a DM process should be divided into 3 blocks (or files), with the CRISP-DM steps 3 to 5 (Fig. 1): preparation, modeling and evaluation (we

[1] In *Windows*, the Tinn-R editor can be used: http://www.sciviews.org/Tinn-R/

[2] Available at: http://www3.dsi.uminho.pt/pcortez/rminer.html

only address the last two here). By separating the computation and generating intermediate outcomes, it is possible to later rerun only one of these steps (e.g. analyze a different metric), thus saving time. Our satellite modeling code is:

```
library(rminer) # load the library
# read the training and test sets:
tr=read.table("sat.trn",sep=" "); ts=read.table("sat.tst",sep=" ")
tr$V37=factor(tr$V37); ts$V37=factor(ts$V37) # convert output to factor
DT=fit(V37~.,tr,model="dt") # fit a Decision Tree with tr
NN=fit(V37~.,tr,model="mlp",search=10) # fit a NN with H=10
SV=fit(V37~.,tr,model="svm",search=2^c(-5,-3)) # fit the SVM
print(DT); print(NN); print(SV) # show and save the trained DM models:
savemodel(DT,"sat.dt"); savemodel(NN,"sat.nn"); savemodel(SV,"sat.sv")
# get the predictions:
PDT=predict(DT,ts); PNN=predict(NN,ts); PSV=predict(SV,ts)
P=data.frame(ts=ts$V37,dt=PDT,nn=PNN,svm=PSV) # create a data.frame
write.table(P,"sat.res",row.names=FALSE) # save output and predictions
```

The read.table and write.table are functions that load/save a dataset from/to a text file (e.g. `".csv"`)[3]. The tr and ts objects are **data.frames**. Since the output target (`V37`) is encoded with numeric values, we converted it into a factor (i.e. set of classes). While rminer includes several classifiers, in the example we tested only a decision tree (DT) (`model="dt"`), a NN (`"mlp"`) and a SVM (`"svm"`). The first parameter of the fit function is a R **formula**, which defines the output (`V37`) to be modeled (`~`) from the inputs (`.` means all other variables). The `search` parameter controls the NN and SVM hyperparameters (H or γ). When `search` contains more than one value (e.g. `SV` fit), then an internal grid search is performed. By default, `search` is set to $H = I/2$ for NN and $\gamma = 2^{-6}$ for SVM. Additional NN/SVM parameters can be set with the optional `mpar` (see Section 3.3). In this case, the default `mpar=c(3,100,"holdout",2/3,"AUC")` (for NN, $N_r = 3$, $M_e = 100$ and internal holdout with 2/3 train and 1/3 test split, while `"AUC"` means use the AUC metric for model selection during the grid search) or `mpar=c(NA,NA,"holdout",2/3,"AUC")` (for SVM, use default C/ϵ heuristics) is assumed. The result of the fit function is a **model** object, which contains the adjusted model (e.g. `DT@object`) and other information (e.g. hyperparameters or fitting time). The execution times (in seconds) were $1.1s$ for DT (stored at `DT@time`), $15.9s$ for NN and $25s$ for SVM. In case of the SVM, the best γ is 2^{-3} (`SV@mpar`). Next, we show the evaluation code:

```
P=read.table("sat.res",header=TRUE); P$ts=factor(P$ts); # read the results
# compute the test errors:
EDT=metrics(P$ts,P[,2:7]); ESV=metrics(P$ts,P[,14:19]);
ENN1=metrics(P$ts,P[,8:13]); ENN2=metrics(P$ts,P[,8:14],D=0.7,TC=4)
# show the full test errors:
print(EDT); print(ESV); print(ENN1); print(ENN2)
mgraph(P$ts,P[,8:13],graph="ROC",PDF="roc4",TC=4) # plot the ROC
NN=loadmodel("sat.nn") # load the best model
```

[3] Further loading functions (e.g. for SPSS files) are available in the **foreign** R package.

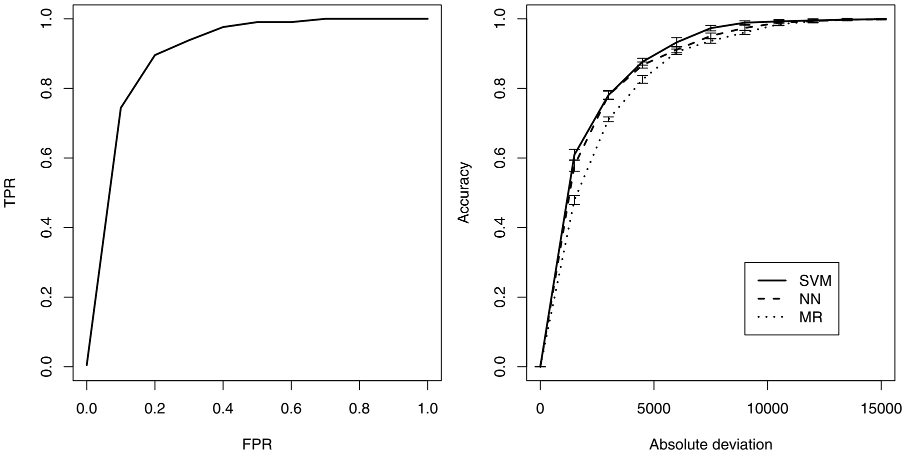


Fig. 4. Examples of the ROC (left) and REC (right) curves

The predictions for each model are matrixes, where each column denotes the p_c for a given $c \in \{“1”, “2”, “3”, “4”, “5”, “7”\}$ (there is no “6” class). The metrics function receives the target and predictions (e.g. `P` columns 8 to 13 for NN) and returns a list with several performance measures. Under the multi-class confusion matrix, the best accuracy result is given by NN, with an $ACC = 86\%$ (`ENN$acc`), followed by the SVM (81%) and the DT (78%). The global AUC, which weights the AUC for each class c according to the prevalence of c in the data [11], also favors the NN, with a value of 98% (`ENN1$tauc`). For the target “4” class (`TC=4`) and NN, we also computed the metrics using a threshold of ($D = 0.7$), leading to the $TPR = 41\%$ and $TNR = 98\%$ values (`ENN2$tpr` and `ENN2$tnr`). This is a point of the ROC curve, whose full graph is created with the mgraph command (Fig. 4).

3.3 Regression Example

The **automobile** dataset goal is to estimate car prices using 16 continuous and 10 nominal attributes. The data includes 205 instances, although there are several missing values. We tested a multiple regression (MR), a NN and a SVM, during the modeling phase:

```
library(rminer) # load the library
d=read.table("imports-85.data",sep=",",na.strings="?") # load the data
d=d[,c(6:8,14,17,19,22,26)] #  variable selection: 6,7,8,14,17,19,22,26
d=na.omit(d) # erases from d all examples with missing data
v=c("kfold",5) # external 5-fold validation
MR=mining(V26~.,d,model="mr",Runs=10,method=v) # 10 runs of 5-fold
m=c(3,100,"kfold",4,"RAE"); s=seq(1,8,1) # m=Nr,Me,... s=1,2,...,8
NN=mining(V26~.,d,model="mlpe",Runs=10,method=v,mpar=m,search=s,feat="s")
m=c(NA,NA,"kfold",4,"RAE"); s=2^seq(-15,3,2) # NA = C/epsilon heuristics
```

```
SV=mining(V26~.,d,model="svm",Runs=10,method=v,mpar=m,search=s,feat="s")
print(MR);print(NN);print(SV) # show mining results and save them:
savemining(MR,"imr"); savemining(NN,"inn"); savemining(SV,"isv")
```

Here, we selected only 7 variables as inputs (e.g. `V8`, the curb weight). Then, we deleted all examples with missing data. Next, the DM models were evaluated using 10 runs of a 5-fold cross validation scheme. The mining function performs several fits and returns a list with the obtained predictions and other fields (e.g. time for each run). The NN and SVM models were optimized (i.e. grid search for the best H and γ parameters) using an internal 4-fold. In this example and for NN, we used an ensemble of 3 networks (`model="mlpe"`). The `seq(from,to,by)` R function was used to define the search ranges for H and γ, while the `feat="s"` argument triggers the input sensitivity analysis. Next, we show the evaluation:

```
MR=loadmining("imr");NN=loadmining("inn");SV=loadmining("isv")
# show paired t-test and RAE mean and confidence intervals for SV:
print(t.test(mmetric(NN,metric="RAE"),mmetric(SV,metric="RAE")))
print(meanint(mmetric(SV,metric="RAE")))
# plot the average REC curves:
M=vector("list",3); # vector list of mining
M[[1]]=SV;M[[2]]=NN;M[[3]]=MR
mgraph(M,graph="REC",leg=c("SVM","NN","MR"),xval=15000,PDF="rec")
# plot the input relevance bars for SVM: (xval is the L x-axis position)
L=c("n-doors","body-style","drive-wheels","curb-weight","engine-size",
"bore","horsepower") # plot the input relevance (IMP) graph:
mgraph(SV,graph="IMP",leg=L,xval=0.3,PDF="imp")
# plot the VEC curve for the most relevant input (xval=4):
mgraph(SV,graph="VEC",leg=L,xval=4,PDF="vec")
```

In this example, the SVM model (median $\gamma = 2^{-3}$, $C = 3$ and $\epsilon = 0.09$, total execution time 148s) obtained the best predictive results. The average $\overline{RAE} = 32.5\% \pm 1.4$ is better when statistically compared with NN (median $H = 3$, $\overline{RAE} = 35.5\% \pm 1.4$) and MR ($41\% \pm 1.0$). In all graphs, whiskers denote the 95% t-student confidence intervals. The first mgraph function plots the vertically averaged REC curves (x-axis from 0 to 15000, Fig. 4) and confirms the SVM performance superiority. The next two graphs are based on the sensitivity analysis procedure and are useful for knowledge discovery. In this case, the relative input importances of the SVM model (ordered by importance, Fig. 5) show the curb weight as the most relevant input. The average VEC curve was plotted for this input (Fig. 5), showing a positive effect, where an increase of the curb weight leads to an higher price, particularly within the range [2519,3550].

3.4 Predictive Performance

We selected 6 classification and 6 regression tasks from UCI [1] for a more detailed predictive performance measurement of the R/rminer capabilities. The aim is show that the R/rminer results are consistent when compared with other DM tools. For a baseline comparison, we adopted the WEKA environment with its default parameters [18]. The datasets main characteristics (e.g number of

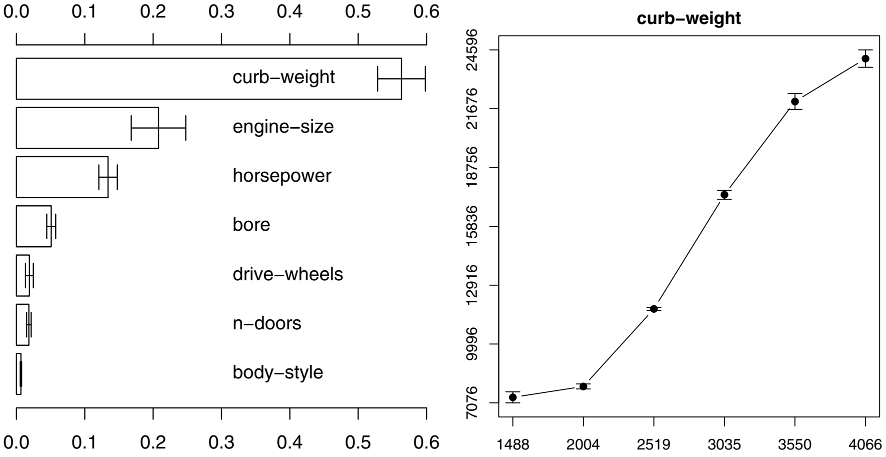


Fig. 5. The input relevances (left) and curb-weight VEC curve (right)

Table 1. Summary of the UCI datasets used

Task	Description	I	Examples	N_c
balance	balance scale weight and distance	4	625	3
cmc	contraceptive method choice	9	1473	3
german	German credit data	20	1000	2
heart	Statlog heart disease	13	270	2
house-votes	congressional voting records	16	435	2
sonar	sonar classification (rocks vs mines)	60	208	2
abalone	age of abalone	8	4177	$\Re$
auto-mpg	miles per gallon prediction	7	392	$\Re$
concrete	concrete compressive strength	8	1030	$\Re$
housing	housing prices in suburbs of Boston	13	506	$\Re$
servo	rise time of a servomechanism	4	167	$\Re$
white	white wine quality	11	4899	$\Re$

inputs, examples and classes) are shown in Table 1. For auto-mpg, all examples with missing data were removed (we used the R `na.omit` function).

For each task, we executed 10 runs of a 5-fold validation. The NN and SVM hyperparameters were ranged within $H \in \{0, 1, 2, \ldots, 9\}$ and $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{3}\}$, in a total of 10 searches per DM model. For NN, we tested the ensemble variant with N_r=3. An internal 3-fold was used during the grid search, which optimized the global AUC (classification) and RRSE (regression) metrics (the code used is available at the rminer Web page).

Table 2 presents the test set results. In general, the R/rminer outperformed the baseline tool (the only exceptions are for NN and the house-votes and sonar tasks). In particular, a higher improvement was achieved for SVM, when compared with the WEKA SVM version, with differences ranging from 3.5 pp

Table 2. Classification and regression test set results (average global AUC and RRSE values, in %; best values are in **bold**; underline denotes significant difference under a paired t-test between R/rminer and WEKA)

	WEKA		R/rminer	
Task	**NN**	**SVM**	**NN**	**SVM**
balance	97.5±0.2	88.1±0.2	**99.5**±0.1	98.9±0.2
cmc	71.4±0.3	63.8±0.3	**73.9**±0.0	72.9±0.2
german	73.5±0.7	67.2±0.7	76.3±0.8	**77.9**±0.5
heart	85.6±1.2	83.7±0.4	88.5±1.4	**90.2**±0.4
house-votes	98.6±0.2	95.7±0.3	98.0±0.5	**99.2**±0.1
sonar	89.2±1.3	76.6±1.8	87.4±0.9	**95.6**±0.8
abalone	72.7±2.1	69.9±0.1	**64.0**±0.1	66.0±0.1
auto-mpg	44.3±3.4	44.5±0.3	37.4±3.1	**34.8**±0.4
concrete	46.5±1.4	65.6±0.2	**31.8**±0.4	35.9±0.5
housing	49.9±3.0	55.2±0.4	**38.3**±1.6	40.1±1.3
servo	46.1±4.5	84.0±0.4	**40.8**±6.0	44.9±1.5
white	91.3±3.1	85.5±0.1	79.0±0.3	**76.1**±0.6

(house-votes) to 39.1 pp (servo). When comparing the two rminer methods, SVM outperforms NN in 4 classification cases, while NN is better in 4 regression datasets.

4 Conclusions

In this work, we present our rminer library, which eases the use of the R tool (e.g. for non-IT specialists) to solve DM supervised tasks. The library is particularly suited for NNs and SVMs, flexible and nonlinear learning techniques that are promising due to their predictive performances. Two tutorial examples (e.g. satellite image classification) were used to show the R/rminer potential under the CRISP-DM methodology. Additional experiments were held in order to measure the rminer library predictive performances. Overall, competitive results were obtained, in particular the SVM model for the classification tasks and NN for the regression ones. In future work, we intend to expand the rminer capabilities (e.g. unsupervised learning) and applications (e.g. telecommunications).

References

1. Asuncion, A., Newman, D.: UCI Machine Learning Repository, Univ. of California, Irvine (2007), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., Wirth, R.: CRISP-DM 1.0: Step-by-step data mining guide. CRISP-DM consortium (2000)
3. Cherkassy, V., Ma, Y.: Practical Selection of SVM Parameters and Noise Estimation for SVM Regression. Neural Networks 17(1), 113–126 (2004)

4. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems 47(4), 547–553 (2009)
5. Cortez, P., Lopes, C., Sousa, P., Rocha, M., Rio, M.: Symbiotic Data Mining for Personalized Spam Filtering. In: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI 2009), pp. 149–156. IEEE, Los Alamitos (2009)
6. Cortez, P., Teixeira, J., Cerdeira, A., Almeida, F., Matos, T., Reis, J.: Using data mining for wine quality assessment. In: Gama, J., Costa, V.S., Jorge, A.M., Brazdil, P.B. (eds.) DS 2009. LNCS, vol. 5808, pp. 66–79. Springer, Heidelberg (2009)
7. Goebel, M., Gruenwald, L.: A Survey of Data Mining and Knowledge Discovery Software Tools. SIGKDD Explorations 1(1), 20–33 (1999)
8. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, NY (2008)
9. Kewley, R., Embrechts, M., Breneman, C.: Data Strip Mining for the Virtual Design of Pharmaceuticals with Neural Networks. IEEE Trans. Neural Networks 11(3), 668–679 (2000)
10. Piatetsky-Shapiro, G.: Data Mining Tools Used Poll (2009), http://www.kdnuggets.com/polls/2009/data-mining-tools-used.htm
11. Provost, F., Domingos, P.: Tree Induction for Probability-Based Ranking. Machine Learning 52(3), 199–215 (2003)
12. R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria (2009), ISBN 3-900051-00-3 http://www.R-project.org
13. Rexer, K.: Second annual data miner survey. Technical report, Rexer Analytics (2008)
14. Rocha, M., Cortez, P., Neves, J.: Evolution of Neural Networks for Classification and Regression. Neurocomputing 70, 2809–2816 (2007)
15. Tinoco, J., Correia, A.G., Cortez, P.: A Data Mining Approach for Jet Grouting Uniaxial Compressive Strength Prediction. In: World Congress on Nature and Biologically Inspired Computing (NaBIC 2009), Coimbatore, India, December 2009, pp. 553–558. IEEE, Los Alamitos (2009)
16. Turban, E., Sharda, R., Aronson, J., King, D.: Business Intelligence, A Managerial Approach. Prentice-Hall, Englewood Cliffs (2007)
17. Williams, G.: Rattle: A Data Mining GUI for R. The R Journal 1(2), 45–55 (2009)
18. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, 2nd edn. Morgan Kaufmann, San Francisco (2005)
19. Wu, T.F., Lin, C.J., Weng, R.C.: Probability estimates for multi-class classification by pairwise coupling. The Journal of Machine Learning Research 5, 975–1005 (2004)

The Orange Customer Analysis Platform

Raphaël Féraud, Marc Boullé, Fabrice Clérot,
Françoise Fessant, and Vincent Lemaire

Orange Labs,
2 avenue Pierre Marzin
22300 Lannion

Abstract. In itself, the continuous exponential increase of the datawarehouses size does not necessarily lead to a richer and finer-grained information since the processing capabilities do not increase at the same rate. Current state-of-the-art technologies require the user to strike a delicate balance between the processing cost and the information quality. We describe an industrial approach which leverages recent advances in treatment automatization and relevant data/instance selection and indexing so as to dramatically improve our capability to turn huge volumes of raw data into useful information.

1 Introduction

The rapid and robust detection of the most predictive variables is a key factor in a marketing application. An industrial customer targeting platform developed at Orange Labs, capable of building predictive models for datasets having a very large number of input variables (ten of thousands) and instances (tens of thousands), is currently in use by Orange marketing. A key requirement is the complete automation of the whole process. The system extracts a large number of variables from a relational database, selects a subset of informative variables and instances, and efficiently builds in a few hours an accurate classifier. When the models are deployed, the platform exploits sophisticated indexing structures and parallelization in order to compute the scores of millions of customers, using the best representation.

The challenge KDD Cup 2009 [1] was to beat the in-house system developed by Orange Labs on three standard marketing campaigns : the propensity of customers to switch provider (churn), buy new products or services (appetency), or buy upgrades or add-ons proposed to them to make the sale more profitable (up-selling). The results of KDD Cup show that automatic modeling on thousands of variables leads within few hours to results close to those obtained by top level researchers in a month. Knowing that a datamart containing tens of thousands of variables describing millions of instances is practically unfeasible, this interesting result raises two questions for industrial use:

1. How to build hundreds of models on tens of thousands of variables ?
2. How to deploy hundreds of models on millions of instances ?

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 584–594, 2010.

These questions were at the origin of the Customer Analysis Platform.

This paper describes this industrial customer targeting platform developed at Orange Labs. The paper is organized as follow : Section 2 describes the process of the targeting of marketing campaign, Section 3 presents the processing architecture and the modeling step, Section 4 presents the Deployment Cycle and Section 5 several experiments.

2 The Targeting of Marketing Campaign

Customer Relationship Management (CRM) is a key element of modern marketing strategies. The most practical way to build useful knowledge on customers in a CRM system is to produce scores to detect churn, propensity to subscribe to a new service... Hundreds of scores are produced by Orange marketing each month. These scores are then injected in the CRM tools to target incoming and outcoming marketing campaigns. The scoring process is an industrial process containing a lot of complex tasks :

1. Each month a customer datamart, called datafolder, is fed from the datawarehouse. As the datamart contains different domains of data such as customer, billing, uses, contacts..., it is ready to be used when the last domain of data is produced. Billing data are the last produced, at the middle of the current month. Few days after, all the scores have to be produced to feed CRM tools.
2. For each marketing campaign, a filter is applied on the datamart. The filter defines the population concerned by the marketing campaign. For example, for a churn purpose the filter selects the customers you want to retain.
3. Then the current model used to target customers of the marketing campaign is tested. To test the current model, the scores of the previous month are compared to the present. If the accuracy indicator such as AUC is not stable in comparison to previous values, a new model is learnt with recent data. The lifetime of a model is usually in the order of one year.
4. The model is deployed to produce scores of the current marketing campaign.

This description of the targeting process shows that the bottleneck is not the modeling task but the deployment task : most of the models are re-used each month, and the time constraint is strong on deployment since hundreds of scores have to be produced for millions of customers in only few days.

3 Platform Architecture

3.1 Introduction

This section gives an overview of the Orange Customer Analysis Platform. The block diagram of the Orange Customer Analysis Platform is presented Figure 1. The next sections of this paper enter more in depth to detail several parts of this platform.

The first step to obtain scores on customers is to build a datafolder: the input data from information system are structured, and stored in a simple relational database (see Section 3.2). Then the platform includes 2 mains cycle:

1. The modeling cycle which includes two different steps:

   – The modeling step: using the extraction language, with specification ‘A’, a modeling database is extracted from the datafolder (see section 3.3). This database contains $P1$ instances and $N1$ explanatory variables. $P1$ is a subset of the customers to be scored. Using this database the modeling step is performed; this step includes two main functions: the variable selection (see Section 3.4) and the construction of a classifier (see Section 3.5). At the end of the modeling step one has a classifier which uses a subset of the $N1$ explanatory variables: $N2$ ($N2 \ll N1$). Only this $N2$ variables will be used in the extraction language with specification ‘B’ and ‘C’.
   – The indexing step: using extraction language, with specification ‘B’, a filtered database is extracted from the datafolder. This database contains $P2$ instances and $N2$ explanatory variables where $P2$ represents all the customers to be scored at the end of the complete process ($P2 \gg P1$). Then the instance selection step is performed to extract a paragon table (see Section 4.2) which contains $P3$ real customers ($P3 \ll P2$), each described by $N2$ explanatory variables. The application of a k nearest neighbor (knn) and Locality Sensitive Hashing (LSH) algorithms on this table allows the creation of an indexation table (see Section 4.3). This indexation table links any customer ($P2$) to a customer of the paragon table ($P3$).

   The complete output of the modeling cycle is therefore: a classifier, an indexation table, and the extraction specification ‘C’ corresponding to the $N2$ explanatory variables, and to the $P3$ paragons.
2. The deployment cycle: knowing the output of the modeling cycle the extraction query with specification ‘C’ can be written in the extraction language and applied on a new data folder. This produces the paragon table (PT table in the Figure 1) and the indentifier table (ID table in the Figure 1). Then the classifier is applied on the paragon table to obtain the scores of the paragons. Finally knowing the scores of the paragons and the indexation table a joint is realized and therefore all the customers are scored (see Section 4.1).

3.2 The Data Folder

Unlike the current practice of data mining architecture, the explanatory variables are not a priori designed and computed in a datamart. In our platform architecture, the input data from information system are structured, and stored in a simple relational database : the data folder (Figure 2). The explanatory variables are built and selected automatically for each specific marketing project. In order to be computed in parallel and in memory, the datafolder is hashed in small datafolders of size 1 Go (Figure 2).

The data folder model provides a unique view of the available input data sources, normalized according a star schema:

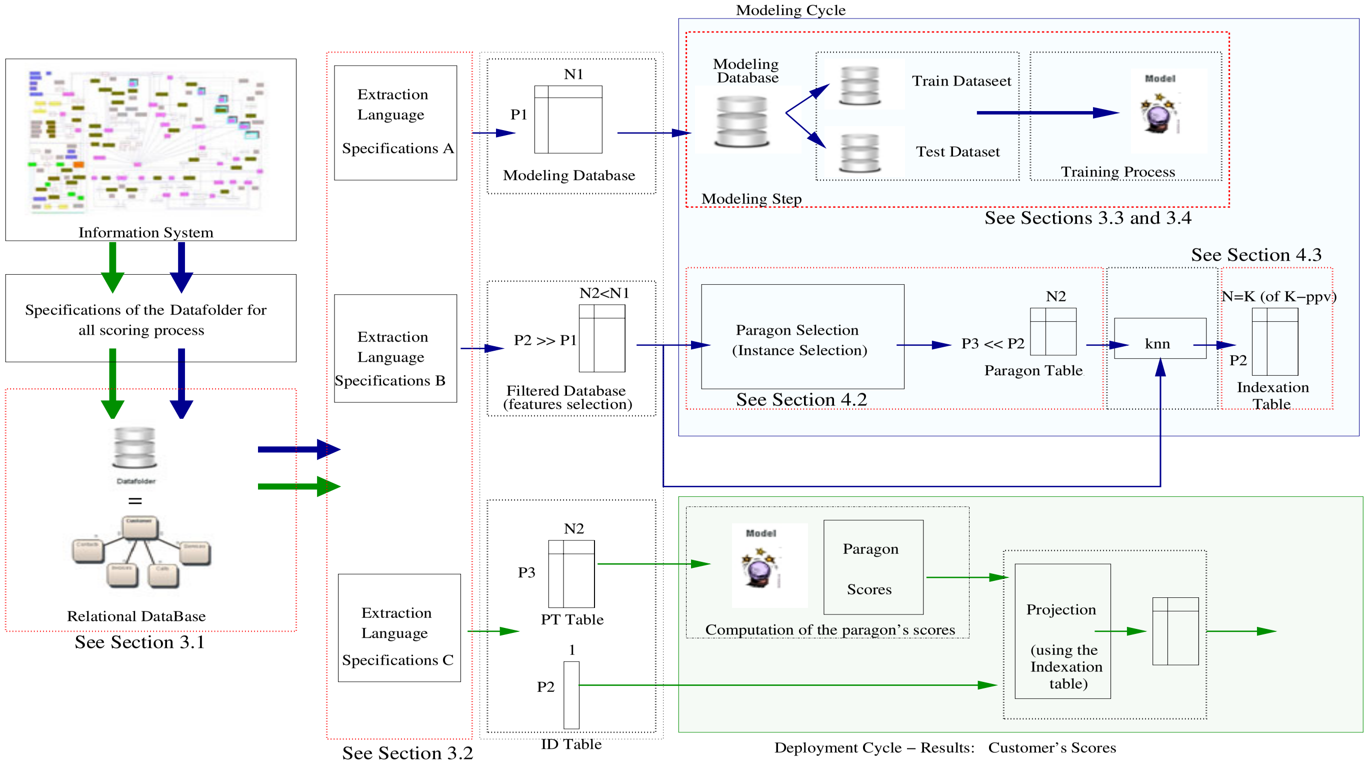

Information System
Specifications of the Datafolder for all scoring process
Datafolder
=
Relational DataBase
See Section 3.1
Extraction Language Specifications A
Extraction Language Specifications B
Extraction Language Specifications C
See Section 3.2
N1
P1
Modeling Database
N2<N1
P2 >> P1
Filtered Database (features selection)
N2
P3
PT Table
1
P2
ID Table
Modeling Cycle
Modeling Database
Train Dataseet
Test Dataset
Model
Training Process
Modeling Step
See Sections 3.3 and 3.4
See Section 4.3
Paragon Selection (Instance Selection)
See Section 4.2
N2
P3 << P2
Paragon Table
knn
N=K (of K−ppv)
P2
Indexation Table
Model
Paragon Scores
Computation of the paragon's scores
Projection (using the Indexation table)
Deployment Cycle – Results: Customer's Scores

Fig. 1. The Orange Customer Analysis Platform. Blue arrows and background: modeling cycle, Green arrows and background: deployment cycle.

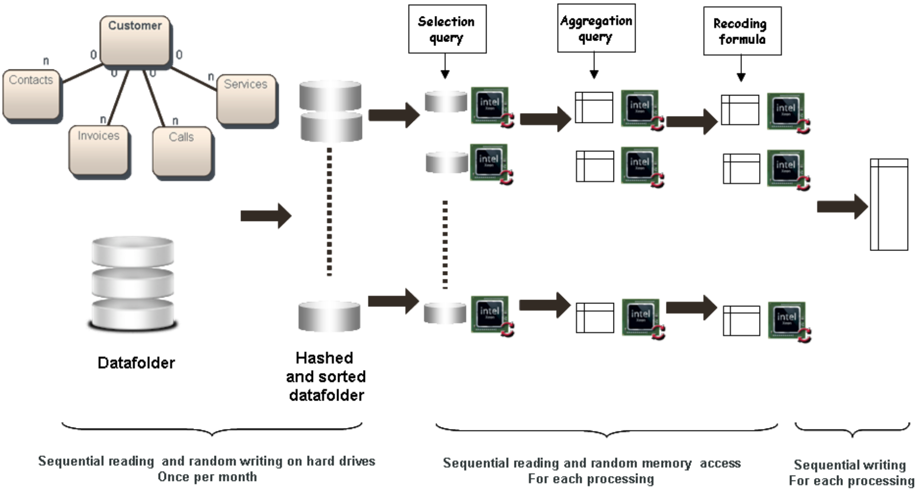


Fig. 2. Principle: data are normalized and hashed in a star schema database. Using extraction languages, learning algorithms drive data preparation, modeling and instance selection. A server executes in parallel most of the process. In this illustrative example, a flat table is extracted from the input datafolder.

– The primary table is related to the marketing domain. For customer data analysis, this table contains all the fields directly connected to the customer, such as his name or address.
– The secondary tables have a 1-N relationship with the primary table. Each instance of the primary table may be related to a variable number of instances of a secondary table. For telecommunication data for example, the secondary tables contains the list of services, of usages of theses services, the call details.

This type of data modeling has a large expressiveness, suitable for many data mining projects. It offers an efficient trade-of between single-table data mining and full multi-relational data mining. The star schema allows to efficiently build many constructed variables, when the join key belongs to the primary table, whereas in a traditional data-warehouse, the construction of one single variable may involve multiple table joins. Finally, this star schema modeling allows the design of formatted data extraction languages, with the purpose of automation of the data mining process.

3.3 Data Extraction

The data extraction functionality of the platform is parametrized using three languages:

- a selection language to filter the instances,
- a construction language to build a flat instance x variables representation from the data folder,
- a preparation language to specify the recoding of the explanatory variables.

These languages are both simple enough to be automatically exploited by the process of variable selection and expressive enough to build a large variety of explanatory variables. Each language expression deals with at most two tables: the primary table plus eventually one secondary table. The join key always belongs to the primary table, and the selection and construction operands exploit the fields of any table, primary or secondary. For example, to build the number of usages of each service per weekday for all customers, one single language expression needs to be specified, with the use of the "Count" operator on the secondary table "Usage" with two operands "WeekDay(Date)" and "Label(ServiceId)". It is then possible to specify up to thousands of variables to construct, using one single expression of the construction language.

3.4 Variable Selection

The platform architecture allows to easily build flat data tables with up to tens of thousands of constructed variables. In order to select the best representation, that is the best subset of informative variables, a powerful variable selection method [2,3] is required, both robust and efficient. In the context of decision trees [4,5,6], supervised discretization methods are employed at each node of the tree in order to select the next split variable, using filter criteria based on statistical tests [7], error rate or entropy [8]. When the number of intervals of the discretization is a free parameter, the trade-off between information and robustness is an issue. In the MODL (Minimum Optimized Description Length) approach, supervised discretization [9] (or value grouping [10]) is treated as a nonparametric model of conditional probability of the output variable given an input variable. The discretization is turned into a model selection problem and solved in a Bayesian way. The best discretization and value groupings are optimized using the bottom-up greedy heuristic described in [9]. One advantage of this filter approach is that non informative variables are discretized in one single interval and can thus be reliably discarded. The algorithmic complexity of $O(n \log n)$ of this heuristic and the excellent reliability of this method allow to preprocess a very large number of variables, around 50000 in our experiments, and to select a small subset of informative variables, typically 10% of the input variables in the marketing domain.

3.5 Modeling

The naive Bayes classification approach [11,12,13] is based on the assumption that the variables are independent within each output label, and simply relies on the estimation of univariate conditional probabilities. In the Orange Customer

Analysis Platform, this approach benefits from the high quality MODL preprocessing. The naive independence assumption can harm the performance when violated. In order to better deal with highly correlated variables, the selective naive Bayes approach [14] exploits a wrapper approach to select the subset of variables by optimizing the classification accuracy. In this seminal work, the search algorithm has a quadratic time complexity w.r.t the number of the variables, and the selection process which is prone to overfitting. In [15], the search algorithm is able to process large numbers of variables with super-linear time complexity, and the over-fitting problem is tackled using a Bayesian regularization approach. Finally, a model averaging approach is applied in order to achieve better accuracy and reliability. Using the naive Bayes assumption, weighting many models of variable selection reduces to one single naive Bayes classifier with weighted variables, allowing an efficient deployment of the ensemble of selective naive Bayes classifiers.

To summarize, in the platform, a selective naïve Bayes classifier [15] leverages the MODL preprocessing, variable selection regularization and model averaging in order to build effective scores fully automatically. This method is efficiently implemented into the Khiops scoring tool (available as shareware, see `www.khiops.com`).

4 Efficient Deployment

4.1 Principle

To produce scores, a model has to be applied for all instances on all explanatory variables. To speed up this process, a table of paragons containing representative individuals is extracted. The paragons are connected by an index to all the population. The scores of all instances are obtained by a simple join between the table of the paragons and the index. This method of deployment is particularly effective when the model is deployed several times. For example for monthly marketing campaigns, only the reduced table of the paragons is built each month to produce the scores of all instances. This approach makes it possible to increase dramatically the number of scores which can be produced on the same technical architecture.

4.2 Paragons Selection

The table of the paragons is crucial for the final performance of the system. A poorly representative paragon table leads to ineffective scores, on the other hand, a too large paragon table increases computational cost.

The table of paragons is drawn from the datafolder to be representative of the variables relevant for the model. To produce and maintain online a sample of size n, Reservoir Sampling algorithm [16]) can be used. An inclusion probability of $n/(t+1)$ is given for each tuple arrived at time t. An interesting property of this algorithm is that, when t tuples have been observed, all the t tuple have the same probability to be included in the reservoir: n/t. Biased versions of this algorithm may take into account recent data ([17]) or weighted data ([18,19,20]).

As the frequencies of discretized explanatory variables are known from the variable selection stage, a biased version of Reservoir Sampling can be used to draw the paragons. To control and speed up the convergence time, we use a deterministic version of Biaised Reservoir Sampling. A reservoir is filled until it reaches the desired size P without removing any instance :

1. The reservoir is initialized with the first K instances.
2. At each iteration an instance is chosen to optimize Khi^2 criterion between theoretical frequencies and frequencies observed in a windows of size M, with $M \ll P$.
3. Then the search window is shifted of L instances in order to fill the reservoir of size P in one pass on the table of size N : $L = (N - M)/P$.

The size of the search windows allows to tune the trade-off between computational time cost and accuracy of the algorithm : the more M is, the more the accuracy is and the less the computational time cost is.

4.3 Data Indexing

The problem to be solved is simple to state: being given an individual, to find his nearest neighbor in the table of paragons. The $L1$ norm between the explanatory variables is used to evaluate the distance between instances. This task has to be executed for all the instances of the datafolder. The search of nearest neighbors is an expensive operation. Its naive implementation implies an exhaustive research among the paragons, therefore a complexity in $\mathrm{O}(nmp)$, n being the number of instances, m the number of explanatory variables and p the number of paragons. In order to accelerate the research of nearest neighbors, a compromise between speed and accuracy can be done : to find a paragon close to the nearest using Locality Sensitive Hashing [21] allows. This algorithm is based on a technique of hashing to select good candidates among the paragons to be close to the nearest. Then an exhaustive search is done on good candidates to find the paragon. Our implementation of this technique makes it possible to bring back the complexity of the search close to $\mathrm{O}(nm\sqrt{p})$. It reduces the computational cost of a factor 300 per 100000 paragons, and leaves to the user the control of the compromise speed / performance.

5 Experiments

We compared the scores produced with our platform (including the Khiops scoring tool) and with the current model for several Orange marketing campaigns.

The current model is built with KXEN [22] on a datamart containing about 700 explanatory variables. To supply the platform, we have collected data on about one million of customers between January and June 2005. The information comes from decisional applications of Orange Company. The first four months

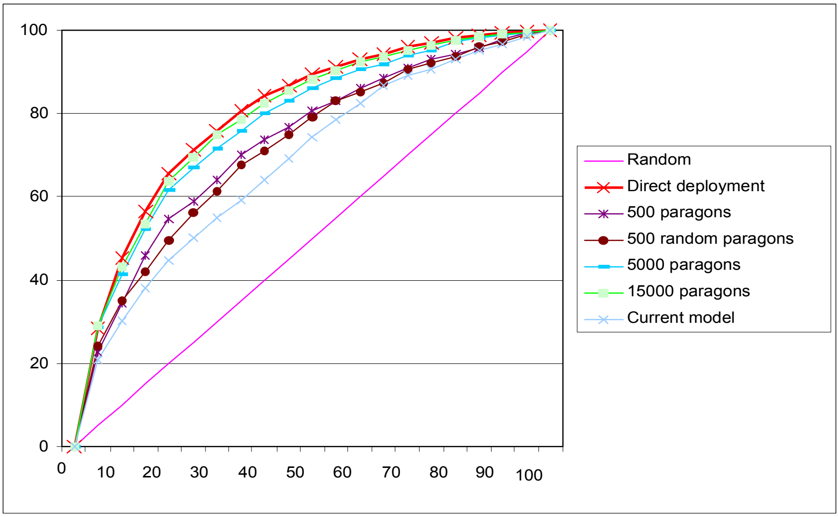


Fig. 3. Lift curve of predictive models for churner detection

have been used to build the customer profiles, the last two to compute the target variable. 20% of the customers are kept for the evaluation of the models.

The performance of a model is measured with the cumulative gain curve (Figure 3). It is a graphical representation of the advantage of using a predictive model to choose which customers to contact. The x-axis gives the proportion of the population with the best probability to correspond to the target, according to the model. The y-axis gives the percentage of the targeted population reached.

The goal of the campaign presented below is to prevent a customer to switch ADSL provider.

We plotted the cumulative gain curves for several predictive models on Figure 3. The diagonal represents the performance of a random model. If we target 20% of the population with this random model, we are able to reach 20% of the customers who will churn in next two months. With the current model, when 20% of the population is targeted, 45% of the fragile customers are reached. Compared with a random targeting we have a gain (G_1) of 2.25 ($G_1 = \frac{45}{20} = 2.25$)

The automation of the search of representation has led us to select a model based on 191 explanatory variables chosen among a set of 50000 variables.

The model deployment is then achieved on all the instances with a variable number of paragons: 500, 5000, 15000 and also directly on the population. In the case of a direct deployment on all the instances, if we contact 20% of the population based on this new modelling, 65% of the fragile customers are targeted. Compared with the current technique, we have a gain (G_2) of 1.4 ($G_2 = \frac{65}{45} = 1.44$). This improvement remains true for the entire cumulative gain curve.

An in-depth analysis of the most relevant variables kept by the targeting model built with the platform can help us draw the portrait of a typical churner. His engagement ends in next 4 months, he lives in a dense area, he is young (between 14 and 27 years old) and his volume of traffic has changed a lot (decrease or strong increase) in last 3 months. 5 of the 10 most important variables are not present in the initial datamart, used for the current model. They have been constructed directly from the data folder and specifically for the churn campaign. This is the strength of our methodology: with our platform we are able to explore a large number of new variables on demand, according to a specific campaign and select the most relevant of them.

Let's turn now to score deployment with paragons. When such a technique is applied, there is a loss of reliability which depends on the number of paragons. The targeting comes close to the best when the number of paragons increases but it is also very costly. For example, when 5000 paragons are used to represent 1000000 customers, at a level of 20% of the targeted population, 60% of the fragile customers are reached (+40% of gain compared with a random targeting and +15% compared with the current technique). With 15 000 paragons, the performances are similar to those of the direct deployment. To evaluate the quality of the algorithm of paragon selection, we have compared the performances obtained when the paragons are randomly selected and when the paragons are using a biased reservoir sampling on the theoretical distribution of explanatory variables. With 500 paragons, at the level of 20% of population, 50% of the target is reached for the random selection and 55% with biased reservoir sampling (Figure 3).

The whole process of extraction of a paragon table from one million customers and a representation space of 50000 variables takes about 3 hours on a server with 16 processors and 32 Go of RAM. One third of processing time is for the selection of the representation and two thirds are for the search and indexation of paragons. Once the paragons are available, the score production from the paragon table takes less than one minute.

One processing hour is necessary in a direct deployment to generate a table of one million instances with 191 explanatory variables and apply the predictive model on this table. It is very efficient to use paragons for the deployment of a recurrent score like fragility scores or ADSL recruiting. For an opportunist score such as appetency to a specific offer, a direct deployment is better.

6 Conclusion

We have described a data-mining platform which allows to build predictive models using two orders of magnitude more explanatory variables than the current state-of-the-art, resulting in a dramatic improvement of performances. The Orange Customer Analysis Platform relies on a novel architecture which allows to leverage recent advances in treatment automatization and relevant data/instances selection and indexing. The processing time associated with data table flattening remains the main limitation to the exploration of an even larger data space. The conception of an explanatory technique guiding the flatening towards the most promising areas of such huge spaces is a direction for further research.

References

1. http://www.kddcup-orange.com/ (last access on December 28)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3, 1157–1182 (2003)
3. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L. (eds.): Feature Extraction: Foundations and Applications. Springer, Heidelberg (2006)
4. Kass, G.: An exploratory technique for investigating large quantities of categorical data. Applied Statistics 29(2), 119–127 (1980)
5. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Classification and Regression Trees. Wadsworth International, California (1984)
6. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
7. Kerber, R.: Chimerge discretization of numeric attributes. In: Proceedings of the 10th International Conference on Artificial Intelligence, pp. 123–128. MIT Press, Cambridge (1992)
8. Kohavi, R., Sahami, M.: Error-based and entropy-based discretization of continuous features. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 114–119. AAAI Press/MIT Press (1996)
9. Boullé, M.: MODL: a Bayes optimal discretization method for continuous attributes. Machine Learning 65(1), 131–165 (2006)
10. Boullé, M.: A Bayes optimal approach for partitioning the values of categorical attributes. Journal of Machine Learning Research 6, 1431–1452 (2005)
11. Langley, P., Iba, W., Thompson, K.: An analysis of Bayesian classifiers. In: 10th National Conference on Artificial Intelligence, pp. 223–228. AAAI Press, Menlo Park (1992)
12. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. Machine Learning 29(2-3), 103–130 (1997)
13. Hand, D., Yu, K.: Idiot bayes? not so stupid after all? International Statistical Review 69(3), 385–399 (2001)
14. Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, pp. 399–406. Morgan Kaufmann, San Francisco (1994)
15. Boullé, M.: Compression-based averaging of selective naive Bayes classifiers. Journal of Machine Learning Research 8, 1659–1685 (2007)
16. Vitter, J.: Random sampling with a reservoir. ACM Trans. Math. Software 11(1), 37–57 (1985)
17. Aggrawal, C.: On biased reservoir sampling in the presence of stream evolution. In: Proceedings of the VLDB conference (2006)
18. Chaudhuri, S., Motwani, R.: On sampling and relational operators. In: IEEE on Data Engineering (1999)
19. Kolonko, M., Wasch, D.: Sequential reservoir sampling with a non-uniform distribution. Technical report, University of Clausthal (2004)
20. Efraimidis, P.S., Spirakis, P.G.: Weighted random sampling. Technical report, Research Academic Computer Technology Institute (2004)
21. Gionis, A., Indyk, P., Motwani, R.: Similarity search in high dimensions via hashing. In: VLDB Conference (1999)
22. http://www.kxen.com/ (last access on December 21)

Semi-supervised Learning for False Alarm Reduction

Chien-Yi Chiu[1], Yuh-Jye Lee[1], Chien-Chung Chang[1],
Wen-Yang Luo[2], and Hsiu-Chuan Huang[2]

[1] Department of Computer Science and Information Engineering
National Taiwan University of Science and Technology
Taipei, 10607 Taiwan
[2] Information & Communication Security Lab,
Chunghwa Telecom Laboratories

Abstract. Intrusion Detection Systems (IDSs) which have been deployed in computer networks to detect a wide variety of attacks are suffering how to manage of a large number of triggered alerts. Thus, reducing false alarms efficiently has become the most important issue in IDS. In this paper, we introduce the semi-supervised learning mechanism to build an alert filter, which will reduce up to 85% false alarms and still keep a high detection rate. In our semi-supervised learning approach, we only need a very small amount of label information. This will save a huge security officer's effort and make the alert filter be more practical for the real systems. Numerical comparison with conventional supervised learning approach with the same small portion labeled data, our method has significantly superior detection rate as well as in the false alarm reduction rate.

Keywords: Machine Learning, Semi-Supervised Learning, False Alarm Reduction, Intrusion Detection.

1 Introduction

In recent years, the rapidly increasing rate of cyber attacks make intrusion detection become a critical issue of network security. By the growth of the Internet and the large amount of network users, network traffic is horribly increasing. This phenomenon leads a result that alarms of intrusion detection system (IDS) become overwhelming for the analysts. Here we introduce a method of using the information of network connections to reduce false alarms. As we know, more and more network applications rely on the TCP protocol, especially the services over the World-Wide-Web (or say, over the http protocol). More and more users shop over Internet, such as booking a ticket or ordering dishes. The great volume of transactions lead the network criminals change their target from end users to popular web servers. Attackers try to embed malicious scripts or malwares into the web server to indirectly attack the great amount of web users. In this case, all the attacks over the http protocol are based on TCP protocol. By comparing

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 595–605, 2010.

with UDP and ICMP, TCP is a connection-oriented protocol. It has some additional information of the connections, such as the connection duration, bytes sent by source host, bytes sent by destination host, and so on. All of the information is used in intrusion detection [9] and holding a KDD'99 cup competition [7].

Different from general purpose IDSs which are using signatures to detect malicious information in packet payload, we use connections to analyze an alarm is suspicious or not. First, we use connection information to aggregate alerts together. After aggregating the alerts, we could directly classify a connection as suspicious one or not. By the classification results of the connections, we could determine whether the aggregated alerts are malicious or not.

In the previous work, analyst has to collect enough labeled data for the machine training. As we know, labeled data is expansive and hard to collect. In our experience, the alerts of the IDSs are very easy to collect. In contrast, to label an alert as true attack or false alarm is hard and expensive. That leads a problem, that we never know the amount of labeled data is enough or not. For using the limited labeled data, the performance of supervised learning techniques is not good enough as it could be. We propose using semi-supervised learning technique, named after *Two-Teachers-One-Student* (2T1S) [5] to solve this problem. With the corresponding connection information of the alerts, we could use the large amount of unlabeled data with few labeled data to enhance the IDSs' performance, just as a supervised learning technique could do with enough labeled data.

1.1 Contributions

Our target is to build a system, which can reduce the great amount of false alarms by corresponding TCP connection information. Moreover, for improving the performance, we use a semi-supervised learning technique 2T1S to gain more useful information from the large amount of unlabeled data.

For achieving the goals, we built a experimental system to test our ideas. Along the experimental results, we conclude the following contributions:

- Successful reducing false alarms with connection information
- Using semi-supervised learning technique 2T1S to improve the performance while only a few labeled data are available.

The remainder of the paper is organized as follows. In Section 2, we review related works, including alert classification and machine learning based intrusion detection system. In Section 3, we introduce the framework of our method. Section 4 describes the numerical experiments and details the results. Section 5 contains some concluding remarks.

2 Related Work

Machine learning techniques used on reducing false positives is not new. Tadeusz Pietraszek [15] using adaptive alert classification for supporting human analyst by classifying alerts into true positives and false positives. He addresses the false alarm problem by building a classifier, which is so-called Alert Classifier that tells true from false alarms. This kind of method is known as *Alert classification.* Another way to apply machine learning techniques on reduce false alarms is named after *Alert Sequence Classification.* For example, Alharby [2] proposed a method to characterize a "normal" stream of alarms. He developed an algorithm for detecting anomalies with continuous and discontinuous sequential patterns.

Except using classification methods to reduce false alarms, machine learning techniques is also used for building a system to detect network attacks. This kind of works are rich and widespread, dating back to at least 1980 with Anderson's [3] initial proposal for such system. Lee [9] developed a methodology to construct additional features using data mining. He used the additional features to classify if a connection is malicious or not. In recent years, semi-supervised learning is brought in this category. Lane [8] proposed a model to fuse misuse detection with anomaly detection and to exploit strengths of both. Chen et al. [6] proposed two semi-supervised classification methods, Spectral Graph Transducer and Gaussian Fields Approach, to detect unknown attacks. They also developed a semi-supervised clustering method, MPCK-means to improve the performances of the traditional purely unsupervised clustering methods. Mao et al. [12] proposed a co-training method framework for intrusion detection, which is a semi-supervised learning method to utilize unlabeled data and combine multi-view data.

To the best of our knowledge, semi-supervised learning technique has not previously used in building classifier to reducing the false alarms. However, some concepts we apply here have been successfully used in intrusion detection and related domains. We adapt Lee's method to construct statistical features of connections, and use these features to judge corresponding alerts are suspicious or not. We also adapt the algorithm, *Two-Teachers-One-Student* (2T1S), proposed by Chang et al. [5] for improving the performance with the unlabeled data.

3 Methodology

In this section, we first describe our motivation below. We construct the statistical connection features for reducing the false alarms, and try to use semi-supervised learning technique to improve the performance by utilizing the unlabeled data. For constructing the statistical features, we have a NCF instance extractor to extract connections from network traffic and compute statistical features with a two seconds time window. We also adapt a semi-supervised learning algorithm, 2T1S, in our machine learning based analysis engine to improve the performance via including the information of unlabeled data.

Fig. 1. illustrates the framework of our proposed system. Except the intrusion detection system as a sensor, our system has an additional sensor to create the TCP connection database, and mapping the alerts generated by the IDS with the connection records to compute the statistical features. All the alerts mapped to the same connection will be aggregate as a small cluster, and share the same

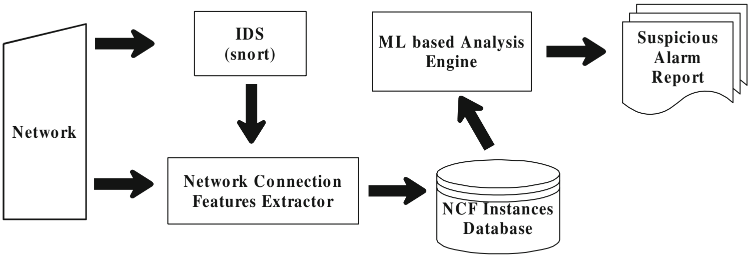


Fig. 1. The System Architecture

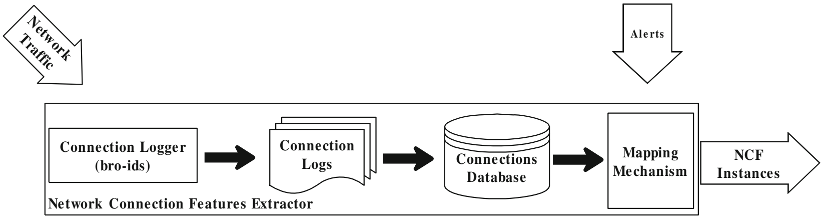


Fig. 2. The Network Connection Features Extractor

connection features. The mapped connection with the corresponding statistical features will be look as a Network Connection Features instance (NCF instance), which is stored in NCF Instance Database. Our machine learning based analysis engine will directly read the NCF instances as input data for analyzing corresponding alerts are true attacks or false alarms. Below, we describe the detail of our system framework.

3.1 IDS(Snort)

We use snort [16] as our IDS sensor due to its popularity and open source. Snort is a libpcap-based packet sniffer/ packet logger/ signature-based IDS developed by Marty Roesch. Snort was originally intended to be a packet sniffer, Roesch added the signature-based analysis (also known as rule-based analysis within the snort community) to be a rules-matching IDS. As time progressed, the size of the latest rules is increasing with the number of exploits available. That is also the reason why snort will generate large amount of false alarms.

3.2 Network Connection Features Extractor

The Network Connection Features Extractor (NCF Extractor) is composed by a TCP connection logger and a mapping mechanism. The architecture is shown in Fig. 2.

Most of the signature-based IDSs use the packet-signature to generate alerts. It means a packet is malicious or not just depends on the packet's header and payload information. However, for the network security managers, the information from only one alert is not helpful for determining whether it is a true alarm or not. The managers need more information of the hosts, including the source host information and the target host information, which triggers the alert. All of the required information for labeling alerts as attack or not is just like the service type, which the hosts provide for, or the alerts triggered by the same hosts. For the purpose, we think the statistical TCP information could make up the needed information.

For generating the connection statistical features, the TCP connection information is needful. We set a TCP connection logger to extract the connections from network traffic, and output all the connections by a connection log. We use Bro [14] as our TCP connection logger. Bro provides the function to log the parsed TCP connection semantics, which we used as the basic features of TCP connection. The summary of the TCP connection information [1] is introduced as follows:

- start: the connection's start time.
- duration: the connection's duration.
- local IP & remote IP: local and remote addresses that participated in the connection.
- service: connection's service, as defined by service.
- local port & remote port: the ports used by the connection.
- org bytes sent & res bytes sent: the number of bytes sent by the originator and responder, respectively. These correspond to the size fields of the corresponding endpoint records.
- state: the state of the connection at the time the summary was written (which is usually either when the connection terminated, or when Bro terminated).
- flags: a set of additional binary state associated with the connection.
- tag: reference tag to log lines containing additional information associated with the connection in other log files(e.g.: http.log).

We provide another viewpoint to judge the alerts are malicious or not. Instead of reading the alarms' information directly, we aggregate alerts by the TCP connections. The alerts belong to the same connection; we set they have the same characteristics on the connection viewpoint. If a connection contains at least one malicious alarm, we set this connection as a malicious one. The alerts belong to the malicious connections; we look them all as suspicious alarms.

When Bro extracts the TCP connections information into logs, we will restore the connection records into a connections database. With the incoming alerts, we use the TCP pairs (Source IP, Destination IP, Source Port and Destination Port) and the trigger time to find if a matching connection exists. If the connection exists, we will compute the statistical features with a two seconds time window. The statistical features will be combined with the basic connection information to generate a NCF instance. We also store the mapping relationship between alerts and connections into database for constructing the alert clusters. After all, the NCF instance will be send to the NCF Instance Database, each NCF instance represent the alert cluster, which mapped to corresponding connection.

3.3 Machine Learning Based Analysis Engine

Machine Learning Based Analysis Engine is used for learning and predicting the NCF instances. In the following statements, we describe the learner, which we used in the machine learning based analysis engine. Beyond what the supervised learning can offer, many real applications need to deal with both labeled and unlabeled data simultaneously. Usually, the amount of labeled data is insufficient and obtaining it is expensive. In contrast, unlabeled data is abundant and easy to collect. For example, we may need to categorize a number of web documents, but only a few of them may be correctly labeled. In another example, determining the functions of biological strings is expensive, and only a small portion of them have been studied (labeled) to date. Semi-supervised learning can help researchers deal with these kinds of problems because it takes advantage of knowing two kinds of data; 1) it uses labeled data to identify the decision boundary between data with different labels; and 2) it uses unlabeled data to determine the data's density, i.e., the data *metric*.

Among the various semi-supervised learning algorithms that have been proposed, the multi-view approach is one of the most widely used. This kind of methods split data attributes into several attribute subsets, called *views*, to improve the learning performance. In the *co-training* algorithm [4], classifiers of different views learn about the decision boundaries from each other. Based on this concept, a number of variants have been developed, e.g., the *tri-training* algorithm [17]. On the other hand, the classifiers of different views can be combined to form an *ensemble* classifier with a high level of confidence. We call this approach *consensus training*.

In this paper, we use a semi-supervised algorithm, Two-Teachers-One-Student (2T1S) [5], as the learner of our machine learning based analysis engine. 2T1S is a multi-view algorithm. Different from regular multi-view methods, 2T1S selects different views in the feature space rather than in the input space. 2T1S elegantly blends the concepts of co-training and consensus training. Through co-training, the classifier generated by one view can "teach" other classifiers constructed from other views to learn, and vice versa; and by consensus training, predictions from more than one view can give us higher confidence for labeling unlabeled data. In practice, given three different views, 2T1S choose two views as teachers for consensus training and the remaining view as the co-training partner. The classification answers from two classifiers (two teachers) represent the consensus result, which is used to teach the third view (the student) to learn the labels for unlabeled data. This process is performed for each choice of teachers-student combination. After the student learns the data, the newly learned labeled data is added to the student's original labeled data set, as the set of guessed labeled data can be included for training in the next step if it is part of the teachers' sets in the next step. The whole process is run iteratively and alternately until some stopping criteria are satisfied. We describe the 2T1S algorithm with the pseudo code in Algorithm 1.

Algorithm 1. The *2T1S* Algorithm

```
Input:
  Initial labeled data $\mathcal{D}_L=\{(\mathbf{x}^i, y_i)\}_{i=1}^{\ell}$, $\mathbf{x}^i \in \mathbb{R}^n$, $y_i \in \{-1, 1\}$.
  Initial unlabeled data $\mathcal{D}_U=\{(\mathbf{x}^i)\}_{i=\ell+1}^{m=\ell+u}$, $\mathbf{x}^i \in \mathbb{R}^n$.
  Initial classifiers $f_1(\mathbf{x})$, $f_2(\mathbf{x})$, $f_3(\mathbf{x})$.
  Initial consensus level $0 \leq \varepsilon \leq 1$.
Output:
  The final discriminant model $f(\mathbf{x})$.
$\mathcal{D}_{L_i} \leftarrow \mathcal{D}_L, i = 1, \ldots, 3.$;
$iter \leftarrow 1.$;
$\mathcal{D}_L^{(0)} \leftarrow \mathcal{D}_L.$;
repeat
    for $i \leftarrow 1$ to 3 do
        for $j \leftarrow 1$ to $u$ do
            $t_1 \leftarrow mod(i-1, 3) + 1$;
            $t_2 \leftarrow mod(i, 3) + 1$;
            $s \leftarrow mod(i+1, 3) + 1$;
            if $(f_{t_1}(\mathbf{x}^j) \geq \varepsilon$ and $f_{t_2}(\mathbf{x}^j) \geq \varepsilon)$ or;
               $(f_{t_1}(\mathbf{x}^j) \leq -\varepsilon$ and $f_{t_2}(\mathbf{x}^j) \leq -\varepsilon)$;
            then
                $\mathcal{D}_{L_s} \leftarrow \mathcal{D}_{L_s} \cup \mathbf{x}^j$;
                $\mathcal{D}_L \leftarrow \mathcal{D}_L \cup \mathbf{x}^j$;
                $\mathcal{D}_U \leftarrow \mathcal{D}_U \setminus \mathbf{x}^j$;
        Retrain the classifier $f_s(\mathbf{x})$ with $\mathcal{D}_{L_s}.$;
    $\mathcal{D}_L^{(iter)} \leftarrow \mathcal{D}_L.$;
    $iter \leftarrow iter + 1.$;
until $\mathcal{D}_L^{(iter)} = \mathcal{D}_L^{(iter-1)}$ ;
Construct an RSVM classifier $f(\mathbf{x})$ with the final labeled data set $\mathcal{D}_L.$;
Return $f(\mathbf{x}).$;
```

Here we need to emphasize the difference between network packets and connections. Most IDSs generate alerts by packet-signature, and the communicating packets between two hosts form a connection. For long duration connections, one connection may contain several alerts in it. If anyone of the alerts, which was contained by a connection is true attack, we label the connection as malicious. All the alerts mapped to the malicious connection will be concerned as suspicious alerts. When NCF Extractor generates NCF instances, we record the relationship between the alerts and connections. Alerts belong to the same connection will share the same features and classification result of NCF instance. In learning phase, we will use the NCF instances to learn a model or so-called a classifier. The model will be used for predicting a new incoming NCF instance in predicting phase. All the suspicious connections predicted by the model will be output in a suspicious alarm report.

4 Experiments

4.1 Dataset Description

DARPA intrusion detection evaluation dataset, which is sponsored by Defense Advanced Research Projects Agency and Air Force Research Laboratory, and managed by MIT Lincoln Laboratory since 1998. The available datasets, including DARPA1998, DARPA1999, and DARPA2000 datasets, generated in a simulated environment; however, they have some flaws identified both in simulation as well as the evaluation procedures [11][13]. We adopt the DARPA1999 dataset for experiments because it provides entire contents of attack database and attack truth files for labeling. For extracting TCP connections to analyze, we need the raw-traffic data as the input to our connection sensor. DARPA1999 dataset contains five weeks inside and outside sniffing data, fulfilling our requirement for extracting connection logs. All the dataset is separated into two parts. The first three weeks of the sniffing data are treated as training data, and the other two weeks are used as testing data.

In our experiment, we combine the inside and outside alerts and NCF instances together. We use Bro 1.4 as our connection logger and Snort 2.8.4.1 as our IDS. Table 1 summarizes the statistics of the datasets. Bro Connections stand for the connection amount extracted by Bro. Snort TCP Alerts means the alerts generated by snort and belong to TCP protocol. The false alarm rate is the false positive rate in training and testing set.

Table 1. Dataset Statistics

	Training set	Testing set
Bro Connections	1640157	1116166
Snort TCP Alerts	13912	16966
False Alarm Rate	92.33%	98.22%

4.2 Evaluation

For evaluating the experiment results, we consider two metrics to assess the performance of the learning methods. The first metric is detection rate, which is used for showing the missing rate of true attacks. The second one is the reduction rate, which stands for displaying the rate of the filtered alarm.

We use RSVM [10] as our supervised learner to test our approach of using connection information to reduce the false alarm. Before we use the dataset to learn a model, we perform three preprocessing works:

- Feature selection using information gain and gain ratio.
- Pick 50 both positive and negative training points as our reduced set to build the kernel matrix.
- Over-sampling the positive points to reform the unbalanced dataset.

Table 2. Testing Result of Supervised Learning and Semi-Supervised Learning with partial labeled data. (repeat 10 times).

	Supervised Learning		Semi-Supervised Learning	
Ratio of Labeled Data	Detection Rate	Reduction Rate	Detection Rate	Reduction Rate
1%	0.6766±0.057	0.8482±0.049	0.7046±0.072	0.7748±0.145
3%	0.7425±0.044	0.6725±0.154	0.7822±0.053	0.7814±0.113
5%	0.7508±0.055	0.6272±0.114	0.7912±0.067	0.8128±0.076
7%	0.7790±0.033	0.5829±0.012	0.8321±0.082	0.7934±0.094
10%	0.7460±0.041	0.6451±0.112	0.8607±0.046	0.8141±0.085
30%	0.8613±0.064	0.8430±0.016	0.8917±0.048	0.8527±0.041
50%	0.8417±0.070	0.8603±0.022	0.8861±0.037	0.8263±0.018
70%	0.9091±0.043	0.8401±0.026	0.8963±0.029	0.8532±0.015

We got a result of reducing 66.5% false alarms by missing 4.4% true attacks on testing data. It stands for filtering 11693 alerts and only less than 0.1% alarms belong to malicious. The result shows the connection features work well with a supervised learner, RSVM.

After we got the previous results, we began to test if the performance of supervised learning technique is affected by the size of labeled data. We random pick a small portion of the training data as our training set to build model, and test if the model could classify the testing data correctly. This process will be repeated 10 times for calculating the mean and standard variation of the results. The results are shown in Table 2.

From the results, we could easily know the performance is affected seriously by the size of the labeled data. The detection rate rise and down between 67.7% to 90%, and the reduction rate is varied from 58% to 86%. Here we need to emphasize the trade off between the detection rate and the reduction rate. When we tuning the RSVM parameters for learning a better model, we could easily find a model with great detection rate but awful reduction rate. It means the model almost classifies all the alerts as malicious. In contrast, a model with high reduction rate but very low detection rate also exists. It means the classifier tell us most of the alerts are benign. In Table 2, we choose a relatively good model in both detection rate and reduction rate. That also the reason why the detection rate does not always monotonic increase with the increase in percent of labeled data. For improving the performance, we attempt to use semi-supervised learning techniques to test if the unlabeled data will be helpful on building the model.

We choose 2T1S as our semi-supervised learner, and make some modification to let it be suitable to apply here. The modification is list as following:

- Feature selection using information gain and gain ratio.
- Over-sampling positive points before base learner training the classifier.
- Apply different parameters on each base learner.

The results of classifying the testing data using the model learned from 2T1S are also shown in Table 2. With 10% partial labeled data, supervised learning can merely reduce 64.5% alerts and detect 74.6% true attacks. At the same time, semi-supervised learning can detect 86.1% true attacks and reduce 81.4% alerts, both detection rate and reduction rate of semi-supervised learning is significantly better than supervised method. In most cases, semi-supervised learning also has better results than supervised one. These results strongly support our ideas that exploiting the information of unlabeled data could improve the performance.

5 Conclusion

In this paper, we successfully using the connection features to reduce false alarms by both supervised and semi-supervised learning techniques.

We use Network Connection Feature instance (NCF instance) to represent corresponding cluster of alerts. NCF instances will be fed to the machine learning based analysis engine to learn a model for classifying alerts into suspicious alerts or false alarms. In our experiments, we use RSVM as our supervised learning algorithm to test whether our framework works well in reducing false alarms by the NCF instances. The results show that we could filter out 65% false alarms and only miss less than 0.1% true attacks in the filtered alarms. However, having entire labeled data to build the alert filter is not practical. Thus, we introduced semi-supervised learning technique in this work. The numerical results show that both detection rate and reduction rate can be improved with very limited labeled data points. While only use small portion of labeled data in supervised learning will not have satisfied the results.

Because of the connection features used with the supervised learner and semi-supervised learner are the same, we believed that the semi-supervised learning technique could bring the improvement with the unlabeled data.

Acknowledgement

We would like to thank the anonymous referees for providing constructive comments. The work has been supported in part by Telecommunication Laboratories, Chunghwa Telecom Co., Ltd. under the grant 98AG054 and 99AG064, and by Ministry of Economic Affairs, R.O.C. under grant 98EC17A02S20137.

References

1. Bro reference manual: Analyzers and events (April 2007), http://www.bro-ids.org/wiki/index.php/Reference_Manual:_Analyzers_and_Events
2. Alharby, A., Imai, H.: IDS false alarm reduction using continuous and discontinuous patterns. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 192–205. Springer, Heidelberg (2005)
3. Anderson, J.P.: Computer security threat monitoring and surveillance. Technical report, Computer Security Division of the Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD (1980)

4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: COLT: Proceedings of the Workshop on Computational Learning Theory, pp. 92–100. Morgan Kaufmann Publishers, San Francisco (1998)
5. Chang, C.-C., Pao, H.-K., Lee, Y.-J.: An RSVM based two-teachers-one-student semi-supervised learning algorithm. Pattern Recognition (under submission)
6. Chen, C., Gong, Y., Tian, Y.: Semi-supervised learning methods for network intrusion detection. In: Proceeding of IEEE International Conference on Systems, Man and Cybernetics, October 2008, pp. 2603–2608 (2008)
7. Hettich, S., Bay, S.D.: The uci kdd archive (1999), http://kdd.ics.uci.edu/
8. Lane, T.: A decision-theoretic, semi-supervised model for intrusion detection. In: Machine learning and data mining for computer security: Methods and applications, number 978-1-84628-029-0 (Print) 978-1-84628-253-9 (Online). Advanced Information and Knowledge Processing, pp. 157–177. Springer, Heidelberg (2006)
9. Lee, W., Stolfo, S.J.: A framework for constructing features and models for intrusion detection systems. ACM Transactions on Information and System Security (TISSEC) 3(4), 227–261 (2000)
10. Lee, Y.-J., Mangasarian, O.L.: RSVM: Reduced support vector machines. In: Proceedings of the First SIAM International Conference on Data Mining (2001)
11. Mahoney, M.V., Chan, P.K.: An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection. LNCS, pp. 220–238. Springer, Heidelberg (2003)
12. Mao, C.H., Lee, H.M., Parikh, D., Chen, T., Huang, S.Y.: Semi-supervised co-training and active learning based approach for multi-view intrusion detection. In: Proceedings of the 2009 ACM symposium on Applied Computing, pp. 2042–2048. ACM, New York (2009)
13. McHugh, J.: Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security 3(4), 262–294 (2000)
14. Paxson, V.: Bro: A system for detecting network intruders in real-time. In: USENIX (ed.) Seventh USENIX Security Symposium proceedings: conference proceedings, San Antonio, Texas, January 26-29. USENIX (1998)
15. Pietraszek, T.: Using adaptive alert classification to reduce false positives in intrusion detection. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 102–124. Springer, Heidelberg (2004)
16. Roesch, M.: Snort - lightweight intrusion detection for networks. In: Large Installation System Administration Conference (LISA Conference), pp. 229–238 (1999)
17. Zhou, Z.-H., Li, M.: Tri-training: Exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering 17(11), 1529–1541 (2005)

Learning from Humanoid Cartoon Designs

Md. Tanvirul Islam, Kaiser Md. Nahiduzzaman, Why Yong Peng, and Golam Ashraf

National University of Singapore, Singapore-117417
tanvirulbd@gmail.com, kaisernahid@yahoo.com,
{psywyp,gashraf}@nus.edu.sg

Abstract. Character design is a key ingredient to the success of any comic-book, graphic novel, or animated feature. Artists typically use shape, size and proportion as the first design layer to express role, physicality and personality traits. In this paper, we propose a knowledge mining framework that extracts primitive shape features from finished art, and trains models with labeled meta-data attributes. The applications are in shape-based query of character databases as well as label-based generation of basic shape scaffolds, providing an informed starting point for sketching new characters. It paves the way for more intelligent shape indexing of arbitrary well-structured objects in image libraries. Furthermore, it provides an excellent tool for novices and junior artists to learn from the experts. We first describe a novel primitive based shape signature for annotating character body-parts. We then use support vector machine to classify these characters using their body part's shape signature as features. The proposed data transformation is computationally light and yields compact storage. We compare the learning performance of our shape representation with a low-level point feature representation, with substantial improvement.

Keywords: Shape Signature, Perception Modeling, Humanoid Cartoons.

1 Introduction

Character design is a key ingredient to the success of any comic-book, graphic novel, or animated feature. Recent advances in digital multimedia technologies have triggered widespread creation, consumption and distribution of digital character art in the form of videos, images, and textual descriptions. We view this large, unorganized, and distributed collection of digital humanoid character art on the internet, as a rich potential source for learning rules of good character design from the experts.

Though characters are remembered mostly for their roles in the story, several layers of visual detailing are employed to bring their roles to life. Starting with basic shape and proportion, artists create layers of skin tones, hair styles, attire, accessories, key postures, gait, action energy, mannerisms and facial expressions [1,2]. Furthermore, drawing styles may vary widely across cultures, mediums and entertainment genres. Thus, it may take years of learning and practice for novice artists to pick up the necessary skills to create impactful characterizations for a certain target audience.

Every year thousands of characters are produced worldwide for the billion dollar markets in animated features and games [22]. While computers are used mostly for

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 606–616, 2010.

shape-modelling/animation/rendering, conceptual character design still relies heavily on the skills and experience of the art department. Tools that could abstract character design rules from finished art would thus be really useful for this industry. It could also help hobbyists pick up better drawing skills.

In this paper, we focus on the basic shapes and body-part proportions layer as it plays a vital role in design and perception [1,2]. Artists use shape scaffolding to pre-visualize the final form, using basic shapes to represent each component or part. Apart from establishing the volume and mass distribution of the figure, these shapes may also help portray a certain personality, as is widely seen in stylized cartoon drawings. For example, in Pixar's recent animated feature titled "UP", the main protagonist had distinctively square features to highlight his "cooped-in" life. The square features were amplified by contrasting with a large round nose, as well as distinctly rounded supporting characters.

In this paper, we propose a knowledge mining framework that extracts primitive shape features from finished art, and trains models with labeled metadata attributes with a goal of finding hidden association rules. We use a primitive shape based vector annotation system for feature extraction. We then use Support Vector Machines to classify the characters into various traits with high accuracy. We compare the learning performance our shape representation with a low level point feature representation, with substantial improvement. The proposed data transformation is computationally light and yields compact storage. We have used physicality metadata and a variety of finished 2D humanoid character art, with a uniform body structure, but with reasonable variation in size, shape and proportions. The strongest contribution in this paper is our novel shape representation that allows learning, synthesis and retrieval in an intuitive data space. This has great implication in knowledge mining, computer vision and creation of expert systems for assisting creative design.

2 Related Work

We compare our work with prominent work in the area of representation and learning applied to character shapes and motion. We differ from these papers on two counts: 1) Our shape representation is in a language that humans can easily visualize; 2) It is easy to compute and can be efficiently hashed.

A variety of shape representation strategies have been used for learning and cognition of visual data. Edelmann and Intrator [4] proposed the use of semantic shape trees to represent well-structured objects like the hammer and airplane. Their goal was to recognize object classes, starting bottom-up with low level features. A drawback of this method is that it needs explicit modeling of the object grammar. Gal et al. [5] propose 2D histogram shape signature that combines two scalar functions defined on the boundary surface, namely a local-diameter and average geodesic distance from one vertex to all other vertices. Though this approach is robust to minor topological changes due to articulation in meshes, the representation lacks intuitiveness.

Classification and regression models on anthropomorphic data have been widely used in the fields of graphics and vision. In most of these models, the feature vectors used are fairly low level; e.g. Cartesian points, curves, distances and moments. Liu et al. [13] perform PCA on low-level point features for original and caricature drawings of human faces. Gooch et al. [8] also cartoonify face photographs by computing Eigenvalues between key facial points after training their system with real face data. Wang et al. [23] used rotational regression to learn deformation offsets of vertices in relation to driver skeletal joints. Meyer and Anderson [15] propose a computation cache for neighborhoods of key points undergoing lighting or deformation calculations, again using PCA analysis on point features. Hsu et al. [10] use CART data mining on body distance measurements (e.g. waist-girth, hip-girth, etc.) and body mass index to classify them into Large/Medium/Small categories for garment production. Marchenko et al. [14] differ from all these approaches by combining ontological metadata (e.g. artist name, style and art period) with low-level image features (e.g. brushwork and color temperature). Though they do not do any shape analysis, they implement a practical learning framework that improves learning results with human-understandable conceptual knowledge layers.

Automatic extraction of information from cartoon images of humanoids poses a number of challenges like perspective distortions, obscured body parts due to posing, and exaggerated non-standard body parts (unlike real humans). We did not find any method that provides a robust solution to this ill-posed problem. Since our goal is not the automation of data collection, which by itself is a significant challenge, we designed a user-friendly system to allow manual annotation of shapes. We derive inspiration from the use of primitive shapes outlined in art books [1,2,3,9,19] as well as shape perception literature [6,16]. We propose a novel vector shape that blends and smoothly morphs between three primitive shapes: circle/triangle/square. According to the Gistalt school of thought [16], we perceive shapes in relation to one another, as well as an overall sum of parts (instead of scrutinizing details of individual parts independently). Keeping this in view, we subjected the full body representation to training, rather than individual body parts. In this paper we demonstrate our shape transformation results with SVM, taking inspiration from Gil-Jiménez et al. [7] who also use SVM to classify shapes (to identify traffic light patterns in live footage).

3 Methodology

We have gathered a collection of nearly 300 approximately front-facing humanoid character images from various digital and physical resources. We intend to find the relationship between perception labels and measurable physical shapes extracted from the body images. We have manually annotated the body part shapes with our shape widget, and collected perception labels for these characters from laymen via online surveys and games. These games were designed in such a way that the player while playing the game, as a byproduct, provides us useful information. In this case it is the perception labels on characters. The perception labels were gathered for the full visual design as well as just the annotated body-part shape outlines. Using our body parts shape vector data as features and the averaged perception labels as classes we then classify these characters. We now outline the key components of the paper, namely data collection, vector shape representation, and training.

4 Data Collection

We have collected humanoid characters from different genres, namely 2D classic, 2D action, 3D movies, Manga, and unpublished art. As of now, we sample data from all these genres in our paper. In future, when we are able to grow our collection, it would probably make more sense to create individual models for each genre.

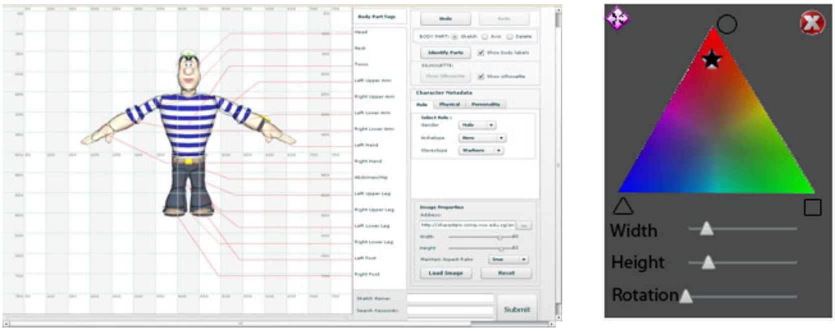


Fig. 1. (L:) The annotation tool (R:) The shape annotation widget

Fig. 1 illustrates the shape annotation tool and the body part shape control widget. The shape control widget allows single gesture control of the given body-part shape. The location of the black star cursor in the shape interpolation triangle controls the shape blend weights for the three primitive shapes. The interpolation space is triangular as we find enough expressivity with the circle, triangle and square shapes. It also allows us to directly plug the normalized Barycentric coordinate offsets of the star cursor from the three shape corners, to the corresponding shape weights.

The annotation was done by artists with reasonable knowledge of character design and concept art. The character perception labels was also captured from both artists as well as general audience. This ensured that the data set contains a mix of opinion from the content creators and the content consumers. Each body part can be created independently by clicking on the corresponding image location, and specifying the length, breadth, medial axis, and shape weights. The annotation tool can automatically guess the identity of the annotated parts; i.e. which shape corresponds to which body part, using ideas from Thorne et al. [18]. In case some really odd-proportioned character breaks our rules, the artist can easily override the label assignments. Once the annotation is done we save the character cages to our database in the following form: {character_id, perception labels, bodyPartVector$_{1\text{-}16}$}

5 Vector Shape Representation

In this section we discuss details of our novel shape representation blending circle, triangle and square. As explained in the literature review, almost all peer methods store low level contour point data before reducing their representation with methods

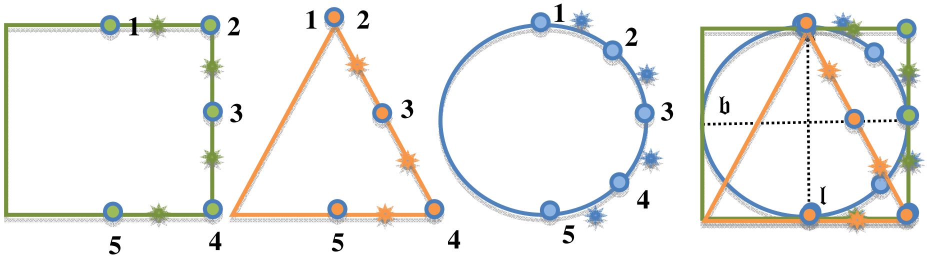


Fig. 2. Consistent interpolation of circle, triangle, and square

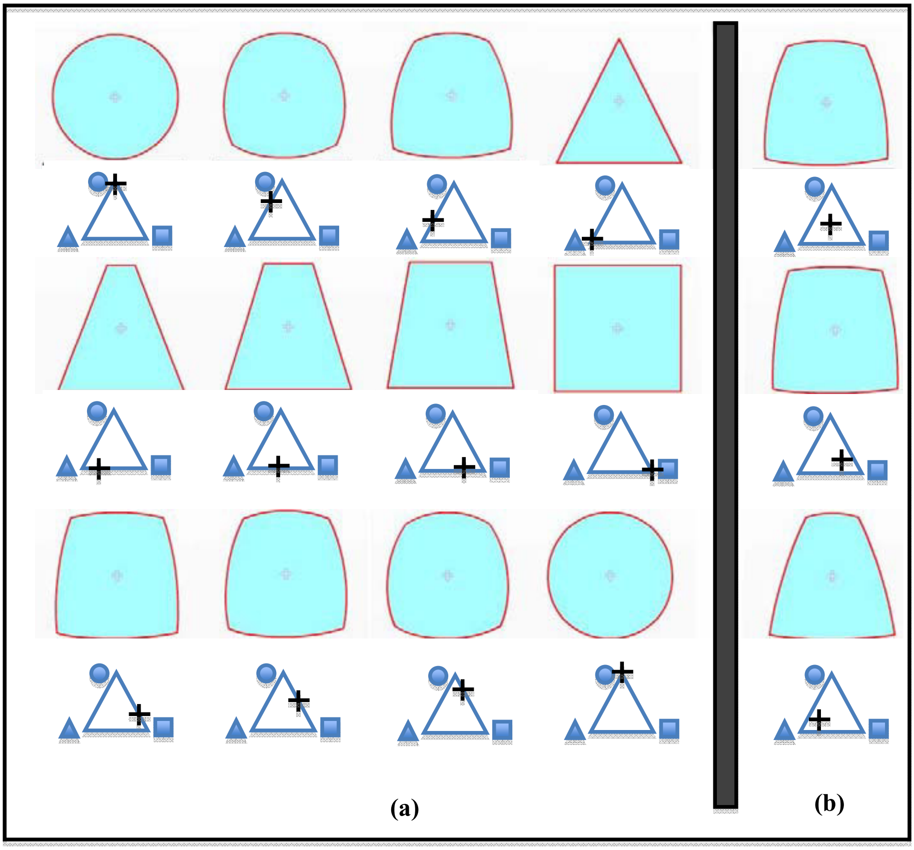


Fig. 3. Smooth shape transition with consistent interpolation. (a) Blending two shapes. (b) Blending all three shapes.

like PCA or some compact shape signature. Usually, these transformations make the data non-intuitive and thus tracking the learning algorithm becomes very difficult.

As shown in Fig. 2, we store each of the three normalized primitive shapes as a set of eight quadratic Bezier curves. The solid points represent segment boundaries and the ragged blotches represent mid-segment control points. Note how a null segment (1-2) had to be created for the apex of the triangle. The reason why our piece-wise curve segments work so well, is that we were able to carefully identify the corresponding segments for the diverse topologies of circle, triangle and square. As a result, even under simple linear interpolation, we do not notice any tears or inconsistent shapes.

The normalized shapes can be affine transformed to any location, scale and rotation. Finally, the shape weights are applied to blend the corresponding Bezier control points, to yield an in-between shape. Note that start-end-mid control points of only corresponding segments are interpolated, as shown in Eqns. 1 and 2.

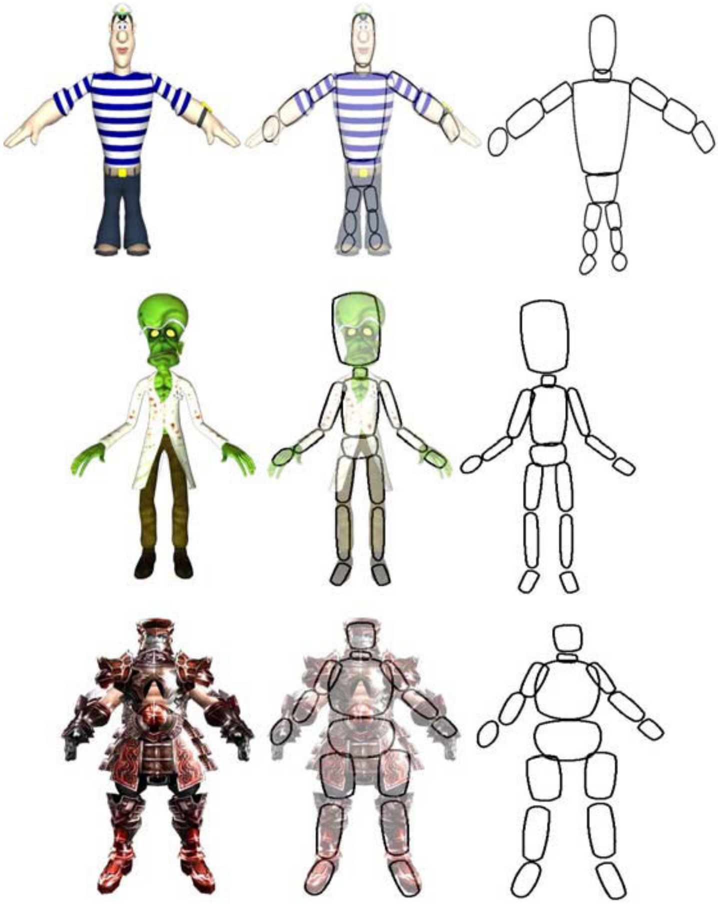


Fig. 4. Expressive vector fitting of body parts

$$p_j^{'} = \sum_{i=1}^{3} (w_i . p_{i,j}) \ . \tag{1}$$

$$m_j^{'} = \sum_{i=1}^{3} (w_i . m_{i,j}) \tag{2}$$

$$\text{where, } \sum_{i=0}^{2} w_i = 1$$

$$\text{And, } j \in \{1,2,3,4,5,6,7,8\}$$

In the above equations, p'_j and m'_j represent the *j-th* blended segment boundary and midpoints respectively, while $p_{i,j}$, and $m_{i,j}$ represent the corresponding control points in the *i-th* primitive shape (circle, triangle, square). w_i is the weight contribution from the *i-th* primitive shape. Results of some blend operations are shown in Fig. 3. The cross hair under the shapes indicate the shape weights.

Fig. 4 illustrates the expressive vector shape fitting of diverse character shapes. As evident from the warrior character in the last row, accessories and loose clothing pose challenges in extracting the true body proportions. In such situations, only a human artist can make an educated guess on where the actual body part lies. This is also true for hidden or fore-shortened body-parts in posed character images, which is usually the case for characters from released games and films.

6 Training and Experimental Results

6.1 Cleaning

To describe each body part in our shape vector representation we need 8 parameters such as, the coordinate of the center, height, width, orientation and three shape weights for circle, triangle, and squares. Thus for 16 body parts we have 128 features. As any body part can be at any orientation and the personality traits and physicality are posture invariant we ignore the rotation of the body parts. Also since we are more interested in the proportion data, the exact location of the cage center can also be disposed of as we have the cage length and breadth parameters. For the shape weights; since we use normalized barycentric coordinates, we can omit one of the shape weights from our feature list, as the three weights add up to 1. In our case we omit the triangle weight. Thus finally the height, width, circles weight and square weight these four features are used for further mining steps.

6.2 SVM Classification

We applied SVM to our vector annotated shapes, which were also manually labeled into classes. For this paper, we tested three physical class labels: *weak, strong* and *average*. To validate the model, we split our collection into a training set (247 characters) and an evaluation set (68 characters). We also implemented a control experiment, using a simple low-level distance measure to represent the same shapes with boundary distance from the body part centroid. This helped us objectively find out how our method compares to those that use low-level features in their training model. As can be seen from Tables 1-3, our primitive shape representation performs much better in terms of correctly classified instances and overall decent values for precision and recall for each class.

Table 1. SVM with primitive shape transform outperforms naïve low level representation

	Primitive Vector Representation	**Centroid-Boundary Distance (r, θ)**
Correctly Classified Instances	61 (89.71%)	32 (47.06%)
Incorrectly Classified Instances	7 (10.29%)	36 (52.94%)
Kappa statistic	0.8454	0.2098
Mean absolute error	0.2451	0.3987
Root mean squared error	0.3096	0.4973
Relative absolute error	55.19 %	89.12 %
Root relative squared error	65.71 %	104.27 %
Total Number of Instances	68	68

In table 2 the TP rate is true positive rate, FP rate is false positive rate

$$tp = \frac{TP}{TP + FN} \dots\dots\dots\dots \quad (3)$$

$$fp = \frac{FP}{FP + TN} \dots\dots\dots\dots \quad (4)$$

Now, to calculate TP rate for class strong in Table 2 primitive vector representation we get from the confusion matrix of Tabl 3 that TP = 19, and FN = 2+1 = 3. So, from equation (3) we get $tp = \frac{19}{19+2+1} = 0.86363636$. Similarly we calculate all the tp and fp values in Table 2. For the F-Measure we use the equation

$$F - measure = \frac{2 \times recall \times precision}{recall + precision} \dots\dots\dots\dots \quad (5)$$

Fig. 5 shows the visual validation results of our model. A sample is shown from the training set in the first column, two correct classification results from the second column, and one incorrect result is shown in the last column. We feel that these results are qualitatively acceptable, and even the incorrect labels are not blatantly wrong.

Table 2. Accuracy Results by Class

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
Primitive Vector Representation	0.864	0.043	0.905	0.864	0.884	0.965	**strong**
	0.864	0.065	0.864	0.864	0.864	0.918	**avg**
	0.958	0.045	0.92	0.958	0.939	0.968	**weak**
Weighted Avg.	0.897	0.051	0.897	0.897	0.897	0.951	
	TP Rate	**FP Rate**	**Precision**	**Recall**	**F-Measure**	**ROC Area**	**Class**
Centroid-Boundary Distance (r, θ)	0.385	0.143	0.625	0.385	0.476	0.659	**strong**
	0.4	0.188	0.471	0.4	0.432	0.595	**avg**
	0.636	0.457	0.4	0.636	0.491	0.593	**weak**
Weighted Avg.	0.471	0.257	0.507	0.471	0.468	0.619	

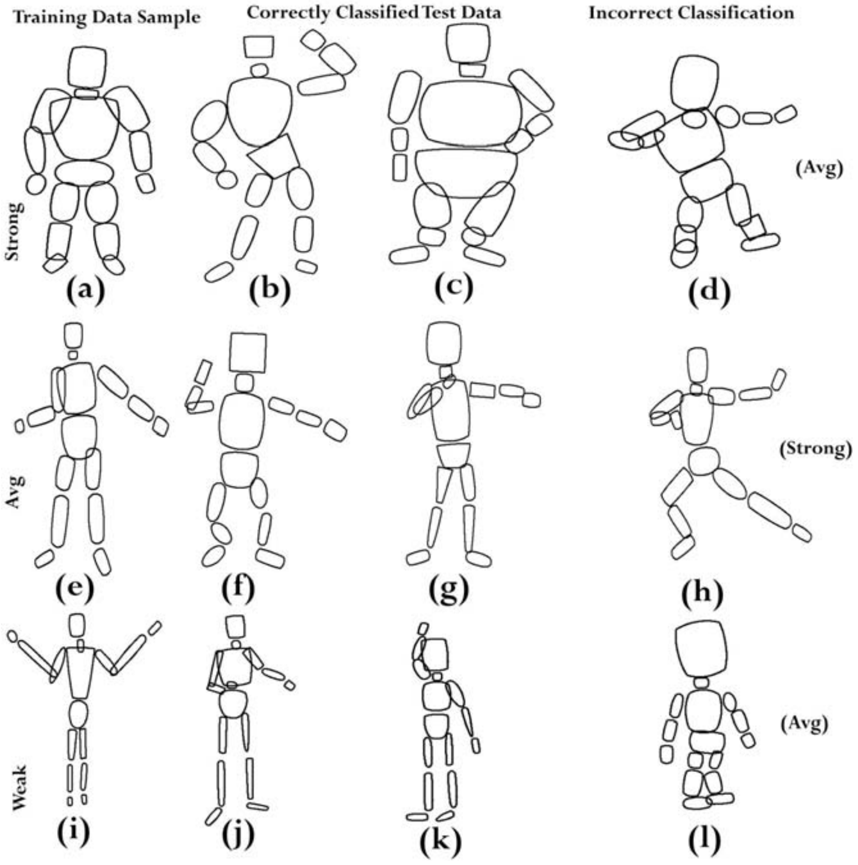


Fig. 5. Visual results of validation tests on ground truth

Table 3. Confusion Matrix

Primitive Vector Representation			
Strong	**Average**	**Weak**	← Classified as
19	2	1	**Strong**
2	19	1	**Average**
0	1	23	**Weak**
Centroid-Boundary Distance (r, θ)			
Strong	**Average**	**Weak**	← Classified as
10	4	12	**Strong**
3	8	9	**Average**
3	5	14	**Weak**

7 Conclusion

This paper describes a new method of representing arbitrary shapes using a blend of circle, triangle and squares. It uses consistent interpolation of quadratic Bezier curves. We have achieved a decent precision and recall rate for our SVM training model, and significantly outperform an example low-level data transformation. We hope to add to our database, and mine relationships between labels as well. One of the limitations of our representation is that it is symmetric about its medial axis, and also that it cannot represent concave surfaces. We are currently working on these limitations, by allowing more than one primitive shape to be fitted to a body part. We are also working on exciting applications in warping and shape deformation that will further empower procedural generation and design reuse.

References

1. Bancroft, T.: Creating Characters with Personality, ISBN: 0-8230-2349-4
2. Beiman, N.: Prepare to Board!: Creating Story and Characters for Animated feature
3. Camara, S.: All about techniques in drawing for animation production, 1st edn. Barron's Education Series, Inc. (2006)
4. Edelman, S., Intrator, N.: Learning as extraction of low-dimensional repre-sentations. In: Medin, D., Goldstone, R., Schyns, P. (eds.) Mechanisms of Perceptual Learning, vol. 36, pp. 353–380. Academic Press, London (1997)
5. Gal, R., Shamir, A., Cohen-Or, D.: Pose-Oblivious Shape Signature. IEEE Transactions on Visualization and Computer Graphics 13(2), 261–271 (2007)
6. Garrett, L.: Visual design: A Problem-Solving Approach
7. Gil-Jiménez, P., Lafuente-Arroyo, S., Maldonado-Bascón, S., Gómez-Moreno, H.: Shape Classification Algorithm Using Support Vector Machines for Traffic Sign Recognition. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 873–880. Springer, Heidelberg (2005)
8. Gooch, B., Reinhard, E., Gooch, A.: Human facial illustrations: Creation and psychophysical evaluation. ACM Transactions on Graphics (TOG) 23(1), 27–44 (2004)
9. Hart, C.: Cartoon Cool: How to Draw New Retro-Style Characters

10. Hsu, C.-H., Wang, M.J.: Using decision tree-based data mining to establish a sizing system for the manufacture of garments. The International Journal of Advanced Manufacturing Technology 26(5-6) (September 2005)
11. Islam, T., Why, Y.P., Ashraf, G.: Mining Human Shape Perception with Role Playing Games. CGAT, Singapore, (to appear, 2010)
12. Judd, T., Durand, F., Adelson, E.: Apparent ridges for line drawing. ACM Trans. Graph. 26(3), article 19 (2007)
13. Liu, J., Chen, Y., Gao, W.: Mapping Learning in Eigenspace for Harmonious Caricature Generation. In: Proceedings of the 14th annual ACM international conference on Multimedia (2006)
14. Marchenko, Y., Chua, T.S., Jain, R.: Ontology-Based Annotation of Paintings Using Transductive Inference Framework. MMM (1), 13–23 (2007)
15. Meyer, M., Anderson, J.: Key Point Subspace Acceleration and soft caching. ACM Trans. Graph. 26(3), article 74 (2007)
16. Pizlo, Z.: 3D Shape: Its Unique Place in Visual Perception. MIT Press, Cambridge (2008) ISBN: 978 0262162517
17. Stathopoulou, E., Alepisa, G.A., Tsihrintzisa, Virvoua, M.: On assisting a visual-facial affect recognition system with keyboard-stroke pattern information. In: I.-OResearch and Development in Intelligent Systems, vol. XXVI, pp. 451–463.
18. Thorne, M., Burke, D., van de Panne, M.: Motion doodles: An Interface for sketching character motion. In: Marks, J. (ed.) ACM SIGGRAPH 2004 Papers. Los Angeles, California, August 08-12, pp. 424–431. ACM, New York (2004)
19. Toll, D.: You Can Draw. Hinkler Books, ISBN: 978-1-7415-7610-8
20. Ueda, N., Suzuki, S.: Learning Visual Models from Shape Contours Using Mul-tiscale Convex/Concave Structure Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(4), 337–352 (1993)
21. Vapnik, V.N.: The Nature of Statistical Learning Theory (Information Science and Statistics). Springer, Heidelberg (1999)
22. Vogel, H.L.: Entertainment Industry Economics: A Guide for Financial Analysis, 7th edn. Cambridge University Press, Cambridge
23. Wang, R.Y., Pulli, K., Popović, J.: Real-time enveloping with rotational regression. ACM Transactions on Graphics (TOG) 26(3) (2007)
24. Waterman, A.D.: A guide to expert systems. The teknowledge series in knowledge engineering. Addison-Wesley, Reading (1986)
25. Zhang, M.: Mining small objects in large images using neural networks. Tech. rep., Victoria University of Wellington, School of Mathematical and Computing Sciences (2005)

Mining Relationship Associations from Knowledge about Failures Using Ontology and Inference

Weisen Guo and Steven B. Kraines*

Science Integration Program (Human)
Department of Frontier Sciences and Science Integration
Division of Project Coordination
The University of Tokyo
277-8568, Kashiwa, Japan
{gws,sk}@scint.dpc.u-tokyo.ac.jp

Abstract. Mining general knowledge about relationships between concepts described in the analyses of failure cases could help people to avoid repeating previous failures. Furthermore, by representing knowledge using ontologies that support inference, we can identify relationships between concepts more effectively than text-mining techniques. A relationship association is a form of knowledge generalization that is based on binary relationships between entities in semantic graphs. Specifically, relationship associations involve two binary relationships that share a connecting entity and that co-occur frequently in a set of semantic graphs. Such connected relationships can be considered as generalized knowledge mined from a set of knowledge resources, such as failure case descriptions, that are formally represented by the semantic graphs. This paper presents the application of a technique to mine relationship associations from formalized semantic descriptions of failure cases. Results of mining relationship associations in a knowledge base containing 291 semantic graphs representing failure cases are presented.

Keywords: Relationship Associations, Semantic Relationships, Ontology, Logical Inference, Failure Knowledge, Graph Mining, Frequent Pattern Mining, Knowledge Discovery.

1 Introduction

Much of the progress of human society is based on the successful experiences of human activities. However, there are also important lessons that could be learned from failures. Scientists have realized that the knowledge that is generated from the analyses of mechanisms behind failures and "near misses" could be useful for avoiding similar mistakes in the future [19]. In the spirit of this realization, in 2001 the Japan Science and Technology Agency developed a failure knowledge database (hereafter the failure knowledge database) and deployed it on the Web to the public [11].

* Corresponding author.

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 617–631, 2010.

There is a lot of research on description and analysis of knowledge about failures in the literature. Hatamura et al. presented a data structure and an effective way for displaying information about failure cases to the users [7]. Nakao et al. developed an associative search method to mine knowledge about failures that is effective for risk management [16].

The question we ask here is: “based on this data about concrete cases of failure occurrences, how can we obtain more generalized knowledge about failures?” This problem has been addressed in some earlier studies. For example, Guo and Kraines developed a method for using graph mining to find common semantic patterns in a set of semantic graphs representing cases in the failure knowledge database [4].

In this paper, we consider a special kind of general knowledge pattern called a “relationship association”. Relationship associations take the following form: if a particular entity *e1* has a relationship *r1* with another entity *e2*, then it is likely that *e1* has another relationship *r2* with a third entity *e3*. In other words, if a particular entity has one specific relationship with some other entity, then it is likely that the entity has a second specific relationship with another entity. These kinds of patterns mined from the failure knowledge database may be useful for people to analyze the reasons behind a new failure occurrence, or to avoid a potential failure from occurring. For example, using the techniques described in this paper, we have mined the following relationship association from the failure knowledge database: “if an operation has failure type ‘poor safety awareness’, then it is likely that this operation has also another failure type ‘poor planning by organization’”. Upon seeing a new failure occurrence where the operation had some kind of “poor safety awareness”, we can suggest from this relationship association the possibility that this operation is also a kind of “poor planning by organization”. We present a method in this paper to mine relationship associations from the information on failure cases in the failure knowledge database.

This paper is organized as follows. In Section 2, we describe the background of this paper. In Section 3, we present our method for mining relationship associations, which uses an inference engine to evaluate if the relationship associations occur in a semantic graph. We also describe a method for selecting potentially interesting relationship associations. In Section 4, we present results of experiments to mine relationship associations from a set of semantic graphs representing cases in the failure knowledge database. We conclude this paper with a discussion of related work and a summary.

2 Background

Most of the information for the failure cases in the failure knowledge database is in unstructured natural language text format. In order to mine semantic relationships automatically, we must convert this natural language text into a structured format. We have chosen to represent the failure cases in OWL-DL, the Web Ontology Language based on description logics that is recommended by the W3C.

We have used the EKOSS (Expert Knowledge Ontology-based Semantic Search) web-based knowledge sharing system (www.ekoss.org) to author the semantic graphs. EKOSS uses domain ontologies formalized in description logics (DL) as the knowledge representation languages for the semantic graphs. It also provides a set of authoring tools for helping users to create semantic graphs representing their knowledge resources and search conditions, tools that include some features for semi-automating the creation process.

Each domain ontology contains a set of classes representing the concepts of this domain and a set of properties representing the types of relationships that can hold between these concepts. Each knowledge resource shared on the EKOSS system is represented by a semantic graph based on a domain ontology. Each semantic graph is composed of nodes representing instances of ontology classes together with arcs representing relationships between the instances. Relationship types are specified by the properties in the ontology. Each instance can have a descriptive text label.

The users generate their semantic graphs using the EKOSS authoring tools and save them in the EKOSS knowledge base. The users can also conduct semantic searches of the knowledge base by creating search queries using the same authoring tools. The EKOSS reasoner uses a logical inference engine, such as RacerPro (www.racer-systems.com) or JTP (the Java Theorem Prover), to infer semantically implied matches between queries and semantic graphs. More details are given in [13].

A corpus of 291 semantic graphs was constructed by research assistants with engineering expertise using the SCINTENG ontology and the EKOSS system. Each semantic graph represents one case in the failure knowledge database. The SCINTENG ontology is an OWL-DL ontology for representing concepts from a wide range of engineering domains. The main classes are divided into seven categories: 1) materials and energy substances; 2) activities and phenomena, including human activities and natural activities; 3) physical objects, including artificial and natural physical objects; 4) spatial location where an activity can occur or a physical object can exist; 5) events that mark the beginning or end of activities and physical objects; 6) actors that can cause activities to occur; and 7) classes of activities, including method of activity, human failure activity, and organization failure activity.

Fig. 1 shows a semantic graph, containing 17 instances of classes from the SCINTENG ontology together with 23 relationships between the instances, which was created to describe a case in the failure knowledge database entitled “As a result of a signal error, the train was stopped.”

The EKOSS reasoner can evaluate if a semantic graph matches with a search query at five levels of complexity. The levels are listed below in order from easy but less accurate to complex but more accurate:

1. Matching classes (nodes) in the query to instances of the same classes in the semantic graphs.
2. Matching classes (nodes) in the query to instances of the same or subsumed classes in the semantic graphs.
3. Matching classes (nodes) in the query to instances of the same or subsumed classes in the semantic graphs having the same or subsumed properties (links) between them.
4. Matching classes (nodes) in the query to instances of the same or subsumed classes in the semantic graphs that can be inferred using logical inference to have the same or subsumed properties (links) between them.
5. Matching classes (nodes) in the query to instances of the same or subsumed classes in the semantic graphs that can be inferred using logical and rule-based inference to have the same or subsumed properties (links) between them.

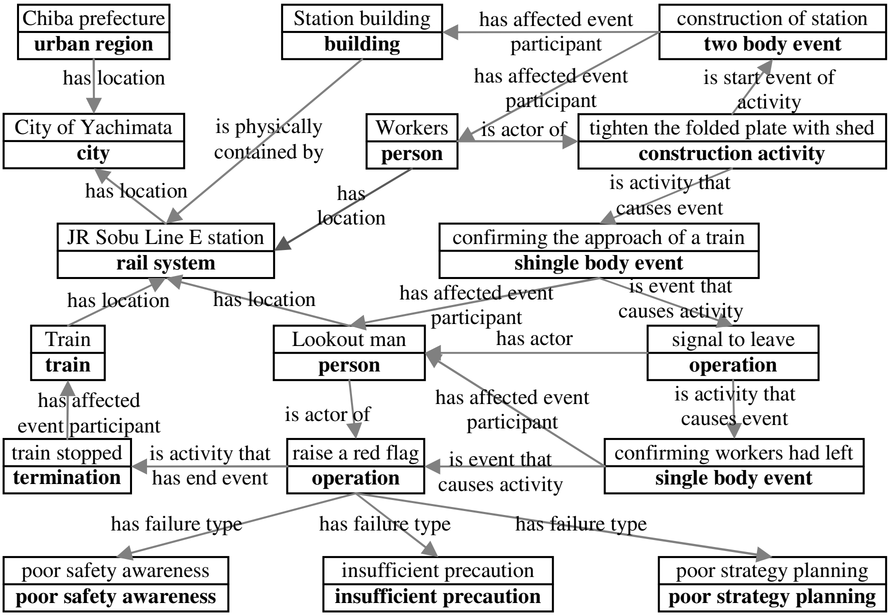


Fig. 1. The semantic graph for the failure case "As a result of a signal error, the train was stopped". Boxes show instances of classes from the domain ontology. The text above the line in a box is the instance label. The text in bold type below the line in a box is the class name of that instance. Arrows show properties expressing the asserted relationships between instances.

We note that all five levels differ from text matching, such as calculating the similarity of two strings [2], in that by using the logical structure of the ontology, each level can provide a measure of the semantic similarity between descriptions.

The first level just determines if the exact classes of a query occur in a semantic graph. Because this way matches concepts of entities rather than labels of entities, it is more semantic than string matching. The second level gives increased matching recall by considering the subsumption hierarchy of a class: if subclasses of the classes in a query occur in a semantic graph, the graph is said to match. At the third level, we consider relationships that have been asserted between the classes in the query, which decreases the number of matching results because the matches must also satisfy the specified relationships between the classes. At the fourth level, we use logical inference to identify implicit relationships between entities based on the logical characteristics of properties such as symmetric, transitive, and inverse. By using logical inference to find hidden relationships that are not explicitly stated between entities, this level increases the recall of matching results. The fifth level uses both logical and rule-based inference. Rule-based inference is accomplished by giving the reasoner a set of rules that are provided by domain experts prior to the matching process. Therefore, the fifth level of semantic matching is expected to achieve the highest accuracy. Details of the semantic matching technique for evaluating the similarity of two semantic graphs are given in [3].

In previous work, we have described a technique for mining common semantic patterns from this corpus of semantic graphs [4]. We used the following graph mining method to discover the semantic patterns that represent generalized knowledge about failures. First, all unique sub-graphs in the corpus are created from information given by SCINTENG ontology. Second, each sub-graph is matched with the corpus of semantic graphs, and the support is obtained. Sub-graphs having a support of more than a minimum threshold are designated as common semantic patterns.

The graph mining algorithm is based on the "Apriori" algorithm [8]. As a consequence, the method does not analyze the co-occurrence of relationships between two sub-graphs. Furthermore, it can only discover common sub-graphs, and the common semantic patterns discovered are often too general to be interesting for the following reason. Because the matching set uses subsumption inference, the sub-graphs with higher level classes and properties will have greater support than the sub-graphs with lower level classes and properties. The "Apriori" algorithm filters out the rare sub-graphs that do not meet the required support. However, often we are most interested in the semantic patterns that are less common but include more specific classes.

We have developed a technique for mining relationship associations in life science [5], [6]. This paper presents an improvement of that work. Specifically, we have added a method for considering association relationships that occur when the order of the two relationships is reversed. And also we added a method to remove the relationship association queries which contain one triple that is subsumed by the other triple. We have then applied the improved process to mine relationship associations from knowledge about failures using ontology and inference.

3 Relationship Associations Mining

In order to mine relationship associations from descriptions of failure cases in natural language text, there are four main tasks that we must address. First, we need an appropriate representation schema to convert the unstructured text to structured knowledge. Second, we need to identify the relationship associations occurring in a specific failure case, including those that can be inferred from the semantics of the description of the case. We have addressed these two aspects in our previous work, as described in Section 2. The third task we must address is to develop a fast and effective mining procedure that obtains potentially interesting results. Often, there will be many relationship associations that are mined. So the fourth task is to identify the most interesting and important relationship associations. Although this final selection task must involve human judgment, we need to reduce the burden on humans as much as possible. So some automatic pre-selection method is needed to filter out the more uninteresting results. In this section, we present our work to address the last two tasks of the relationship association mining process.

The procedure that we have developed for mining relationship associations, corresponding to the third and fourth task, takes the set of semantic graphs as the input. The output is a set of relationship associations in the form of linked pairs of semantic triples.

First, we generate triple queries from the set of semantic graphs.

A semantic triple - consisting of a domain instance, a range instance, and a property that connects the domain instance to the range instance - is the minimum unit of a semantic graph. Each semantic graph contains one semantic triple for each property in the graph. For each triple in a semantic graph, we create one triple query by converting the instances of the triple into variables with the same classes. In essence, a triple query represents the asserted relationship between two specific entities made by the triple as a generalized relationship between ontology classes.

Relationship associations must be comprised of two linked triple queries whose triples both appear in the same semantic graph and share a common entity. So even though there may be duplicate triple queries generated from a set of semantic graphs, at this point we keep all of the generated triple queries.

Second, we match each triple query with each semantic graph to get the support for that query and discard queries with insufficient support.

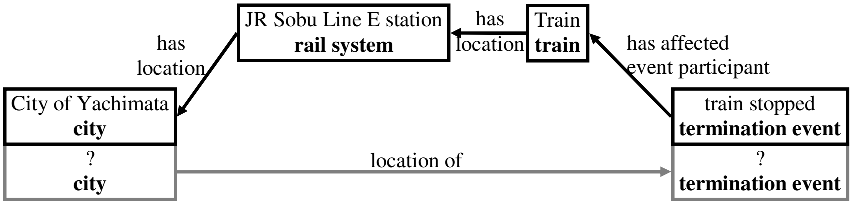


Fig. 2. An example of semantic matching. The part outlined in black is from the semantic graph. The part outlined in gray is the query. Class variables are shown with boxes where the first line of text is "?" and the second line is the ontology class. See Fig.1 for meanings of other symbols.

In the work reported here, we evaluate the match between a semantic query and a semantic graph by using the fifth level of semantic matching supported by the EKOSS reasoner as described in section 2. The matching process is executed as follows. First, we add the semantic graph to the reasoner's knowledge base together with the ontology used to create the graph. Then, we use the reasoner to determine if the query matches the semantic graph. Consider the result of matching the triple query "find some instance of a **city** that is the location of some instance of **termination event**" against the semantic graph shown in Fig. 1. We can get the matching result shown in Fig. 2 only by using the fifth level of semantic matching, because the match depends on the logical axiom in the ontology that "location of" is the inverse of "has location", the transitivity of the property "has location", and the rule that states "if A has affected event participant B and B has location C, then A has location C."

Using the EKOSS reasoner at the fifth level semantic matching mode, we match all triple queries with all semantic graphs to find the number of semantic graphs in which each triple query occurs. If a triple query only occurs in one semantic graph, then it cannot be involved in a relationship association. Therefore, we remove the triple queries occurring only in one semantic graph. The remaining triple queries are used to create relationship association queries in the next step.

Third, we generate relationship association queries.

We create relationship association queries from the set of triple queries generated in the previous steps as follows. For each graph, we find all pairs of triples that share one instance, e.g. that form a connected quintuple with three instances and two properties. If both of the corresponding triple queries are in the set of triple queries generated in previous step, then we use this pair of triples to create a relationship association query.

Fourth, we remove the duplicate relationship association queries that have same semantic meaning.

Because we use semantic matching to match a relationship association query with a semantic graph, two relationship association queries with the same semantic meaning will get the same matching results. The evaluation of matches between the relationship association queries and the semantic graphs using the EKOSS reasoner is the computationally most expensive step in the mining process. By removing relationship association queries with the same semantic meaning, we can reduce the number of reasoning tasks that must be performed.

Three types of relationships in description logics can be considered to imply semantic equivalence. We consider the applicability of each to our mining task in the following.

Subsumption: All of the relationship association queries have the same structure of three classes and two properties. Therefore, if the two properties in *query1* are subproperties of *query2* and each of the classes in *query1* are sub-classes of the classes in *query2* that have the same connectivity, then we can say that *query2* subsumes *query1*. In this case, all of the semantic graphs that match with *query1* will also match with *query2*, which implies some kind of semantic equivalence. However, the reverse is not true: there may be some semantic graphs that match with the more general *query2* but not the more specific *query1*. Furthermore, the more specific query contains more information about a possible relationship association than the general query, as we discussed in section 2. Therefore, we do not consider this characteristic as implying semantic equivalence here.

Symmetric: From the definition of OWL-DL [17], if a property *p* is symmetric, then if *x* is related to *y* by *p*, then *y* is also related to *x* by *p*. This means that if property *p* is symmetric, then the query (*c1 –p> c2*) and query (*c2 –p> c1*) have exactly the same semantic meaning. Therefore, we consider this characteristic to imply semantic equivalence.

Inverse: From the definition of OWL-DL [17], if the property *p1* is stated to be the inverse of the property *p2*, then if *x* is related to *y* by *p2*, then *y* is related to *x* by *p1*. Therefore, we can say that if property *p1* is the inverse of property *p2*, then the query (*c1 –p1> c2*) and query (*c2 –p2> c1*) have exactly the same semantic meaning, and so we can consider this characteristic also to imply semantic equivalence.

We use the following notation to illustrate the process of removing duplicate relationship association queries. One relationship association query is comprised of two triple queries that share one connecting class. We use C_{1d} to indicate the domain class of the first triple query, P_1 to indicate the property of the first triple query, C_{1r} to indicate the range class of the first triple query, C_{2d} to indicate the domain class of the second triple query, P_2 to indicate the property of the second triple query, C_{2r} to indicate the range class of the second triple query, and C_c to indicate the connecting class of the two triple queries.

We use the algorithm presented in [6] to remove the duplicate relationship association queries with same semantic meaning based on the two logical characteristics, inverse and symmetric. However, based on examination of the results using that algorithm, we found that the relationship association queries where one triple is subsumed by the other are also unlikely to be interesting candidates for relationship associations. Consider, for example, the relationship association "if one **operation** has failure type **poor safety awareness**, then it is possible that it has failure type **poor value perception by organization**." Because we already know that **poor safety awareness** is a subclass of **poor value perception by organization**, it is clear that this relationship association does not tell us anything useful. Therefore, we have created a second algorithm to remove this kind of relationship association query. The new algorithm considers subsumption, inverse and symmetric characteristics of the two triples making up a relationship association query. Because we are looking for the condition where the property and the classes of the first triple subsume or are subsumed by those of the second, we only need to check if the property of first triple is inverse or symmetric.

```
Algorithm. Removing associations with duplicate triples
  For each relationship association query QQ
    If (($C_{1d} \subseteq C_{2d}$) and ($P_1 \subseteq P_2$) and ($C_{1r} \subseteq C_{2r}$)) or
       (($C_{2d} \subseteq C_{1d}$) and ($P_2 \subseteq P_1$) and ($C_{2r} \subseteq C_{1r}$))
      Then remove QQ
    Else If $P_1$ has an inverse INV($P_1$) and
      ((($C_{1r} \subseteq C_{2d}$) and (INV($P_1$) $\subseteq P_2$) and ($C_{1d} \subseteq C_{2r}$)) or
      (($C_{2d} \subseteq C_{1r}$) and ($P_2 \subseteq$ INV($P_1$)) and ($C_{2r} \subseteq C_{1d}$)))
      Then remove QQ
    Else If $P_1$ is symmetric and
      ((($C_{1r} \subseteq C_{2d}$) and ($P_1 \subseteq P_2$) and ($C_{1d} \subseteq C_{2r}$)) or
      (($C_{2d} \subseteq C_{1r}$) and ($P_2 \subseteq P_1$) and ($C_{2r} \subseteq C_{1d}$)))
      Then remove QQ
  End For
```

Here, the symbol "$\subseteq$" indicates inclusive subsumption (A $\subseteq$ B means that A and B are equivalent classes or A is subsumed by B). After applying these algorithms, we get a set of potentially meaningful relationship association queries with unique semantics.

Fifth, we match each of the remaining relationship association queries with each semantic graph to obtain their supports.

The matching method described in the second step used to match the relationship association queries with each of the semantic graphs and calculate the number of graphs in which they occur. Relationship association queries that only occur in one semantic graph cannot be considered as relationship associations, so they are removed. The rest of the relationship association queries are candidates for relationship associations.

Sixth, we select the potentially interesting relationship associations and represent them in natural language.

The previous steps may produce a large number of candidates for relationship associations. Here we use a probabilistic approach to identify the relationship associations that have the most potential for being interesting. Specifically, we use the probability that the first triple of a relationship association query occurs in the set of semantic graphs, $P(t_1)$, together with the two conditional probabilities for each relationship association query defined below:

- The probability that the second triple of a relationship association query occurs if the first triple occurs, $P(t_2|t_1)$.
- The probability that the second triple of a relationship association query occurs if the connecting class occurs, $P(t_2|cc)$.

We use the following reasoning to identify potentially interesting relationship associations. As we discussed in section 2, the triples in the relationship associations can have a wide range of specificity. For example, a triple such as "a **physical object** participates in a **human activity**" will match with almost all of the semantic statements. A relationship association that is comprised of two commonly occurring triples such as this is not likely to be interesting because it does not tell us anything new or specific. Therefore, our first criterion is that the first triple in a relationship association query must occur relatively infrequently in the corpus, which means that $P(t_1)$ must be relatively small.

Our second criterion is that if such a triple t_1 exists in a relationship association query with another triple t_2, then the probability that t_2 occurs in just the semantic graphs that contain t_1 must be much larger than the probability that t_2 occurs in all of the semantic graphs that contain the connecting class. The reasoning behind this selection criterion is as follows. An interesting relationship association should tell us that the two triples tend to occur more frequently together than alone. However, although some t_2's may not occur very often in the entire set of semantic graphs, they may occur with a high probability whenever the connecting class occurs. We will not be interested in these relationship associations either, because if simply the occurrence of the connecting class is sufficient to assure a high probability of the occurrence of t_2, then the presence of t_1 is not required.

We select the relationship associations that meet both of these criteria to be reviewed by human experts. Using the notation above, this means:

$$P(t_1) << 1 \text{ and } P(t_2| t_1) / P(t_2|cc)>> 1$$

We can evaluate the second criterion using the following condition:

$$(S_r/S_{t1})/(S_{t2}/S_{cc}) >> 1$$

where S_{ti} is the support of triple t_i, S_r is the support of relationship association r, and S_{cc} is the support of connecting class cc (proof is given in [6]).

4 Experiments

As described in Section 2, we have constructed a corpus of 291 semantic graphs representing cases from the failure knowledge database. We have conducted experiments to obtain relationship associations from this corpus using the method described in section 3. In this section, we report the results of this experiment.

Of the 1264 classes and 236 properties in the SCINTENG ontology, 491 classes and 107 properties were used in the 291 semantic graphs. On average, each semantic graph has 24 instances and 31 properties. The entire set of semantic graphs contains 7115 instances and 9151 properties. However, 296 properties are data type properties that we ignore here. Therefore, we created 8855 (= 9151 – 296) triple queries from the 291 semantic graphs using the process described in Section 3. We then used the EKOSS reasoner to determine how many semantic graphs contain each triple. We removed all triple queries that only matched with one semantic graph, the graph from which the triple was obtained, which left 7381 triple queries available for creating relationship association queries.

We created 15447 relationship association queries from the 7381 triple queries and 291 semantic graphs as described in Section 3. After removing semantic duplicates using the method from Section 3, 10137 relationship association queries remained.

We matched these relationship association queries with all of the semantic graphs using the EKOSS reasoner and removed all relationship association queries that only appeared once. This resulted in a total of 6138 relationship association queries appearing in at least two of the semantic graphs.

We calculated the probabilities of selection criteria for these 6138 relationship association queries. We used a cutoff value of $P(t_1) \leq 0.1$ for the first selection criterion, which means that the first triple in the relationship association query must occur in at least 29 semantic graphs because the size of the corpus is 291. We used a cutoff value of $P(t_2|t_1)/P(t_2|cc) \geq 2$ for the second selection criterion, which means that the probability of a relationship association query occurs when the first triple occurs must be twice as the probability of the second triple occurs when the connecting class occurs. Here, we consider the order of the two triples in each relationship association, so the size of relationship association queries before applying selection criteria is 12276 (= 6138 × 2). A total of 2687 relationship associations met these selection criteria.

Finally, in order to improve readability of these selected relationship associations, we converted them from the quintuple format into natural language using a previously developed natural language generation algorithm [12].

Table 1 shows a sample of ten of the 2687 relationship associations together with their selection criteria.

Three relationship associations mined in this experiment that were judged to be interesting are shown in Fig. 3. The natural language representations are given as follows:

Relationship association (a): "If an **operation** has failure type **poor safety awareness**, then the **operation** is likely to have failure type **poor planning by organization**."

Relationship association (b): "If a **mass body freefalling** has activity participant **person**, then the **person** is likely to be on the top of an **artificial fixed object**."

Relationship association (c): "If a **destruction event** is the end of event of a **chemical reaction**, then the **destruction event** is likely to have as a changed event participant a **container**."

Insights into the nature of these relationship associations can be obtained by examining the semantic graphs for the failure cases that contain them. Due to limited space, we just list the seven failure cases containing the relationship association (c). Class names are shown in bold and instance names are shown in italics.

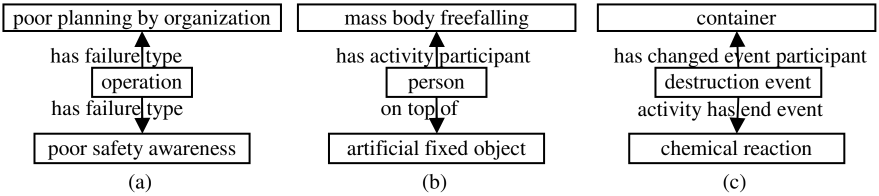


Fig. 3. Three examples of relationship associations

Table 1. Ten relationship associations and their selection criteria. Each triple is shown in the form “domain class | property | range class”. The conditional triple (t_1) is separated from the consequent triple (t_2) using “>”. The connecting class is shown in bold type. S_r denotes the number of semantic graphs in which the relationship association occurs. S_{t1} denotes the number of semantic graphs in which triple t_1 occurs. S_{t2} denotes the number of semantic graphs in which triple t_2 occurs. S_{cc} denotes the number of semantic graphs in which the connecting class occurs.

Relationship Association	S_r	S_{t1}	S_{t2}	S_{cc}	$P(t_1)$	$P(t_2 \mid t_1)/$ $P(t_2 \mid cc)$
operation | has failure type | poor safety awareness > **operation** | has failure type | poor planning by organization	8	18	22	164	0.06	3.4
mass body freefalling | has activity participant | **person** > **person** | on top of | artificial fixed object	7	14	13	200	0.05	7.1
chemical reaction | has end event | **destruction event** > **destruction event** | has changed event participant | container	7	20	14	81	0.07	2.1
operation | has failure type | disregard of procedure > **operation** | has failure type | poor safety awareness	8	29	18	164	0.10	2.2
chemical activity | has activity class | poor management > **chemical activity** | has activity class | poor safety awareness	7	12	21	100	0.04	2.8
physical activity | has activity class | disregard of procedure > **physical activity** | has activity class | poor management	7	23	26	246	0.08	2.7
person | on top of | fixed object > mass body freefalling | has actor | **person**	7	15	8	200	0.05	10.0
operation | has failure type | poor safety awareness > **operation** | has failure type | poor strategy planning	7	18	16	164	0.06	3.3
two body event | has affected event participant | person > operation | has end event | **two body event**	6	12	9	41	0.04	2.3
chemical reaction | has activity class | poor management > **chemical reaction** | has activity class | poor safety awareness	6	10	19	96	0.03	3.0

From the failure case “Explosion of 5-t-butyl-m-xylene upon restarting an agitator during the nitration reaction”: “*reaction vessel exploded* is an **artifact destruction event** that has changed event participant a **reaction vessel** called *reaction vessel*, and *promotes the reaction seriously* is a **chemical reaction** that has end event *reaction vessel exploded*.”

From the failure case “Explosion during preparation of manufacturing pesticide caused by stopping of the cooling water due to an error of the failure position setting for a control valve”: “*reaction vessel exploded* is an **artifact destruction event** that has changed event participant a **reaction vessel** called *reaction vessel*, and *caused a runaway reaction* is a **chemical reaction** that has end event *reaction vessel exploded*.”

From the failure case “Explosion caused by local heating of a drum can containing insecticide (chloropyrifosmethyl)”: “*drum exploded* is an **artifact destruction event** that has changed event participant a **tank container** called *drum tank*, and *caused runaway reaction of the material* is a **chemical reaction** that has end event *drum exploded*.”

From the failure case “Explosion and fire of LPG tanks”: “*tank exploded and blew out* is a **destruction event** that has changed event participant an **artificial tank** called *LPG tank*, and *ignited* is a **combustion reaction** that is caused by *tank exploded and blew out*.”

From the failure case “Filling Station Explosion due to LP Gas Discharge from an Overfilled Container”: “*chain explosion of the every size of container* is a **destruction event** that has changed event participants some **steel bottles** called *small and large containers*, and *both horizontal tank for propane and butane exploded* is a **combustion reaction** that causes the event *chain explosion of the every size of container*.”

From the failure case “Rupture of waste liquid container caused due to hypergolic reaction in chemical analysis at LSI factory”: “*the organic solvent reacted with the acid in the container* is a **chemical reaction** that has end event a **destruction event** called *explosion of the container for the acid disposal*, and *explosion of the container for the acid disposal* has changed event participant a **container** called *container for the acid disposal*.”

From the failure case “Silane Gas Explosion at Osaka University”: “*ignition* is a **combustion reaction** that causes a **destruction event** called *explosion*, and *mono silane container* is a **container** that is a participant of *explosion*.”

These seven failure cases are from different areas of engineering. The first is about an explosion on restarting an agitator during the nitration reaction. The second is about an explosion on preparation of manufacturing pesticide. The third is about an explosion caused by a hot spot phenomenon resulting from uneven heating. The fourth is about an explosion and fire of LPG tank. The fifth is about an explosion in a filling station. The sixth is about an explosive reaction between an acid and an organic solvent at a factory. The seventh is about a silane gas explosion at a university. However, the semantic graph of each failure case involves a container (such as reaction vessel, tank, steel bottle), a destruction event (such as explosion, fire), and a chemical reaction (such as combustion reaction, runaway reaction). And furthermore, the relationships given by the relationship association (c) can all be inferred from the actual relationships expressed between the three entities in each semantic graph.

From relationship association (c), we can suggest that when a chemical reaction results in a destruction event, there is likely to be a container that was affected by the event. This kind of relationship association cannot be mined using traditional text mining methods because they cannot support semantic matching at the predicate level.

5 Related Work

The goal of the work presented in this paper is to use ontology and inference techniques to mine generalized knowledge about failures in the form of relationship associations from a corpus of semantic graphs that has been created in previous work.

In the graph mining research field, several algorithms have been developed that can find characteristic patterns and generalized knowledge from large sets of structured data, and even semi-structured or unstructured data.

A method for mining one kind of semantic networks for knowledge discovery from text was presented in [18]. This method used a concept frame graph to represent a concept in the text. A concept frame graph is a simple semantic network with one center concept and some other related concepts. However, the method did not support semantic matching at the predicate level, and the mining goal was concepts, not relationship associations.

The AGM algorithm [8], which was developed to mine frequent patterns from graphs, derives all frequent induced sub-graphs from both directed and undirected graph structured data. The graphs can have loops (including self-loops), and labeled vertices and edges as supported. An extension of AGM, called AcGM [9], uses algebraic representations of graphs that enable operations and well-organized constraints to limit the search space efficiently. An efficient method [10] was proposed to discover all frequent patterns which are not over-generalized from labeled graphs that have taxonomies for vertex and edge labels. However, all of these graph mining methods are restricted to taxonomies and cannot address the special properties of the OWL-DL ontologies that we have used here, such as the logic restrictions and inference rules. Furthermore, the relationship associations that we are interested in are often not frequent patterns. A relationship association occurs when at least two semantic graphs express the association between the two relationships. A pattern in graph mining is considered frequent only if it occurs often in the whole set of graphs. Furthermore, our approach considers the semantics of failure cases at the predicate level to find the implied relationships between entities in the cases. Therefore, more sub-graphs can be obtained than by using traditional graph mining methods.

Inductive logic programming (ILP) has been used to discover links in relational data [15]. Given background knowledge and a set of positive and negative examples, ILP can infer a hypothesis in the form of a rule. In our work, knowledge is represented in the form of semantic graphs, and reasoning with logical and rule-based inference is used to determine if a query occurs in a particular semantic graph. While the goal of ILP is to define target relation hypotheses, our goal is to mine general relationship associations from a set of semantic graphs.

Liao et al. use case-based reasoning to identify failure mechanisms [14]. They represent failure cases by attribute-value pairs, with weights for each attribute determined by using a genetic algorithm. The case-based reasoning system retrieves the failure mechanisms of archived cases that are calculated to be similar to the target case. Case-based reasoning can handle uncertainties in unstructured domains. However, because attribute-value pairs cannot represent the semantic relationships between entities occurring in a failure case, a case-based reasoning approach based on similarities calculated from attribute values is not suitable for mining relationship associations. Furthermore, case-based reasoning cannot mine general knowledge from large sets of cases. Still, to the extent that our approach supports the retrieval of cases that are similar to a semantic graph that expresses a target case, the semantic matching used in our approach could also be used for case-based reasoning.

6 Conclusions

Failure occurrences are a potential but largely untapped source of knowledge for human society. Mining useful general knowledge from information on specific failure occurrences could help people avoid repeating the same failures.

This paper presented a new technique to mine relationship associations from the Web-based failure knowledge database that has been created by the Japan Science and Technology Agency. The relationship associations that are mined consist of two co-occurring semantic triples, each of which is comprised of a domain instance, a range instance, and a connecting property. Instance classes and properties are defined in an ontology that is formalized in a description logic.

A relationship association mined from the failure knowledge database can be considered as a form of generalized knowledge about failure cases. The association implies that if one relationship occurs in a failure case, then the associated relationship is also likely to occur. In contrast, traditional literature-based discovery methods, such as the Swanson ABC model in medical science, generally mine non-specified relationships between pairs of concepts through keyword co-occurrence or other natural language processing techniques.

In this paper, we adopted Semantic Web techniques that can produce more meaningful results by using inference methods and that use ontology knowledge representation methods to handle relationships between concepts more accurately than natural language processing techniques. We reviewed our previous work to create a corpus of 291 semantic graphs representing information about failure cases, and we described our method for mining relationship associations using ontology and inference. Finally, we presented the results of an experiment using this method to mine relationship associations from the corpus of semantic graphs, and we discussed some of the interesting relationship associations that were mined.

In future work, we will develop additional filters to identify potentially interesting relationship associations. Also, we plan to apply our relationship association mining approach to literature-based discovery of relationships between relationships.

Acknowledgments. We are grateful for advice and information from Professors Y Hatamura, H Kobayashi, M Kunishima, M Nakao, and M Tamura concerning the analysis of the cases in the failure knowledge database. Funding for this research was provided by the Knowledge Failure Database project at the Japan Science and Technology Agency and the Office of the President of the University of Tokyo.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Schneider, P.P.: The Description Logic Handbook: Theory, implementation and applications. CUP (2003)
2. Cohen, W.W., Ravikumar, P., Fienberg, S.E.: A Comparison of String Distance Metrics for Name-Matching Tasks. In: Proc. of the ACM Workshop on Data Cleaning, Record Linkage and Object Identification 2003 (2003)
3. Guo, W., Kraines, S.: Explicit Scientific Knowledge Comparison Based on Semantic Description Matching. In: Proc. of American Society for Information Science and Technology 2008 Annual Meeting (2008)
4. Guo, W., Kraines, S.B.: Mining Common Semantic Patterns from Descriptions of Failure Knowledge. In: Proc. of the 6th International Workshop on Mining and Learning with Graphs (2008)
5. Guo, W., Kraines, S.B.: Discovering Relationship Associations in Life Sciences Using Ontology and Inference. In: Proc. of the 1st International Conference on Knowledge Discovery and Information Retrieval, pp. 10–17 (2009)
6. Guo, W., Kraines, S.B.: Extracting Relationship Associations from Semantic Graphs in Life Sciences. In: Fred, A., et al. (eds.) Knowledge Discovery, Knowledge Engineering and Knowledge Management, IC3K 2009, Revised Selected Papers. CCIS. Springer, Heidelberg (2010)
7. Hatamura, Y., IIno, K., Tsuchiya, K., Hamaguchi, T.: Structure of Failure Knowledge Database and Case Expression. CIRP Annals- Manufacturing Technology 52(1), 97–100 (2003)
8. Inokuchi, A., Washio, T., Motoda, H.: An Apriori-based Algorithm for Mining Frequent Substructures from Graph Data. In: Proc. of the 4th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 13–23 (2000)
9. Inokuchi, A., Washio, T., Nishimura, Y.: A Fast Algorithm for Mining Frequent Connected Subgraphs. IBM Research Report, RT0448 (Feburary 2002)
10. Inokuchi, A.: Mining Generalized Substructures from a Set of Labeled Graphs. In: Proc. of the 4th IEEE International Conference on Data Mining, pp. 415–418 (2004)
11. JST Failure Knowledge Database, `http://shippai.jst.go.jp/en/`
12. Kraines, S., Guo, W.: Using Human Authored Description Logics ABoxes as Concept Models for Natural Language Generation. In: Proc. of American Society for Information Science and Technology 2009 Annual Meeting (2009)
13. Kraines, S., Guo, W., Kemper, B., Nakamura, Y.: EKOSS: A Knowledge-User Centered Approach to Knowledge Sharing, Discovery, and Integration on the Semantic Web. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 833–846. Springer, Heidelberg (2006)
14. Liao, T.W., Zhang, Z.M., Mount, C.R.: A Case-Based Reasoning System for Identifying Failure Mechanisms. Engineering Applications of Artificial Intelligence 13, 199–213 (2000)
15. Mooney, J.R., Melvile, P., Tang, L.R., Shavlik, J., Castro Dutra, I., Page, D., Costa, V.S.: Relational Data Mining with Inductive Logic Programming for Link Discovery. In: Proc. of the National Science Foundation Workshop on Next Generation Data Mining (2002)
16. Nakao, M., Tsuchiya, K., Harita, Y., Iino, K., Kinukawa, H., Kawagoe, S., Koike, Y., Takano, A.: Extracting Failure Knowledge with Associative Search. In: Satoh, K., Inokuchi, A., Nagao, K., Kawamura, T. (eds.) JSAI 2007. LNCS (LNAI), vol. 4914, pp. 269–276. Springer, Heidelberg (2008)
17. OWL Web Ontology Language Guide, `http://www.w3.org/TR/owl-guide/`
18. Rajaraman, K., Tan, A.H.: Mining Semantic Networks for Knowledge Discovery. In: Proc. of the 3rd IEEE International Conference on Data Mining (2003)
19. Tamura, M.: Learn from Failure! Failure Knowledge in Chemical Substances and Plants and Its Use. Chemistry 58(8), 24–29 (2003) (Japanese)

Event Prediction in Network Monitoring Systems: Performing Sequential Pattern Mining in Osmius Monitoring Tool*

Rafael García[1], Luis Llana[1], Constantino Malagón[2], and Jesús Pancorbo[3]

[1] Universidad Complutense de Madrid,
Madrid, Spain
rafaelg.aranda@gmail.com, llana@sip.ucm.es
[2] Universidad Nebrija, Madrid, Spain
cmalagon@nebrija.es
[3] Peopleware, S.L.
Madrid, Spain
jesus.pancorbo@peopleware.es

Abstract. Event prediction is one of the most challenging problems in network monitoring systems. This type of inductive knowledge provides monitoring systems with valuable real time predictive capabilities. By obtaining this knowledge, system and network administrators can anticipate and prevent failures.

In this paper we present a prediction module for the monitoring software Osmius (www.osmius.net). Osmius has been developed by Peopleware (peopleware.es) under GPL licence. We have extended the Osmius database to store the knowledge we obtain from the algorithms in a highly parametrized way. Thus system administrators can apply the most appropriate settings for each system.

Results are presented in terms of positive predictive values and false discovery rates over a huge event database. They confirm that these pattern mining processes will provide network monitoring systems with accurate real time predictive capabilities.

1 Introduction

Nowadays, Information technologies departments are intimately associated with the usual business workflow of every center (including companies, factories, universities, etc.), and it has become a key aspect of the business process itself. Due to the great number of electronic devices, computers and applications connected and the huge volume of data and information generated that has to be saved, assured and managed, business centers have to dedicate great efforts and a lot of their resources to this Data Management process. Thus, the existence of these

* This paper has been supported by Peopleware S.L. and the Project *Osmius 2008*, by the Spanish Ministry of Industry, Tourism and Commerce through the Plan Avanza R&D (TSI-020100-2008-58).

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 632–642, 2010.

IT departments is a response of the necessity of controlling every aspect of this process by the companies.

All of these processes carried out by these IT departments are considered as services that they provide to other departments (e-mail services, printer services, etc.) or to the company clients (company website, e-commerce services, etc.). In order to improve the quality of these services, companies make use of the ITIL (Information Technology Infrastructure Library) framework [11]. It provides good practice guidelines for the development of IT services, from infrastructure and security management processes to the final service deployment. A direct consequence of ITIL is the necessity of a robust and accurate system monitoring tool which reports everything occurring in this infrastructure. A monitoring tool can be extremely helpful for IT departments to detect the presence of system failures. This kind of system can monitor several indicators of those critical systems in a network infrastructure.

The addition of predictive analysis into a network monitoring system allows the discovery of behaviour trends. So it is possible to foresee events before they happen. This fact will provide the IT department the possibility of planning their system capabilities in such a way that the quality of services that they provide will be improved. Thus, event prediction in monitoring systems has become a challenging problem in which the most important monitoring software projects are interested.

In this work, trend analysis has been developed within the framework if the Osmius open source monitoring tool. Osmius provides a framework to easily monitor processes in distributed and multi platform environments. This predictive analysis has been carried out using two known pattern mining techniques. The first analysis used the *frequent pattern mining* technique, by which we have been be able to predict future events based on previous gathered events. A second analysis was performed by using sequential pattern mining techniques; in such a way that not only future events can be predicted, but also its arrangement within a sequence.

The rest of the paper is structured as follows. Next, in Section 2 we will briefly present the main concepts of the Osmius monitoring tool concerning this paper. In Section 3 we will present the algorithms we have used to make the predictive analysis. In order to adapt the data in the Osmius database to be suitable for the previous algorithms, we have had to extend the Osmius data model; this extension is presented in Section 4. Then we will present the experiments (Section 5 we have developed to test the tool and the obtained results (Section 6). Finally, in Section 7 we present some conclusions and future research guidelines.

2 Osmius

Osmius has been recognized as one of the best Open Source monitoring tools. Osmius is capable of monitoring services that have to fulfil service availability requirements also known as Service Level Agreement (or SLA). This SLA is defined as the percentage of time a service must have to be available for a certain period of time. For example, the printer service has to be available 95% of the time at least during the working day, while the e-commerce service has to be available over percentage 99.99% any day at any time. Thus, SLA works as a service quality measure provided by IT departments and allows the possibility of making trend analysis.

Every service in Osmius consists of a set of instances which are being monitored. An instance could be anything connected to the network, from a MySql database to a Unix file server, an Apache web server or a Microsoft Exchange Server. Therefore a service such as a mail service is made up of an instance A, which might be a Unix server, an instance B which might be a Exchange mail server, and another instance C which might be an Apache web server.

Every instance is regularly consulted by Osmius for specific and inherent events, like percentage of CPU, the time in milliseconds taken by an Apache web server to serve a web page, the number of users connected to a mail server, etc. When this instance is monitored by Osmius, it receives an event with three possible values: OK, warning or alarm. The aim of this research is to predict future events based on these gathered events in order to anticipate failures on those instances that are being monitored.

3 Event Prediction Techniques

In this section we are going to describe the algorithms we have considered in order to make event predictions. As it has been said in section 1, these predictions have been carried out by using two different techniques: *frequent pattern mining* and *sequential pattern mining.*

While this kind of predictive analysis has not been widely used in specific real-world applications within monitoring systems industry, both of them have been successfully applied to inter-disciplinary domains beyond data mining. Thus, frequent pattern mining has been applied in many domains such as basket market analysis, indexing and similarity search of complex structured data, spatio temporal and multimedia data mining, mining data streams and web mining [4]. On the other hand, typical applications in real-world domain applications of sequential pattern mining are closer to the aim of this paper as it has been successfully applied to either sensor-based monitoring, such as telecommunications control [13] or log-based monitoring, such as network traffic monitoring [7] or intrusion detection systems [12]. There are also numerous applications in many other fields like bioinformatics (e.g. DNA sequentation) or web mining .

3.1 Frequent Pattern Mining

Frequent pattern mining plays an essential role in many data mining tasks and real world applications, including web mining, bioinformatics or market trends studies. Frequent patterns are defined as patterns whose support value (i.e. the number of times that this pattern appears in a transaction database) is more than a minimum support. By using this minimum support, those patterns which appear the most frequently can be obtained. These are the more useful and interesting patterns in our real application domain.

Thus, in the case of a monitoring system like Osmius, the main objective of mining frequent patterns consists of making associations among gathered events, so we are able to predict future events and therefore identify trends.

It has to be noted that we are interested in non-trivial association rules, i.e., the objective is to discover frequent pattern association that cannot be extracted by only using the domain knowledge. These frequent patterns are defined as patterns that appear frequently together in a transaction data set. For example:

> *Customers who buy the Salingers book "The Catcher in the Rye" are likely to buy an umbrella*

This kind of valuable knowledge is the result of a knowledge discovery process commonly known as *Market basket analysis*.

In our case, we are trying to associate events (i.e., criticity and availability see Section 2) for a set of monitoring instances that belongs to a certain service. Thus, we can describe our main goal with this possible extracted rule:

> *If criticity(Instance_sqlServ,Service_SIP) and*
> *criticity(Instance_YOUTUBE,service_SER1) then*
> *criticity(Instance_Apache1,service1)*

Many algorithms have been developed to mining frequent patterns, from classic Agrawal's Apriori algorithm [1], the one which has been used in this work, to FP-Growth algorithm proposed by Han [5], in which frequent pattern are obtained without candidate generation. Almost all of them represent the data in a different format, determining the heuristic used in the searching process.

Apriori extracts frequent patterns (items in its nomenclature) by using a breadth-first searching process. It previously generates candidate pattern sets (also named itemsets) of length k from itemsets of length $k-1$ by an iterative process. It is based on a property of the itemsets stating that a candidate set of length k must also contains all frequent $k-1$ itemsets.

In this work we have used DMTL [6] (Data Mining Template Library) software, a frequent pattern mining library developed in C++ language to extract frequent patterns from massive datasets [16]. This choice was due to the requirement of integrating this prediction module into the Osmius infrastructure. Thus, event association rules are mined from Osmius data sets by using this DMTL implementation of the classic Agrawal's Apriori algorithm.

3.2 Sequential Pattern Mining

Sequential pattern mining can be defined as the process of extracting frequently ordered events (i.e. sequences whose support exceed a predefined minimal support threshold) or subsequences [3]. A sequential pattern, or simply a sequence, can be then defined as a sequence of events that frequently occurred in a specific order. As in the frequent pattern mining process, it has to be noted that each of these transactions have a time stamp.

Sequences are an important type of data which occur frequently in many real world applications, from DNA sequencing to personalized web navigation [9].

Sequence data have several distinct characteristics, which include:

1. The relative ordering relationship between elements in sequences.
2. Patterns can also be seen as subsequences within a sequence. The only condition is that the order among patterns in a subsequence must be preserved from the corresponding ordering sequence.
3. The time stamp is an important attribute within the process of data mining. This time stamp is then taken into account not only to order the events into a sequence but to get a time prediction in which a future event is going to occur. Thus, if we take time stamp into account then we can get more accurate and useful predicted knowledge such as: *Event A* implies *Event B* within a week.

As it has been said in 3.1, we have used the implementation of Zaki's Spade algorithm [14] in DMTL software, also developed by M. Zaki. SPADE algorithm (*Sequential Pattern Discovery using Equivalent classes*) uses a candidate generate-and-test approach with a vertical data format, instead of the classic horizontal data format used in classic GSP algorithm [10]. Thus, instead of representing data as ($sequence\_ID$: $sequence\ of\ items$), SPADE transforms this representation to a vertical data format (where the data is represented as ($itemset : sequence\_ID,\ event\_ID$).

This vertical data format allows SPADE to outperform GSP by a factor of three [15] as all the sequences are discovered with only three passes over the database.

4 Data Model

In this Section we will summarize the data model we have used to develop the prediction module, as depicted in Figure 1. The data in the historical database of Osmius is not in the format needed for the techniques we have considered. Thus, events are stored in the database indicating when they have happened and how long they have been in that state:

```
osmius@localhost> select * from OSM_HISTINST_AVAILABILITIES limit 6;
+--------------+---------------------+---------------------+------------------+
| IDN_INSTANCE | DTI_INIAVAILABILITY | DTI_FINAVAILABILITY | IND_AVAILABILITY |
+--------------+---------------------+---------------------+------------------+
| cr0101h      | 2009-04-14 00:12:32 | 2009-04-14 00:13:02 |                0 |
| cr0101h      | 2009-04-14 00:13:02 | 2009-04-14 00:22:32 |                1 |
| cr0101h      | 2009-04-14 00:22:32 | 2009-04-14 00:23:02 |                0 |
| cr0101h      | 2009-04-14 00:23:02 | 2009-04-14 02:34:02 |                1 |
| OSMap        | 2009-04-14 00:34:55 | 2009-04-14 00:39:52 |                0 |
| OSMap        | 2009-04-14 00:39:52 | 2009-04-14 01:00:01 |                1 |
+--------------+---------------------+---------------------+------------------+
6 rows in set (0.00 sec)
```

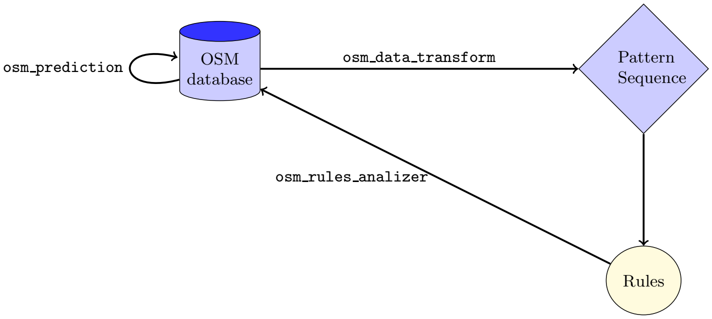


Fig. 1. Data model for Osmius prediction module

The algorithms we have considered group the events in transactions, so we first group the historical Osmius database events in transactions of the desired length. In this paper we have considered that a transaction consists of the events that occur within 1 hour (this length can be easily changed). So we have developed a data model to classify the events appearing in the historical Osmius database. We have called the program that perform this classification `osm_data_transform`. The output of this program has the format required by the DTML software. The output corresponding to the sequence pattern mining technique has the following form:

```
1577 1 3 av0201h--1 cr0201h--1 cr0201h--2
1577 2 2 cr0301h--1 cr0301h--2
1577 3 2 av0301h--1 UBUNTU--1
1577 4 4 cr0101h--1 cr0101h--2 ELMUNDO--1 ELMUNDO--2
1578 1 1 av0201h--1
1578 3 1 av0101h--1
```

Each transaction has been divided into 4 time intervals in such a way that each line represents what has happened in each of those time intervals. Therefore, the first line means that in the first period of transaction 1577, the events `av0201h--1`, `cr0201h--1`, and `cr0201h--2` have occurred. More in general, the first number in each line corresponds to the transaction identifier, the second number indicates the order within the transaction and the third number is the number of events in each line, then all the events in the corresponding time interval appear.

Next we apply the corresponding learning algorithm. Its output must be analyzed and added into the Osmius database. This action is carried out by a program called `osm_rules_analyzer`. The output corresponding to the sequent pattern mining technique will be a file whose lines contain the rules to be analyzed:

```
av0301h--1   UBUNTU--1 -- Support: 12
av0301h--1   av0101h--1 -- Support: 12
av0301h--1   fed_APA--1 -- Support: 11
fed_APA--1   UBUNTU--1 -- Support: 10
fed_APA--1   av0101h--1 -- Support: 11
cr0201h--1   cr0301h--1   cr0101h--1 -- Support: 13
cr0201h--1   cr0301h--2   cr0101h--1 -- Support: 13
cr0201h--1   cr0301h--1   cr0101h--2 -- Support: 15
```

As it can be seen, each line contains a rule and its support. In the previous example the first line contains a rule indicating that after the event `av0301h--1`, the event `UBUNTU--1` has happened 12 times. The file corresponding to the frequent pattern mining technique is similar. Therefore, a *learning* is a set of rules obtained in this way.

Let us remark that different learnings can be applied taking into account different parameters: period of learning, considering only some Osmius services, different periods of the day (morning, afternoon, night), etc. The data model has been designed to store different learnings in order the be able to select the most appropriate one to each circumstance.

Finally there is a program called `osm_prediction` that carry out the actual prediction. It takes as an argument the learning we want to apply and then it takes the *current events* to make a prediction. We want to consider as *current events* those events that have occurred in the current transition. Let us recall that we have considered that a transition consists of the events within 1 hour (this time can be easily changed). So we consider *current events* as the events that have occurred in the last hour.

Lastly, we want to remark that the predictions are also stored in the database. Thus, it is possible to check the accuracy of the predictions by comparing the prediction with the actual events that have actually occurred. Then if the accuracy of the predictions is high we can mark the corresponding learning as valid, we mark it as invalid otherwise.

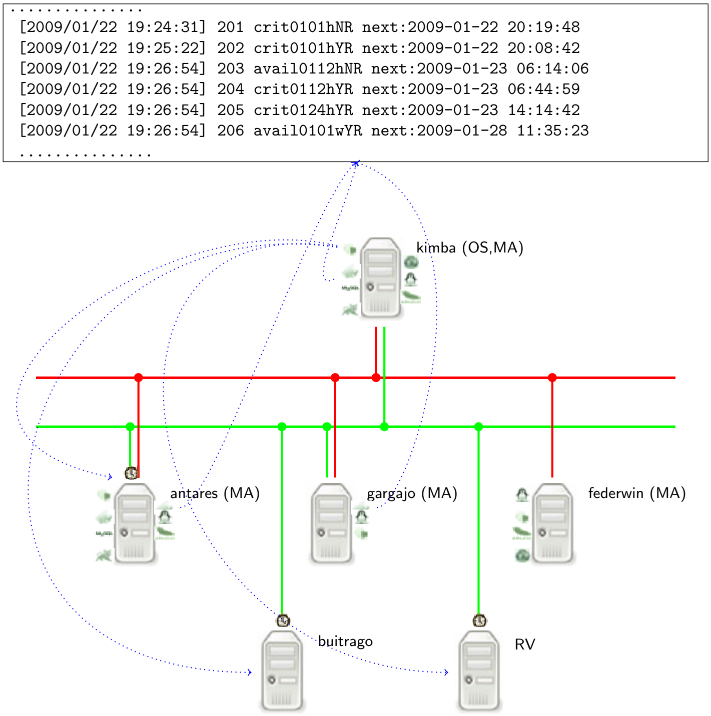
5 Experiments

In order to validate our developed tool, we have prepared a test environment, as depicted in Figure 2. We have installed an Osmius Central Server in a machine called *kimba*. In this machine we have deployed the usual Osmius agent instances plus other instances that we will describe later. We have also deployed three master agents in other machines connected to the same local network: *antares*, *gargajo*, and *federwin*. In addition to this local network, *kimba* is the server of an OpenVPN (`http://openvpn.net`) network that will be used to monitor remote machines located in the Internet with dinamyc IP numbers: *RV*, *buitrago*, and *antares* (in spite of the fact that antares is in the local network, it is also connected to the OpenVPN network); we will take advantage of the OpenVPN network to simulate errors in the own network.

5.1 Osmius Intances

Apart from the usual Osmius instances in kimba we have added some other instances to generate events to provide a more realistic checking environment, and some instances in order to have controlled and correlated errors that will be used to check the prediction tool:

http instances. In order to provide a realistic environment which has a relatively high number of events, the easiest way has been to monitor several typical



```
...............
 [2009/01/22 19:24:31] 201 crit0101hNR next:2009-01-22 20:19:48
 [2009/01/22 19:25:22] 202 crit0101hYR next:2009-01-22 20:08:42
 [2009/01/22 19:26:54] 203 avail0112hNR next:2009-01-23 06:14:06
 [2009/01/22 19:26:54] 204 crit0112hYR next:2009-01-23 06:44:59
 [2009/01/22 19:26:54] 205 crit0124hYR next:2009-01-23 14:14:42
 [2009/01/22 19:26:54] 206 avail0101wYR next:2009-01-28 11:35:23
 ...............
```

Fig. 2. Osmius laboratory layout

web pages: the Universidad Complutense de Madrid (`http://www.ucm.es`), the El Pais newspaper (`http://www.elpais.com`), the El Mundo newspaper (`http://www.elmundo.es`).

IP instances. One easy way to have controlled errors is by using the OpenVPN network. In an ordinary network it is difficult to have automatic controlled errors. We have done this by using the OpenVPN network. We have deployed three IP instances in the kimba master agent to monitor the OpenVPN address of the *RV*, *antares* and *buitrago* computers. By using the the linux cron task, the OpenVPN program in these computers is killed at correlated times.

LOG instances. Since the OpenVPN network cannot be switched off as often we desire, we have decided to deployed more instances in order to have controlled errors. We have used LOG instances because they are easily controllable. We want to have correlated errors in the intervals of 1 hour, so we have defined 3 log agent instances. These instances react when certain strings appears in the a file. The strings and the file are established in the configuration of the agent. Finally we have programmed a daemon that generates the corresponding strings in the appropriate order in correlated times.

6 Results

Results for frequent pattern mining in terms of percentage of events predicted are shown in table 1. This process of event prediction can be described as follows: Once the actual system state is fixed, the event prediction is made by comparing this system state with the knowledge base of event association rules. This actual state means the concrete time at which the prediction is made plus one hour before, so the events predicted are expected to occur during the hour after this point. That is, the prediction is made considering a one hour time window in such a way that events predicted will be gathered within this posterior hour.

As it can be seen in table 1, our module has predicted an average of 72% of events regarding the total of events that have been gathered. This percentage is also known as precision (usually dubbed PPV or positive predictive value), that is, the fraction of events predicted by the system that have really taken place. On the other hand, the false discovery rate (or FDR) is about 28%, that is, the percentage of events predicted by the system that didn't occur within this time window.

Results for sequential pattern mining in terms of True Positive Rate and False Positive Rate for a one-hour time window are shown in table 2. As it can be seen, TPR obtained for sequential pattern mining is about 65%, while the performance in terms of false positives is about 35%. These results are slightly worse than those for frequent pattern mining, and this difference can be explained by the fact that in order to predict frequent sequences, the arrangement of events within a time window has to be considered, resulting in a more difficult prediction process. However, these results are very promising and this type of prediction provides valuable information for prediction in monitoring systems regarding the order in which future events will probably occur.

In Section 6, Results, it will be useful if you can comment on the acceptability of the predictive accuracy of the sequential mining approach for event predictions, and whether more robust predictive mining approaches may be necessary.

Obviously there will be a certain percentage of events that cannot be predicted, mainly due to the fact that frequent events associated with these errors don't exist in the learning model (i.e. the event database) This occurs both in frequent and sequence pattern mining as they are based in similar learning models. In order to decrease this non predicted event rate it is necessary to carry

Table 1. Results for frequent pattern mining for a one-hour time window

One-hour time window	Precision	False discovery rate
	0.723	0.277

Table 2. Results for frequent pattern mining for a one-hour time window

One-hour time window	Precision	False discovery rate
	0.648	0.352

out an incremental learning ([17], [8], [2]) as adding these new events into Osmius database would expand the learning model and it will make possible the detection of these events through the new knowledge base.

7 Conclusions and Future Work

Event prediction is one of the most challenging problems in monitoring systems. It provides monitoring systems with valuable real time predictive capabilities. This prediction is based on the past events of the monitored system; its history is analyzed by using data mining techniques. In this paper we have carried out the prediction by using frequent and sequential pattern mining analysis. As it has been pointed out previously, these techniques have been successfully applied in many fields, such as telecommunications control, network traffic monitoring, intrusion detection systems, bioinformatics and web mining.

We have developed our work within the framework of the open source monitoring tool Osmius (`www.osmius.net`). Osmius stores the past events of the monitored system in a database that makes its analysis easy. In order to make the frequent and sequential pattern mining analysis, it has been necessary to group Osmius events into transactions. We have considered that the transactions consist of the events that happen within an hour; we have studied another time intervals for associations, but the results have not been satisfactory. Smaller intervals are not useful because there is no time to react to correct the problem. Bigger intervals are not useful because there are too many events in each association and then to many events will be predicted. Anyway, the tool we have generated has been designed to be easily adapted to any time interval.

We have built a laboratory to test our tool. In this laboratory we have installed Osmius to monitor the network of computers of our institution. The problem with this system is the low number of failure events produced. So in order to test the tool, we have introduced in that laboratory programmed and correlated failures. The experimental results are shown in terms of true and positive rates. They have confirmed that these pattern mining analysis can provide monitoring systems like Osmius with accurate real time predictive capabilities. The results of this laboratory can be consulted in `http://kimba.mat.ucm.es/osmius/osmius_prediction.tar.bz2`.

As future work we plan to study another well known technique: neural networks. In Osmius the monitored system has an overall grade that indicates how well the system is performing. With neural networks we plan not only to predict failures in the system, but also the future overall grade of the system.

Acknowledgements

We want to thank J.L Marina, R+D Director at Peopleware for fruitful discussions.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. SIGMOD 22(2), 207–216 (1993)
2. Cheng, H., Yan, X., Han, J.: Incspan: Incremental mining of sequential patterns in large database. In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining (2004)
3. Dong, G., Pei, J.: Sequence Data Mining. Springer, Heidelberg (2007)
4. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Min. Knowl. Disc. 5, 55–86 (2007)
5. Han, J., Pei, J., Yiwein, Y., Runying, M.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery 8, 53–87 (2004)
6. Hasan, M., Chaoji, V., Salem, S., Parimi, N., Zaki, M.: Dmtl: A generic data mining template library. In: Workshop on Library-Centric Software Design (LCSD 2005), with Object-Oriented Programming, Systems, Languages and Applications (OOPSLA 2005) conference, San Diego, California (2005)
7. Kim, S., Park, S., Won, J., Kim, S.-W.: Privacy preserving data mining of sequential patterns for network traffic data. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 201–212. Springer, Heidelberg (2007)
8. Leung, C.K.-S., Khan, Q.I., Li, Z., Hoque, T.: Cantree: a canonical-order tree for incremental frequent-pattern mining. Knowl. Inf. Syst. 11, 287–311 (2007)
9. Olson, D., Delen, D.: Advanced Data Mining Techniques. Springer, Heidelberg (2008)
10. Srikant, R., Vu, Q., Agrawal, R.: Mining association rules with item constraints. In: Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining, Newport Beach, CA, pp. 67–73 (1997)
11. Van Bon, J.: The guide to IT service management. Addison-Wesley, Reading (2002)
12. Wu, L., Hunga, C., Chen, S.: Building intrusion pattern miner for snort network intrusion detection system. Journal of Systems and Software 80, 1699–1715 (2007)
13. Wu, P., Peng, W., Chen, M.: Mining sequential alarm patterns in a telecommunication database. In: Jonker, W. (ed.) VLDB-WS 2001 and DBTel 2001. LNCS, vol. 2209, p. 37. Springer, Heidelberg (2001)
14. Zaki, M.: Scalable algorithms for association minning. IEEE Trans. Knowledge and Data Engineering 12, 372–390 (2000)
15. Zaki, M.: Spade: An efficient algorithm for mining frequent sequences. Machine Learning 42(1-2), 31–60 (2001)
16. Zaki, M.: DMTL (December 2007), http://sourceforge.net/projects/dmtl
17. Zequn, Z., Eseife, C.I.: A low-scan incremental association rule maintenance method based on the apriori property. In: Stroulia, E., Matwin, S. (eds.) Canadian AI 2001. LNCS (LNAI), vol. 2056, pp. 26–35. Springer, Heidelberg (2001)

Selection of Effective Network Parameters in Attacks for Intrusion Detection

Gholam Reza Zargar[1] and Peyman Kabiri[2]

[1] Khouzestan Electric Power Distribution Company, Ahwaz, Iran
[2] Iran University of Science and Technology / Intelligent Automation Laboratory, School of Computer Engineering, Tehran, Iran
Zargar@vu.iust.ac.ir, Peyman.Kabiri@iust.ac.ir

Abstract. Current Intrusion Detection Systems (IDS) examine a large number of data features to detect intrusion or misuse patterns. Some of the features may be redundant or with a little contribution to the detection process. The purpose of this study is to identify important input features in building an IDS that are computationally efficient and effective. This paper proposes and investigates a selection of effective network parameters for detecting network intrusions that are extracted from Tcpdump DARPA1998 dataset. Here PCA method is used to determine an optimal feature set. An appropriate feature set helps to build efficient decision model as well as to reduce the population of the feature set. Feature reduction will speed up the training and the testing process for the attack identification system considerably. Tcpdump of DARPA1998 intrusion dataset was used in the experiments as the test data. Experimental results indicate a reduction in training and testing time while maintaining the detection accuracy within tolerable range.

Keywords: Intrusion Detection, Principal Components Analysis, Clustering, Data Dimension Reduction, Feature Selection.

1 Introduction

Basically, Intrusion Detection System (IDS) is classified into two categories: signature-based intrusion detection and anomaly-based intrusion detection. Signature-based intrusion detection tries to find attack signatures in the monitored resource. Anomaly-based intrusion detection typically relies on knowledge of normal behavior and identifies any deviation from it.

In practice, the huge amount of data flowing on the internet makes the real-time intrusion detection nearly impossible. Even though computing power is increasing exponentially, internet traffic is still too large for real-time computation. Parameter selection can reduce the needed computation power and model complexity. This makes it easier to understand and analyze the model for the network and to make it more practical to launch real-time intrusion detection system in large networks. Furthermore, the storage requirements of the dataset and the computational power needed to generate indirect features, such as traffic signature and statistics, can be reduced by the feature reduction [1].

P. Perner (Ed.): ICDM 2010, LNAI 6171, pp. 643–652, 2010.

Intrusion detection systems are typically classified as host-based or network-based. Host-based IDS will monitor resources such as system logs, file systems and disk resources; whereas a network-based intrusion detection system monitors the data passing through the network. Different detection techniques can be employed to search for the attack patterns in the monitored data [2].

Intrusion detection systems that are currently in use typically require human input to create attack signatures or to determine effective models for normal behavior. Support for learning algorithms provides a potential alternative to expensive human input.

Problems such as, problem of irrelevant and redundant features are two major problems in the dataset collected from network traffic. These problems not only hinder the detection speed but also decline the detection performance of intrusion detection systems [3]. Details of these problems are described in the following.

In general, the quantity of data processed by the IDS is large. It includes thousands of traffic records with a number of various features such as the length of the connection, type of the protocol, type of the network service and lots other information. Theoretically and ideally, the ability to discriminate attacks from normal behavior should be improved if more features are used for the analysis. However, this assumption is not always true, since not every feature in the traffic data is relevant to the intrusion detection. Among the large amount of features, some of them may be irrelevant or confusing with respect to the target patterns. Some of the features might be redundant due to their high inter-correlation with one or more of the other features in the dataset [4]. To achieve a better overall detection performance, any irrelevant and redundant features should be discarded from the original feature space.

In intrusion detection process, some features may be irrelevant or redundant which complicates the detection process and will increase the detection time. The main goal in feature selection is to reduce the volume of the data that are less important to the detection cause and can be eliminated. This has the benefit of decreasing storage and data transfer requirements, reducing processing time and improving the detection rate. An IDS has to examine a very large audit data [5]. Therefore, it should reduce the volume of data to save the processing time. Feature reduction can be performed in several ways [6, 7 and 8]. This paper proposes a method based on features extracted from TCP/IP header parameters. In the proposed approach, Principle Component Analysis (PCA) method is used as a dimension reduction technique.

2 Related Works

Srinivas Mukkamala and Andrew H. Sung [9] use performance-based method (PFRM) for to identify the important features of the TCP/IP data, making the following conclusions: only the important features are used, the result achieves a higher accuracy than when all the features are used. However, PFRM neglects the relationship between the features. Generally, the capability of anomaly-based IDS is often hindered by its inability to accurately classify variation of normal behavior as an intrusion. Additionally, Mukkamala and et al say that “network traffic data is huge and it causes a prohibitively high overhead and often becomes a major problem for IDS” [10]. Chakraborty [11] reports that existence of these irrelevant and redundant features generally affects the performance of machine learning or pattern classification algorithms. Hassan et al [12] proved that proper selection of feature set has resulted in better classification performance. Sung and Mukkamala [7] have used SVM [13] and Neural Network to identify and categorized features with respect to their importance in regard to detection of specific kinds of attacks such as probe, DoS, Remote to Local (R2L), and User to Root (U2R). They have also demonstrated that the elimination of these unimportant and irrelevant features did not significantly reduce the performance of the IDS. Chebrolu et al [6], tackled the issue of effectiveness of an IDS in terms of real-time operation and detection accuracy from the feature reduction perspective.

3 Denial of Service (DoS) Attacks

A denial of service attack is member of a class of attacks in which the attacker consumes computing or memory resources in such a way that the targeted system will be unable to handle legitimate requests, or denies legitimate user access to a machine. Apache2, Back, Land, Mail bomb, SYN Flood, Ping of death, Process table, Smurf, Syslogd, Teardrop, Udpstorm and Neptune attacks are some examples of the Dos attack.

4 Syn Flood Attack

TCP needs to establish a connection between a source host and a destination host before any data can be transmitted between them. The connection process is called the three-way handshake. In The first step a SYN packet is sent from Source to Destination node. Then destination node sends a message to source with its SYN and Ack flags set. In the third step source sends a massage with its ACK flag set to the destination node. Here a connection is established between source and destination nodes. The third message may contain user payload data.

Syn flood is a DoS attack in which every TCP/IP implementation is vulnerable to some degree. Each half-open TCP connection made to a machine causes the 'tcpd' server to add a record to the data structure that stores information describing all pending connections. [14].

Christopher [15] believes that "typical Synflood attacks can vary several parameters: the number of SYN packets per source address sent in a batch, the delay between successive batches, and the mode of source address allocation".

5 Data Reduction and Feature Selection Using PCA

One of the mathematical methods for transforming a number of possibly correlated variables into a smaller number of uncorrelated variables is Principal Component Analysis (PCA). In this method, the first principal component stands for the highest variability in the data, and each succeeding component stands for the less variability in the data [16].

This transformation is carried out by finding those orthogonal linear combinations of the original variables with the largest variance. In many datasets, the first several principal components have the highest contribution to the variance in the original dataset. Therefore, the rest can be ignored with minimal loss of the information value during the dimension reduction process [5][17]. The transformation works as follows:

Given a set of observations x_1, x_2, …, x_n where each observation is represented by a vector of length m, the dataset is thus represented by a matrix $X_{n\times m}$.

$$X_{n\times m}=\begin{bmatrix} x_{11} & x_{12} & . & . & . & x_{1m} \\ x_{21} & x_{22} & . & . & . & x_{2m} \\ . & . & . & . & . & . \\ x_{n1} & x_{n2} & . & . & . & x_{nm} \end{bmatrix}=[x_1,x_2,\ldots x_n] \tag{1}$$

ï

ï

The average for each observation is defined by the equation (2).

$$\mu=\frac{1}{n}\sum_{i=1}^{n}x_i \tag{2}$$

The deviation from the mean is defined in equation (3).

$$\phi_i=x_i-\mu \tag{3}$$

The sample covariance matrix of the dataset is defined in equation (4).

$$C=\frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)^T=\frac{1}{n}\sum_{i=1}^{n}\phi_i\phi_i^T=\frac{1}{n}AA^T \tag{4}$$

Where $A=[\phi_1,\phi_2,\ldots,\phi_n]$

Applying PCA for the data dimension reduction, eigenvalues and corresponding eigenvectors of the sample covariance matrix C have to be calculated [18]. Let $(\lambda_1,u_1),(\lambda_2,u_2),\ldots.(\lambda_m,u_m)$ present m eigenvalue-eigenvector pairs of the sampled covariance matrix C. The k eigenvectors associated with the largest eigenvalues are selected. The dimensionality of the subspace k can be determined by the following equation [19]:

The resulted $m\times k$ matrix U, with k eigenvectors as its columns is called eigenvectors matrix or coefficient matrix. Data transformation using principal components into the k-dimensional subspace is carried-out using equation (5) [5].

$$y_i=U^T(x_i-\mu)=U^T\phi_i \tag{5}$$

6 K-Nearest-Neighbor Classifiers

Nearest-neighbor classifiers are based on learning by analogy, that is, comparing a given test tuple against training tuples that are similar to it. The training tuple is described by n attributes. Each tuple represents a point in an n-dimensional space. In this way, all of the training tuples are stored in an n-dimensional space. Given an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k "nearest neighbors" of the unknown tuple. "Closeness" is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two points or tuples, say, ï $X_2 = (x_{21}, x_{22}, ..., x_{2n})$ and $X_1 = (x_{11}, x_{12}, ..., x_{1n})$ is

$$Dist\,(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2} \qquad (6)$$

The square root is subtracted from the total sum of the distances. Typically, before evaluating the Equation (6), values of each attribute should be normalized [20].

7 The Dataset and Pre-processing

In the following subsections the dataset used in the reported work and the pre-processing applied on the dataset are presented.

7.1 The Dataset Used in This Work

In this work, basic features are extracted from TCP/IP dump data [21]. These features can be derived from packet headers without inspecting the payload. In the reported work, TCP dump from the DARPA'98 dataset is used as the input dataset. Extracting the basic features, packet information in the TCP dump file is summarized into connections. Specifically, a connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows from a source IP address to a target IP address under some well defined protocol [22].

DARPA'98 dataset provides around 4 gigabytes of compressed TCP dump data [23] for 7 weeks of network traffic [24]. This dataset can be processed into about 5 millions connection records each about 100 bytes in size. The resulted dataset contains the payload of the packets transmitted between hosts inside and outside a simulated military base. BSM audit data from one UNIX Solaris host for some network sessions were also provided. DARPA 1998 TCP dump Dataset [25] was preprocessed and labeled with two class labels, e.g. normal and attack. The dataset contains different types of attacks. Smurf and Neptune from DoS, Eject, Ffb, Perlmagic, Eject-fail, Loadmodule from U2R and Portsweep from Probing attack category are extracted. Denial of Service (DoS) is a class of attacks where an attacker makes some computing or memory resource too busy or too full to handle legitimate requests, thus denying legitimate users access to a machine. User to root (U2R) exploits are a class of attacks where an attacker starts out with access to a normal user account on the system and is able to exploit vulnerability to gain root access to the system. Probing is a class of attacks in which an attacker scans a network of computers to collect information or find known vulnerabilities. This information can be used by an intruder with a map of machines and services that are available on a network to look for exploits.

7.2 Pre-processing

Features extracted from TCP, IP, UDP and ICMP protocols are presented in Table 1. There are 32 basic features extracted using Wireshark and Editcap softwares. Wireshark and Editcap software are used to analyze and minimize TCP dump files [26][23].

Table 1. Basic features extracted from the TCP/IP header

No.	Feature	Description	No.	Feature	Description
1	Protocol	Type of Protocol	17	Src_port	Source Port
2	Frame_lenght	Length of Frame	18	Dst_port	Destination port
3	Capture_lenght	Length of Capture	19	Stream_index	Stream Index number
4	Frame_IS_marked	Frame IS Marked	20	Sequence_number	Sequence number
5	Coloring_rule_name	Coloring Rule name	21	Ack_number	Acknowledgment number
6	Ethernet_type	Type of Ethernet Protocol	22	Cwr_flag	Cwr Flag(status flag of the connection)
7	Ver_IP	IP Version	23	Ecn_echo_flag	Ecn Echo flag (status flag of the connection)
8	Header_lenght_IP	IP Header length	24	Urgent_flag	Urgent flag(status flag of the connection)
9	Differentiated_S	Differentiated Service	25	Ack_flag	Acknowledgment flag(status flag of the connection)
10	IP_Total_Lenght	IP total length	26	Psh_flag	push flag(status flag of the connection)
11	Identification_IP	Identification IP	27	Rst_flag	Reset flag(status flag of the connection)
12	MF_Flag_IP	More Fragment flag	28	Syn_flag	Syn flag (status flag of the connection)
13	DF_Flag_IP	Don't Fragment flag	29	Fin_flag	Finish flag(status flag of the connection)
14	Fragmentation_offset_IP	Fragmentation offset IP	30	ICMP_Type	specifies the format of the ICMP message such as: (8=echo request and 0=echo reply)
15	Time_to_live_IP	Time to live IP	31	ICMP_code	Further qualifies the ICMP message
16	Protocol_no	Protocol number	32	ICMP_data	ICMP data

In this work, intention is to reduce the processing and data transfer time needed for the intrusion detection. To do so, an accurate feature selection scheme is proposed to select important features with minimum loss of information. Paper also aims to select features in such a way that their discrimination set to be categorical. This means that the selection criteria will be the same or with a low variance for the attacks in the same category. This property will increase the adaptability of the IDS that is using this feature set to the variation of the attack patterns that fall in the same category.

8 Experiments and Results

Table 2 shows number of records that are extracted from the TCP/IP dump dataset. As shown in Table 3, a large part of the records from Table 2 are selected for the experiments. In the experiments, 9459 normal records are selected randomly to mix with attacks records.

Table 2. Number of records in different attacks

Category	Class name	Number of Records
Normal	Normal	88860
DOS	Smurf	178855
	Neptune	304871
U2R	Eject	231
	Ffb	103
	Perlmagic	54
	Eject-fail	92
	Loadmodule	33
Prob	Portsweep	158
Sum		573345

Table 3. Number of record in different attacks for calculate

Category	Class name	Number of Records
Normal	Normal	9459
DOS	Smurf	9611
	Neptune	9829
U2R	Eject	231
	Ffb	103
	Perlmagic	54
	Eject-fail	92
	Loadmodule	33
prob	Portsweep	158
Sum		29570

As depicted in figure 1, attacks and normal graphs are presented in the same diagram. In this figure the graph for the normal behavior is depicted in red color. After the experiment, using PCA algorithm effective parameters for detecting normal and attack states are extracted. Table 4 shows relevant features with accumulated percentage of information for each attack. Table 4, shows effective parameters for different attacks. These feature sets are different for different attack categories. They can be used to improve intrusion detection and reduce detection time without seriously affecting the detection accuracy. For example, in smurf attack, parameter number 11 has 100% of information value. This means that if a threshold is determined for this parameter, IDS will be able to detect smurf attack. If intrusion detection uses this selected parameter for detection, detection time will be reduced. KNN classification was used in this experiment, initially all 32 parameters presented in Table 1 are used. Later on, in a second experiment, KNN classification was performed using effective parameters reported in Table 4. Table 5 shows detection time needed for the smurf attack in the two experiments. Attack was detected in, 32.46 and 25.87 seconds during the first and the second experiments respectively.

In regard to Neptune attack scenario in section 3, this attack only uses flag features. Comparing results of this experiment (Table 4, row 2) against Neptune attack scenario, one can conclude that, if a threshold is defined for effective parameters 26, 25

Table 4. List of feature for which the class is selected most

Class name	Relevant features	Percent
SMURF	11	100
NEPTUNE	26,25,28	99.98
EJECT	25,28,12,13,5,29,26,19	99.92
FFB	26,28	100
PERLMAGIC	25,28,12,13,29,5,26,19,1	99.91
EJECT-FAIL	12,13,25,28,26,5,1	99.98
LOADMODULE	29,19,26,27,10	99.46
PORTSWEEP	26,28,25,20,27,5,19	99.97
NORMAL	27,	98.22
NORMAL	27,25,12,13	99.45

and 28 or Ack_flag, psh_flag and syn_flag, Neptune attack can be identified. Using effective parameters 26, 25 and 28 may reduce detection time without effecting accuracy detection significantly. In Table 4, effective parameters for the detection of the normal state are presented. There are parameters 27, 25, 12 and 13 with 99.45% information value, among them parameter 27 or Ack_flag has the highest information value. First three parameters that are present in all the attacks have the highest information value. Parameter number 27 isn't in any of them, so if a threshold is defined for parameter number 27, intrusion detection will be able to distinguish normal record from the attack record with a reduced detection time and without a great change in the detection accuracy. Figure 2 shows the Scree graph for both normal and attack states.

Additionally, for evaluating the performance of the proposed approach, results from the KNN classification algorithm was selected and values for the true positive and false alarm rates were calculated. As reported in Table 5, after dimension reduction using PCA, calculation time is less than before. In this experiment true positive and false alarm rates with PCA and without it are calculated. Different effective parameters extracted for each attack scenario using PCA method are used in classification.

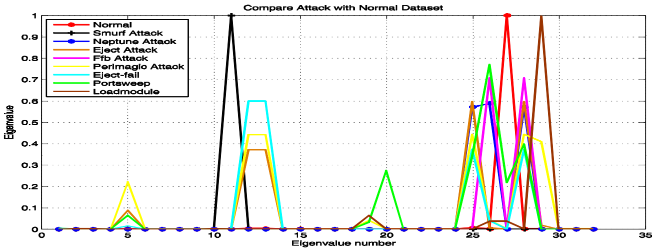


Fig. 1. Compare Effective features in normal and different attack states

Table 5. Comparison between execution time and detection accuracy for KNN classification before and after dimension reduction by PCA

Attack name		TP	FA	Time (second)
Smurf	All 32 parameter	89.97	0.03	32.46
	only parameter number 11	78	11.99	25.87
eject-fail	all 32 parameter	89.99	0	7.76
	only parameter number 12,13,25,28,26,5,1	89.31	0.68	7.46
Loadmodule	all 32 parameter	89.92	0.06	7.66
	only parameter number 29,19,26,27,10	89.75	0.23	6.73
Portsweep	all 32 parameter	89.53	0.45	8.13
	only parameter number 26,28,25,20,27,5,19	89.54	0.44	6.75

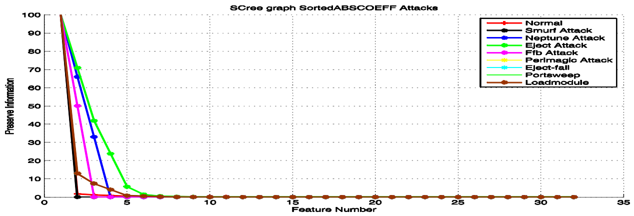


Fig. 2. A comparison between Scree graph for normal and attacks

9 Conclusion

This paper presents a method based on the TCP/IP basic features that uses PCA for dimension reduction and feature analysis for intrusion detection. Therefore, considering the reported results, it can be concluded that using PCA for dimension reduction can reduce calculation time in intrusion detection, and keep detection accuracy intact. Table 4, shows relevant features obtained for different attack categories and effectiveness of these feature sets. These effective features for any attack and normal traffic are distinct. It can be concluded that using these effective features, all the selected attacks are detectable with calculation shorter detection time while the detection accuracy is not increased significantly.

10 Future Work

Plan for the future work is to calculate PCA for other attack categories and find feature sets for different attack categories. Later on, intention is to use classification methods to detect intrusion. Intension is to extend the work to prove its reliability with regard to changes in the attack pattern.

References

[1] Ng, W.W.Y., Rocky, K.C., Chang, Daniel, Yeung, S.: Dimensionality Reduction for Denial of Service Detection Problems Using RBFNN Output Sensitivity. In: Proceedings of the Second International Conference on Machine Learning and Cybernetics, Wan, November 2-5 (2003)

[2] Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. J. Machine Learning Research 3, 1157–1182 (2003)

[3] Chou, T.S., Yen, K.K., Luo, J.: Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms. J. Computational Intelligence 4(3), 196–208 (2008)

[4] Sabahi, F., Movaghar, A.: Intrusion Detection: A Survey. In: 3rd international conference on system and network communication, ICSNC 2008, pp. 23–26 (2008)

[5] Zargar, G., Kabiri, P.: Identification of Effective Network Feature for Probing Attack Detection. In: First International Conference on Network Digital Technologies (NDT 2009), pp. 392–397 (2009)

[6] Chebrolu, S., Abraham, A., Thomas, J.: Feature Deduction and Ensemble Design of Intrusion Detection Systems. J. Computers and Security 24(4), 295–307 (2005)

[7] Sung, A.H., Mukkamala, S.: Identifying important features for intrusion detection using support vector machines and neural networks. In: International Symposium on Applications and the Internet (SAINT), pp. 209–216 (2003)

[8] Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High dimensional Data for Data Mining applications. In: ACMSIGMOD International Conference on Management of Data, Seattle, WA, pp. 94–105 (1998)

[9] Sung, A.H., Mukkamala, S.: Identifying Important Features for Intrusion Detection Using Support Vector Machines and Neural Networks. In: SAINT, pp. 209–217 (2003)

[10] Sung, A.H., Mukkamala, S.: The Feature Selection and Intrusion Detection Problems. In: Maher, M.J. (ed.) ASIAN 2004. LNCS, vol. 3321, pp. 468–482. Springer, Heidelberg (2004)

[11] Chakraborty, B.: Feature Subset Selection by Neurorough Hybridization. LNCS, pp. 519–526. Springer, Heidelberg (2005)

[12] Hassan, A., Nabi Baksh, M.S., Shaharoun, A.M., Jamaluddin, H.: Improved SPC Chart Pattern Recognition Using Statistical Feature. J. of Production Research 41(7), 1587–1603 (2003)

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

[14] Hassanzadeh, A., Sadeghian, B.: Intrusion Detection with Data Correlation Relation Graph. In: Third International Conference on Availability, Reliability and Security, ARES 2008, pp. 982–989 (2008)

[15] Christopher, L., Schuba, V., Ivan, Krsul, et al.: Analysis of a denial of service attack on TCP. In: The IEEE Symposium on Security and Privacy, p. 208 (1997)

[16] A Tutorial on Principal Component Analysis: Derivation, Discussion and Singular Value Decomposition, `http://www.dgp.toronto.edu/~aranjan/tuts/pca.pdf`

[17] Wang, W., Battiti, R.: Identifying Intrusions in Computer Networks based on Principal Component Analysis (2009), `http://eprints.biblio.unitn.it/archive/00000917/` (as visited on January 20, 2009)

[18] Golub, G.H., Van Loan, C.F.: Matrix Computation. Johns Hopkins Univ. Press, Baltimore (1996)

[19] Jolliffe, I.T.: Principal Component Analysis, 2nd edn. Springer, NY (2002)

[20] Han, J., Kamber, M.: Data Mining: Concepts and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2006)

[21] `http://www.wireshark.org` (as visited on January 29, 2009)

[22] Knowledge discovery in databases DARPA archive. Task Description, `http://www.kdd.ics.uci.edu/databases/kddcup99/task.html` (as visited on January 15, 2009)

[23] `http://www.Tcpdump.org` (as visited on January 28, 2009)

[24] MIT Lincoln Laboratory, `http://www.ll.mit.edu/IST/ideval/` (as visited on January 27, 2009)

[25] Lee, W.: A Data Mining Framework for Constructing Feature and Model for Intrusion Detection System. PhD thesis University of Columbia (1999)

[26] `http://www.wireshark.org/docs/man-ages/editcap.html` (as visited on January 20, 2009)

Author Index