

ECML 2007 PRDD
WARSAW POLAND

THE 18TH EUROPEAN CONFERENCE ON MACHINE LEARNING
AND
THE 11TH EUROPEAN CONFERENCE ON PRINCIPLES AND PRACTICE
OF KNOWLEDGE DISCOVERY IN DATABASES

PROCEEDINGS OF THE
THIRD INTERNATIONAL
WORKSHOP ON
MINING COMPLEX DATA

MCD 2007

September 17 and 21, 2007

Warsaw, Poland

Editors:

Zbigniew W. Raś

University of North Carolina at Charlotte, USA

Djamel Zighed

Universite Lyon II, France

Shusaku Tsumoto

Shimane Medical University, Japan

Preface

Data mining and knowledge discovery, as stated in their early definition, can today be considered as stable fields with numerous efficient methods and studies that have been proposed to extract knowledge from data. Nevertheless, the famous golden nugget is still challenging. Actually, the context evolved since the first definition of the *KDD* process has been given and knowledge has now to be extracted from data getting more and more complex.

In the framework of Data Mining, many software solutions were developed for the extraction of knowledge from tabular data (which are typically obtained from relational databases). Methodological extensions were proposed to deal with data initially obtained from other sources, like in the context of natural language (text mining) and image (image mining). *KDD* has thus evolved following a unimodal scheme instantiated according to the type of the underlying data (tabular data, text, images, etc), which, at the end, always leads to working on the classical double entry tabular format.

However, in a large number of application domains, this unimodal approach appears to be too restrictive. Consider for instance a corpus of medical files. Each file can contain tabular data such as results of biological analyzes, textual data coming from clinical reports, image data such as radiographies, echograms, or electrocardiograms. In a decision making framework, treating each type of information separately has serious drawbacks. It appears therefore more and more necessary to consider these different data simultaneously, thereby encompassing all their complexity.

Hence, a natural question arises: how could one combine information of different nature and associate them with a same semantic unit, which is for instance the patient? On a methodological level, one could also wonder how to compare such complex units via similarity measures. The classical approach consists in aggregating partial dissimilarities computed on components of the same type. However, this approach tends to make superposed layers of information. It considers that the whole entity is the sum of its components. By analogy with the analysis of complex systems, it appears that knowledge discovery in complex data can not simply consist of the concatenation of the partial information obtained from each part of the object. The aim would rather be to discover more global knowledge giving a meaning to the components and associating them with the semantic unit. This fundamental information cannot be extracted by the currently considered approaches and the available tools.

The new data mining strategies shall take into account the specificities of complex objects (units with which are associated the complex data). These specificities are summarized hereafter:

Different kind. The data associated to an object are of different types. Besides classical numerical, categorical or symbolic descriptors, text, image or audio/video data are often available.

Diversity of the sources. The data come from different sources. As shown in the context of medical files, the collected data can come from surveys filled in by doctors, textual reports, measures acquired from medical equipment, radiographies, echograms, etc.

Evolving and distributed. It often happens that the same object is described according to the same characteristics at different times or different places. For instance, a patient may often consult several doctors, each one of them producing specific information. These different data are associated with the same subject.

Linked to expert knowledge. Intelligent data mining should also take into account external information, also called expert knowledge, which could be taken into account by means of ontology. In the framework of oncology for instance, the expert knowledge is organized under the form of decision trees and is made available under the form of “best practice guides” called Standard Option Recommendations (SOR).

Dimensionality of the data. The association of different data sources at different moments multiplies the points of view and therefore the number of potential descriptors. The resulting high dimensionality is the cause of both algorithmic and methodological difficulties.

The difficulty of Knowledge Discovery in complex data lies in all these specificities.

We wish to express our gratitude to all members of the Program Committee and the Organizing Committee. Hakim Hacid (Chair of the Organizing Committee) did a terrific job of putting together and maintaining the home page for the workshop as well as helping us to prepare the workshop proceedings.

Zbigniew W. Raś
Djamel Zighed
Shusaku Tsumoto

MCD 2007 Workshop Committee

Workshop Chairs:

Zbigniew W. Raś (Univ. of North Carolina, Charlotte)
Djamel Zighed (Univ. Lyon II, France)
Shusaku Tsumoto (Shimane Medical Univ., Japan)

Organizing Committee:

Hakim Hacid (Univ. Lyon II, France)(Chair)
Rory Lewis (Univ. of North Carolina, Charlotte)
Xin Zhang (Univ. of North Carolina, Charlotte)

Program Committee:

Aijun An (York Univ., Canada)
Elisa Bertino (Purdue Univ., USA)
Ivan Bratko (Univ. of Ljubljana, Slovenia)
Michelangelo Ceci (Univ. Bari, Italy)
Juan-Carlos Cubero (Univ of Granada, Spain)
Tapio Elomaa (Tampere Univ. of Technology, Finland)
Floriana Esposito (Univ. Bari, Italy)
Mirsad Hadzikadic (UNC-Charlotte, USA)
Howard Hamilton (Univ. Regina, Canada)
Shoji Hirano (Shimane Univ., Japan)
Mieczyslaw Kłopotek (ICS PAS, Poland)
Bożena Kostek (Technical Univ. of Gdansk, Poland)
Nada Lavrac (Jozef Stefan Institute, Slovenia)
Tsau Young Lin (San Jose State Univ., USA)
Jiming Liu (Univ. of Windsor, Canada)
Hiroshi Motoda (AFOSR/AOARD & Osaka Univ., Japan)
James Peters (Univ. of Manitoba, Canada)
Jean-Marc Petit (LIRIS, INSA Lyon, France)
Vijay Raghavan (Univ. of Louisiana, USA)
Jan Rauch (Univ. of Economics, Prague, Czech Republic)
Henryk Rybiński (Warsaw Univ. of Technology, Poland)
Dominik Slezak (Infobright, Canada)
Roman Slowiński (Poznan Univ. of Technology, Poland)
Jurek Stefanowski (Poznan Univ. of Technology, Poland)
Juan Vargas (Microsoft, USA)
Alicja Wierzchowska (PJIIT, Poland)
Xindong Wu (Univ. of Vermont, USA)
Yiyu Yao (Univ. Regina, Canada)
Ning Zhong (Maebashi Inst. of Tech., Japan)

Table of Contents

Using Text Mining and Link Analysis for Software Mining	1
<i>Miha Grcar, Marko Grobelnik, and Dunja Mladenic</i>	
Generalization-based Similarity for Conceptual Clustering	13
<i>S. Ferilli, T.M.A. Basile, N. Di Mauro, M. Biba, and F. Esposito</i>	
Finding Composite Episodes	25
<i>Ronnie Bathoorn and Arno Siebes</i>	
Using Secondary Knowledge to Support Decision Tree Classification of Retro- spective Clinical Data	37
<i>Dympna O’Sullivan, William Elazmeh, Szymon Wilk, Ken Farion, Stan Matwin, Wojtek Michalowski, Morvarid Sehatkar</i>	
Evaluating a Trading Rule Mining Method based on Temporal Pattern Extraction .	49
<i>Hidenao Abe, Satoru Hirabayashi, Miho Ohsaki, Takahira Yamaguchi</i>	
Discriminant Feature Analysis for Music Timbre Recognition	59
<i>Xin Zhang, Zbigniew W. Raś</i>	
Discovery of Frequent Graph Patterns that Consist of the Vertices with the Com- plex Structures	71
<i>Tsubasa Yamamoto, Tomonobu Ozaki, Takenao Okawa</i>	
Learning to Order Basic Components of Structured Complex Objects	83
<i>Donato Malerba, Michelangelo Ceci</i>	
ARoGS: Action Rules Discovery based on Grabbing Strategy and LERS	95
<i>Zbigniew W. Raś, Elżbieta Wyrzykowska</i>	
Discovering Word Meanings Based on Frequent Termsets	106
<i>Henryk Rybinski, Marzena Kryszkiewicz, Grzegorz Protaziuk, Aleksandra Kon- tkiewicz, Katarzyna Marcinkowska, Alexandre Delteil</i>	
Feature Selection: Near Set Approach	116
<i>James F. Peters, Sheela Ramanna</i>	
Contextual Adaptive Clustering with Personalization	128
<i>Krzysztof Ciesielski, Mieczysław A. Kłopotek, Sławomir Wierzchoń</i>	
Unsupervised Grouping of Trajectory Data on Laboratory Examinations for Find- ing Exacerbating Cases in Chronic Diseases	139
<i>Shoji Hirano, Shusaku Tsumoto</i>	

Improving Boosting by Exploiting Former Assumptions	151
<i>Emna Bahri, Nicolas Nicoloyannis, Mondher Maddouri</i>	
Ordinal Classification with Decision Rules	163
<i>Krzysztof Dembczyński, Wojciech Kotłowski, Roman Słowiński</i>	
Quality of Musical Instrument Sound Identification for Various Levels of Accompanying Sounds	175
<i>Alicja Wieczorkowska, Elżbieta Kolczyńska</i>	
Estimating Semantic Distance Between Concepts for Semantic Heterogeneous Information Retrieval	185
<i>Ahmad El Sayed, Hakim Hacid, Djamel Zighed</i>	
Clustering Individuals in Ontologies: a Distance-based Evolutionary Approach . .	197
<i>Nicola Fanizzi, Claudia d'Amato, Floriana Esposito</i>	
Data Mining of Multi-categorized Data	209
<i>Akinori Abe, Norihiro Hagita, Michiko Furutani, Yoshiyuki Furutani, and Rumiko Matsuoka</i>	
POM Centric Multiaspect Data Analysis for Investigating Human Problem Solving Function	221
<i>Shinichi Motomura, Akinori Hara, Ning Zhong, Shengfu Lu</i>	
Author Index	233

Using Text Mining and Link Analysis for Software Mining

Miha Grcar¹, Marko Grobelnik¹, Dunja Mladenic¹

¹ Jozef Stefan Institute, Dept. of Knowledge Technologies, Jamova 39,
1000 Ljubljana, Slovenia
{miha.grcar, marko.grobelnik, dunja.mladenic}@ijs.si

Abstract. Many data mining techniques are these days in use for ontology learning – text mining, Web mining, graph mining, link analysis, relational data mining, and so on. In the current state-of-the-art bundle there is a lack of “software mining” techniques. This term denotes the process of extracting knowledge out of source code. In this paper we approach the software mining task with a combination of text mining and link analysis techniques. We discuss how each instance (i.e. a programming construct such as a class or a method) can be converted into a feature vector that combines the information about how the instance is interlinked with other instances, and the information about its (textual) content. The so-obtained feature vectors serve as the basis for the construction of the domain ontology with OntoGen, an existing system for semi-automatic data-driven ontology construction.

Keywords: software mining, text mining, link analysis, graph and network theory, feature vectors, ontologies, OntoGen, machine learning

1 Introduction and Motivation

Many data mining (i.e. knowledge discovery) techniques are these days in use for ontology learning – text mining, Web mining, graph mining, network analysis, link analysis, relation data mining, stream mining, and so on [6]. In the current state-of-the-art bundle mining of software code and the associated documentations is not explicitly addressed. With the growing amounts of software, especially open-source software libraries, we argue that mining such data is worth considering as a new methodology. Thus we introduce the term “software mining” to refer to such methodology. The term denotes the process of extracting knowledge (i.e. useful information) out of data sources that typically accompany an open-source software library.

The motivation for software mining comes from the fact that the discovery of reusable software artifacts is just as important as the discovery of documents and multimedia contents. According to the recent Semantic Web trends, contents need to be semantically annotated with concepts from the domain ontology in order to be discoverable by intelligent agents. Because the legacy content repositories are relatively large, cheaper semi-automatic means for semantic annotation and domain

ontology construction are preferred to the expensive manual labor. Furthermore, when dealing with software artifacts it is possible to go beyond discovery and also support other user tasks such as composition, orchestration, and execution. The need for ontology-based systems has yielded several research and development projects supported by EU that deal with this issue. One of these projects is TAO (<http://www.tao-project.eu>) which stands for Transitioning Applications to Ontologies. In this paper we present work in the context of software mining for the domain ontology construction. We illustrate the proposed approach on the software mining case study based on GATE [3], an open-source software library for natural-language processing written in Java programming language.

We interpret “software mining” as being a combination of methods for structure mining and for content mining. To be more specific, we approach the software mining task with the techniques used for text mining and link analysis. The GATE case study serves as a perfect example in this perspective. On concrete examples we discuss how each instance (i.e. a programming construct such as a class or a method) can be represented as a feature vector that combines the information about how the instance is interlinked with other instances, and the information about its (textual) content. The so-obtained feature vectors serve as the basis for the construction of the domain ontology with OntoGen [4], a system for semi-automatic, data-driven ontology construction, or by using traditional machine learning algorithms such as clustering, classification, regression, or active learning.

2 Related Work

When studying the literature we did not limit ourselves to ontology learning in the context of software artifacts – the reason for this is in the fact that the more general techniques also have the potential to be adapted for software mining.

Several knowledge discovery (mostly machine learning) techniques have been employed for ontology learning in the past. Unsupervised learning, classification, active learning, and feature space visualization form the core of OntoGen [4]. OntoGen employs text mining techniques to facilitate the construction of an ontology out of a set of textual documents. Text mining seems to be a popular approach to ontology learning because there are many textual sources available (one of the largest is the Web). Furthermore, text mining techniques are shown to produce relatively good results. In [8], the authors provide a lot of insight into the ontology learning in the context of the Text-To-Onto ontology learning architecture. The authors employ a multi-strategy learning approach and result combination (i.e. they combine outputs of several different algorithms) to produce a coherent ontology definition. In this same work a comprehensive survey of ontology learning approaches is presented.

Marta Sabou’s thesis [13] provides valuable insights into ontology learning for Web Services. It summarizes ontology learning approaches, ontology learning tools, acquisition of software semantics, and describes – in detail – their framework for learning Web Service domain ontologies.

There are basically two approaches to building tools for software component discovery: the information retrieval approach and the knowledge-based approach. The

first approach is based on the natural language documentation of the software components. With this approach no interpretation of the documentation is made – the information is extracted via statistical analyses of the words distribution. On the other hand, the knowledge-based approach relies on pre-encoded, manually provided information (the information is provided by a domain expert). Knowledge-based systems can be “smarter” than IR systems but they suffer from the scalability issue (extending the repository is not “cheap”).

In [9], the authors present techniques for browsing amongst functionality related classes (rather than inheritance), and retrieving classes from object-oriented libraries. They chose the IR approach for which they believe is advantageous in terms of cost, scalability, and ease of posing queries. They extract information from the source code (a structured data source) and its associated documentation (an unstructured data source). First, the source code is parsed and the relations, such as derived-from or member-of, are extracted. They used a hierarchical clustering technique to form a browse hierarchy that reflected the degree of similarity between classes (the similarity is drawn from the class documentation rather than from the class structure). The similarity between two classes was inferred from the browse hierarchy with respect to the distance of the two classes from their common parent and the distance of their common parent from the root node.

In this paper we adopt some ideas from [9]. However, the purpose of our methodology is not to build browse hierarchies but rather to describe programming constructs with feature vectors that can be used for machine learning. In other words, the purpose of our methodology is to transform a source code repository into a feature space. The exploration of this feature space enables the domain experts to build a knowledge base in a “cheaper” semi-automatic interactive fashion.

3 Mining Content and Structure of Software Artifacts

In this section we present our approach and give an illustrative example of data preprocessing from documented source code using the GATE software library. In the context of the GATE case study the content is provided by the reference manual (textual descriptions of Java classes and methods), source code comments, programmer’s guide, annotator’s guide, user’s guide, forum, and so on. The structure is provided implicitly from these same data sources since a Java class or method is often referenced from the context of another Java class or method (e.g. a Java class name is mentioned in the comment of another Java class). Additional structure can be harvested from the source code (e.g. a Java class contains a member method that returns an instance of another Java class), code snippets, and usage logs (e.g. one Java class is often instantiated immediately after another). In this paper we limit ourselves to the source code which also represents the reference manual (the so called *JavaDoc*) since the reference manual is generated automatically out of the source code comments by a documentation tool.

A software-based domain ontology should provide two views on the corresponding software library: the view on the data structures and the view on the functionality [13]. In GATE, these two views are represented with Java classes and their member

methods – these are evident from the GATE source code. In our examples we limit ourselves to Java classes (i.e. we deal with the part of the domain ontology that covers the data structures of the system). This means that we will use the GATE Java classes as text mining instances (and also as graph vertices when dealing with the structure).

Let us first take a look at a typical GATE Java class. It contains the following bits of information relevant for the understanding of this example (see also Fig. 1):

- **Class comment.** It should describe the purpose of the class. It is used by the documentation tool to generate the reference manual (i.e. JavaDoc).

It is mainly a source of textual data but also provides structure – two classes are interlinked if the name of one class is mentioned in the comment of the other class.

- **Class name.** Each class is given a name that uniquely identifies the class. The name is usually a composed word that captures the meaning of the class.

It is mainly a source of textual data but also provides structure – two classes are interlinked if they share a common substring in their names.

- **Field names and types.** Each class contains a set of member fields. Each field has a name (which is unique within the scope of the class) and a type. The type of a field corresponds to a Java class.

Field names provide textual data. Field types mainly provide structure – two classes are interlinked if one class contains a field that instantiates the other class.

- **Field and method comments.** Fields and methods can also be commented. The comment should explain the purpose of the field or method.

These comments are a source of textual data. They can also provide structure in the same sense as class comments do.

- **Method names and return types.** Each class contains a set of member methods. Each method has a name, a set of parameters, and a return type. The return type of a method corresponds to a Java class. Each parameter has a name and a type which corresponds to a Java class.

Methods can be treated similarly to fields with respect to taking their names and return types into account. Parameter types can be taken into account similarly to return types but there is a semantic difference between the two pieces of information. Parameter types denote classes that are “used/consumed” for processing while return types denote classes that are “produced” in the process.

- **Information about inheritance and interface implementation.** Each class inherits (fields and methods) from a base class. Furthermore, a class can implement one or more interfaces. An interface is merely a set of methods that need to be implemented in the derived class.

The information about inheritance and interface implementation is a source of structural information.

3.1 Textual Content

Textual content is taken into account by assigning a textual document to each unit of the software code – in our illustrative example, to each GATE Java class. Suppose we focus on a particular arbitrary class – there are several ways to form the corresponding document.

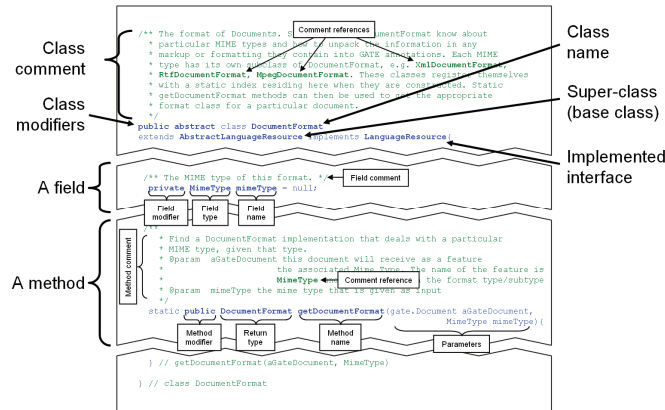


Fig. 1. Relevant parts of a typical Java class.

It is important to include only those bits of text that are not misleading for the text mining algorithms. At this point the details of these text mining algorithms are pretty irrelevant provided that we can somehow evaluate the domain ontology that we build in the end.

Another thing to consider is how to include composed names of classes, fields, and methods into a document. We can insert each of these as:

- a composed word (i.e. in its original form, e.g. “XmlDocumentFormat”),
- separate words (i.e. by inserting spaces, e.g. “Xml Document Format”), or
- combination of both (e.g. “XmlDocumentFormat Xml Document Format”).

The text-mining algorithms perceive two documents that have many words in common more similar than those that only share a few or no words. Breaking composed names into separate words therefore results in a greater similarity between documents that do not share full names but do share some parts of these names.

3.2 Determining the Structure

The basic units of the software code – in our case the Java classes – that we use as text-mining instances are interlinked in many ways. In this section we discuss how this structure which is often implicit can be determined from the source code.

As already mentioned, when dealing with the structure, we represent each class (i.e. each text mining instance) by a vertex in a graph. We can create several graphs – one for each type of associations between classes. This section describes several graphs that can be constructed out of object-oriented source code.

Comment Reference Graph. Every comment found in a class can reference another class by mentioning its name (for whatever the reason may be). In Fig. 1 we can see four such references, namely the class *DocumentFormat* references classes *XmlDocumentFormat*, *RtfDocumentFormat*, *MpegDocumentFormat*, and *MimeType*

used to weight the edges. Edges with weights lower than 0.6 and vertices of degree 0 were removed to simplify the visualization. In Fig. 3 we have removed class names and weight values to clearly show the structure. The evident clustering of vertices is the result of the Kamada-Kawai graph drawing algorithm [14] employed by Pajek [1] which was used to create graph drawings in this paper. The Kamada-Kawai algorithm positions vertices that are highly interlinked closer together.

Type Reference Graph. Field types and method return types are a valuable source of structural information. A field type or a method return type can correspond to a class in the scope of the study (i.e. a class that is also found in the source code repository under consideration) – hence an arc can be drawn from the class to which the field or the method belongs towards the class represented by the type.

Inheritance and Interface Implementation Graph. Last but not least, structure can also be determined from the information about inheritance and interface implementation. This is the most obvious structural information in an object-oriented source code and is often used to arrange classes into the browsing taxonomy. In this graph, an arc that connects two vertices is directed from the vertex that represents a base class (or an interface) towards the vertex that represents a class that inherits from the base class (or implements the interface). The weight of an arc is always 1.

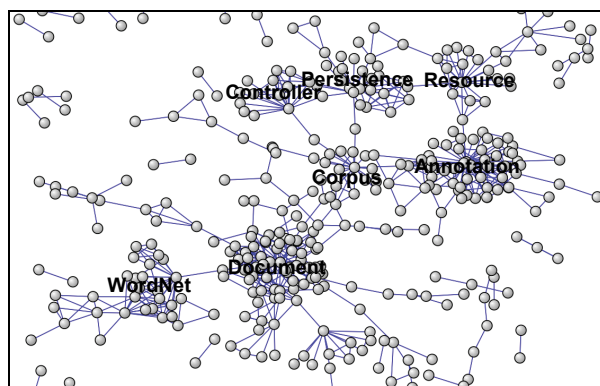


Fig. 3. The GATE name similarity graph. The most common substrings in names are shown for the most evident clusters.

4 Transforming Content and Structure into Feature Vectors

Many data-mining algorithms work with feature vectors. This is true also for the algorithms employed by OntoGen and for the traditional machine learning algorithms such as clustering or classification. Therefore we need to convert the content (i.e.

documents assigned to text-mining instances) and the structure (i.e. several graphs of interlinked vertices) into feature vectors. Potentially we also want to include other explicit features (e.g. in- and out-degree of a vertex).

4.1 Converting Content into Feature Vectors

To convert textual documents into feature vectors we resort to a well-known text mining approach. We first apply stemming¹ to all the words in the document collection (i.e. we normalize words by stripping them of their suffixes, e.g. *stripping* \rightarrow *strip*, *suffixes* \rightarrow *suffix*). We then search for *n-grams*, i.e. sequences of consecutive words of length n that occur in the document collection more than a certain amount of times [11]. Discovered *n-grams* are perceived just as all the other (single) words. After that, we convert documents into their bag-of-words representations. To weight words (and *n-grams*), we use the TF-IDF weighting scheme ([6], Section 1.3.2).

4.2 Converting Structure into Feature Vectors

Let us repeat that the structure is represented in the form of several graphs in which vertices correspond to text-mining instances. If we consider a particular graph, the task is to describe each vertex in the graph with a feature vector.

For this purpose we adopt the technique presented in [10]. First, we convert arcs (i.e. directed links) into edges (i.e. undirected links)². The edges adopt weights from the corresponding arcs. If two vertices are directly connected with more than one arc, the resulting edge weight is computed by summing, maximizing, minimizing, or averaging the arc weights (we propose summing the weights as the default option). Then we represent a graph on N vertices as a $N \times N$ sparse matrix. The matrix is constructed so that the X th row gives information about vertex X and has nonzero components for the columns representing vertices from the neighborhood of vertex X . The neighborhood of a vertex is defined by its (restricted) domain. The *domain of a vertex* is the set of vertices that are path-connected to the vertex. More generally, a *restricted domain of a vertex* is a set of vertices that are path-connected to the vertex at a maximum distance of d_{max} steps [1]. The X th row thus has a nonzero value in the X th column (because vertex X has zero distance to itself) as well as nonzero values in all the other columns that represent vertices from the (restricted) domain of vertex X . A value in the matrix represents the importance of the vertex represented by the column for the description of the vertex represented by the row. In [10] the authors propose to compute the values as $1/2^d$, where d is the minimum path length between the two vertices (also termed the *geodesic* distance between two vertices) represented by the row and column.

¹ We use the Porter stemmer for English (see <http://de.wikipedia.org/wiki/Porter-Stemmer-Algorithmus>).

² This is not a required step but it seems reasonable – a vertex is related to another vertex if they are interconnected regardless of the direction. In other words, if vertex A *references* vertex B then vertex B *is referenced* by vertex A .

We also need to include edge weights into account. The easiest way is to use the weights merely for thresholding. This means that we set a threshold and remove all the edges that have weights below this threshold. After that we construct the matrix which now indirectly includes the information about the weights (at least to a certain extent).

The simple approach described above is based on more sophisticated approaches such as ScentTrails [12]. The idea is to metaphorically “waft” scent of a specific vertex in the direction of its out-links (links with higher weights conduct more scent than links with lower weights – the arc weights are thus taken into account explicitly). The scent is then iteratively spread throughout the graph. After that we can observe how much of the scent reached each of the other vertices. The amount of scent that reached a target vertex denotes the importance of the target vertex for the description of the source vertex.

The ScentTrails algorithm shows some similarities with the probabilistic framework: starting in a particular vertex and moving along the arcs we need to determine the probability of ending up in a particular target vertex within m steps. At each step we can select one of the available outgoing arcs with the probability proportional to the corresponding arc weight (assuming that the weight denotes the strength of the association between the two vertices). The equations for computing the probabilities are fairly easy to derive (see [7], Appendix C) but the time complexity of the computation is higher than that of ScentTrails and the first presented approach. The probabilistic framework is thus not feasible for large graphs.

4.3 Joining Different Representations into a Single Feature Vector

The next issue to solve is how to create a feature vector for a vertex that is present in several graphs at the same time (remember that the structure can be represented with more than one graph) and how to then also “append” the corresponding content feature vector. In general, this can be done in two different ways:

- **Horizontally.** This means that feature vectors of the same vertex from different graphs are first multiplied by factors α_i ($i = 1, \dots, M$) and then concatenated into a feature vector with $M \times N$ components (M being the number of graphs and N the number of vertices). The content feature vector is multiplied by α_{M+1} and simply appended to the resulting structure feature vector.
- **Vertically.** This means that feature vectors of the same vertex from different graphs are first multiplied by factors α_i ($i = 1, \dots, M$) and then summed together (component-wise) resulting in a feature vector with N components (N being the number of vertices). Note that the content feature vector cannot be summed together with the resulting structure feature vector since the features contained therein carry a different semantic meaning (not to mention that the two vectors are not of the same length). Therefore also in this case, the content feature vector is multiplied by α_{M+1} and appended to the resulting structure feature vector.

Fig. 4 illustrates these two approaches. A factor α_i ($i = 1, \dots, M$) denotes the importance of information provided by graph i , relative to the other graphs. Factor α_{M+1} , on the other hand, denotes the importance of information provided by the content relative to the information provided by the structure. The easiest way to set

the factors is to either include the graph or the content (i.e. $\alpha_i = 1$), or to exclude it (i.e. $\alpha_i = 0$). In general these factors can be quite arbitrary. Pieces of information with lower factors contribute less to the outcomes of similarity measures used in clustering algorithms than those with higher factors. Furthermore, many classifiers are sensitive to this kind of weighting. For example, it has been shown in [2] that the SVM regression model is sensitive to how this kind of factors are set.

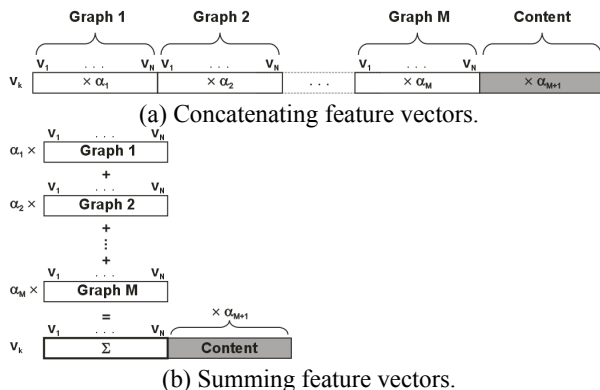


Fig. 4. The two different ways of joining several different representations of the same instance.

OntoGen includes a feature-space visualization tool called Document Atlas [5]. It is capable of visualizing high-dimensional feature space in two dimensions. The feature vectors are presented with two-dimensional points while the Euclidean distances between these points reflect cosine distances between feature vectors. It is not possible to perfectly preserve the distances from the high-dimensional space but even an approximation gives the user an idea of how the feature space looks like. Fig. 5 shows two such visualizations of the GATE case study data. In the left figure, only the class comments were taken into account (i.e. all the structural information was ignored and the documents assigned to the instances consisted merely of the corresponding class comments). In the right figure the information from the name similarity graph was added to the content information from the left figure. The content information was weighted twice higher than the structural information. d_{max} of the name similarity graph was set to 0.44.

The cluster marked in the left figure represents classes that provide functionality to consult WordNet (see <http://wordnet.princeton.edu>) to resolve synonymy³. The cluster containing this same functionality in the right figure is also marked. However, the cluster in the right figure contains more classes many of which were not commented thus were not assigned any content⁴. Contentless classes are stuck in the top left corner in the left figure because the feature-space visualization system did not know where to put them due to the lack of association with other classes. This missing

³ The marked cluster in the left figure contains classes such as Word, VerbFrame, and Synset.

⁴ The marked cluster in the right figure contains the same classes as the marked cluster in the left figure but also some contentless classes such as WordImpl, VerbFrameImpl, (Mutable)LexKBSynset(Impl), SynsetImpl, WordNetViewer, and IndexFileWordNetImpl.

association was introduced with the information from the name similarity graph. From the figures it is also possible to see that clusters are better defined in the right figure (note the dense areas represented with light color).

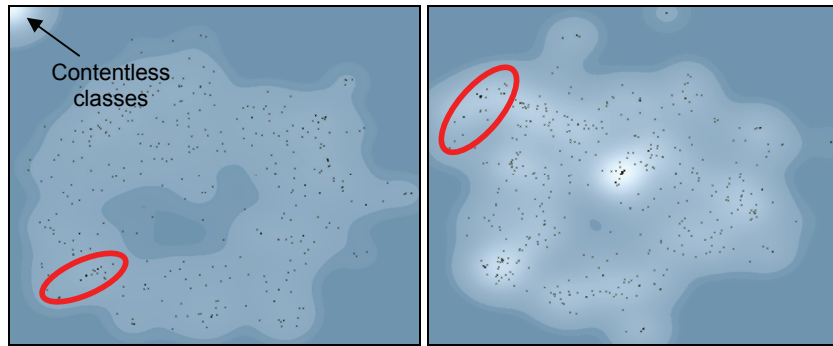


Fig. 5. Two different semantic spaces obtained by two different weighting settings.

With these visualizations we merely want to demonstrate the difference in semantic spaces between two different settings. This is important because instances that are shown closer together are more likely to belong to the same cluster or category after applying clustering or classification. The weighting setting depends strongly on the context of the application of this methodology.

5 Conclusions

In this paper we presented a methodology for transforming a source code repository into a set of feature vectors, i.e. into a feature space. These feature vectors serve as the basis for the construction of the domain ontology with OntoGen, a system for semi-automatic data-driven ontology construction, or by using traditional machine learning algorithms such as clustering, classification, regression, or active learning. The presented methodology thus facilitates the transitioning of legacy software repositories into state-of-the-art ontology-based systems for discovery, composition, and potentially also execution of software artifacts.

This paper does not provide any evaluation of the presented methodology. Basically, the evaluation can be performed either by comparing the resulting ontologies with a golden-standard ontology (if such ontology exists) or, on the other hand, by employing them in practice. In the second scenario, we measure the efficiency of the users that are using these ontologies (directly or indirectly) in order to achieve certain goals. The aspects on the quality of the methods presented herein will be the focus of our future work.

We recently started developing an ontology-learning framework named LATINO which stands for Link-analysis and text-mining toolbox [7]. LATINO will be an open-source general purpose data mining platform providing (mostly) text mining, link analysis, machine learning, and data visualization capabilities.

Acknowledgments. This work was supported by the Slovenian Research Agency and the IST Programme of the European Community under TAO Transitioning Applications to Ontologies (IST-4-026460-STP) and PASCAL Network of Excellence (IST-2002-506778).

References

1. Batagelj, V., Mrvar, A., de Nooy, W.: Exploratory Network Analysis with Pajek. Cambridge University Press (2004)
2. Brank, J., Leskovec, J.: The Download Estimation Task on KDD Cup 2003. In ACM SIGKDD Explorations Newsletter, volume 5, issue 2, 160–162, ACM Press, New York, USA (2003)
3. Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics ACL'02 (2002)
4. Fortuna, B., Grobelnik M., Mladenic D.: Semi-automatic Data-driven Ontology Construction System. In Proceedings of the 9th International Multi-conference Information Society IS-2006, Ljubljana, Slovenia (2006)
5. Fortuna, B., Mladenic, D., Grobelnik, M.: Visualization of Text Document Corpus. In Informatica 29, 497-502 (2005)
6. Grcar, M., Mladenic, D., Grobelnik, M., Bontcheva, K.: D2.1: Data Source Analysis and Method Selection. Project report IST-2004-026460 TAO, WP 2, D2.1 (2006)
7. Grcar, M., Mladenic, D., Grobelnik, M., Fortuna, B., Brank, J.: D2.2: Ontology Learning Implementation. Project report IST-2004-026460 TAO, WP 2, D2.2 (2006)
8. Maedche, A., Staab, S.: Discovering Conceptual Relations from Text. In Proc. of ECAI 2000, 321–325 (2001)
9. Helm, R., Maarek, Y.: Integrating Information Retrieval and Domain Specific Approaches for Browsing and Retrieval in Object-oriented Class Libraries. In Proceedings of Object-oriented Programming Systems, Languages, and Applications, 47–61, ACM Press, New York, USA (1991)
10. Mladenic, D., Grobelnik, M.: Visualizing Very Large Graphs Using Clustering Neighborhoods. In Local Pattern Detection, Dagstuhl Castle, Germany, April 12–16 (2004)
11. Mladenic, D., Grobelnik, M.: Word Sequences as Features in Text Learning. In Proceedings of the 17th Electrotechnical and Computer Science Conference ERK-98, Ljubljana, Slovenia (1998)
12. Olston, C., Chi, H. E.: ScentTrails: Integrating Browsing and Searching on the Web. In ACM Transactions on Computer-human Interaction TOCHI, volume 10, issue 3, 177–197, ACM Press, New York, USA (2003)
13. Sabou, M.: Building Web Service Ontologies. In SIKS Dissertation Series No. 2004-4, ISBN 90-9018400-7 (2006)
14. Kamada, T., Kawai, S.: An Algorithm for Drawing General Undirected Graphs. In Information Processing Letters 31, 7–15 (1989)

Generalization-based Similarity for Conceptual Clustering

S. Ferilli, T.M.A. Basile, N. Di Mauro, M. Biba, and F. Esposito

Dipartimento di Informatica
Università di Bari
via E. Orabona, 4 - 70125 Bari - Italia
{ferilli, basile, ndm, biba, esposito}@di.uniba.it

Abstract. Knowledge extraction represents an important issue that concerns the ability to identify valid, potentially useful and understandable patterns from large data collections. Such a task becomes more difficult if the domain of application cannot be represented by means of an attribute-value representation. Thus, a more powerful representation language, such as First-Order Logic, is necessary. Due to the complexity of handling First-Order Logic formulæ, where the presence of relations causes various portions of one description to be possibly mapped in different ways onto another description, few works presenting techniques for comparing descriptions are available in the literature for this kind of representations. Nevertheless, the ability to assess similarity between first-order descriptions has many applications, ranging from description selection to flexible matching, from instance-based learning to clustering. This paper tackles the case of Conceptual Clustering, where a new approach to similarity evaluation, based on both syntactic and semantic features, is exploited to support the task of grouping together similar items according to their relational description. After presenting a framework for Horn Clauses (including criteria, a function and composition techniques for similarity assessment), classical clustering algorithms are exploited to carry out the grouping task. Experimental results on real-world datasets prove the effectiveness of the proposal.

1 Introduction

The large amount of information available nowadays makes more difficult the task of extracting useful knowledge, i.e. valid, potentially useful and understandable patterns, from data collections. Such a task becomes more difficult if the collection requires a more powerful representation language than simple attribute-value vectors. First-order logic (FOL for short) is a powerful formalism, that is able to express relations between objects and hence can overcome the limitations shown by propositional or attribute-value representations. However, the presence of relations causes various portions of one description to be possibly mapped in different ways onto another description, which poses problems of computational effort when two descriptions have to be compared to each other.

Specifically, an important subclass of FOL refers to sets of *Horn clauses*, i.e. logical formulæ of the form $l_1 \wedge \dots \wedge l_n \Rightarrow l_0$ where the l_i 's are *atoms*, usually represented in Prolog style as $l_0 :- l_1, \dots, l_n$ to be interpreted as “ l_0 (called *head* of the clause) is true, provided that l_1 and ... and l_n (called *body* of the clause) are all true”. Without loss of generality [16], we will deal with the case of linked Datalog clauses.

The availability of techniques for the comparison between FOL (sub-)descriptions could have many applications: helping a subsumption procedure to converge quickly, guiding a generalization procedure by focussing on the components that are more similar and hence more likely to correspond to each other, implementing flexible matching, supporting instance-based classification techniques or conceptual clustering. Cluster analysis concerns the organization of a collection of unlabeled patterns into groups (clusters) of homogeneous elements based on their similarity. The similarity measure exploited to evaluate the distance between elements is responsible for the effectiveness of the clustering algorithms. Hence, the comparison techniques are generally defined in terms of a metric that must be carefully constructed if the clustering is to be relevant. In supervised clustering there is an associated output class value for each element and the efficacy of the metric exploited for the comparison of elements is evaluated according to the principle that elements belonging to the same class are clustered together as much as possible.

In the following sections, a similarity framework for first-order logic clauses will be presented. Then, Section 5 will deal with related work, and Section 6 will show how the proposed formula and criteria are able to effectively guide a clustering procedure for FOL descriptions. Lastly, Section 7 will conclude the paper and outline future work directions.

2 Similarity Formula

Intuitively, the evaluation of similarity between two items i' and i'' might be based both on the presence of common features, which should concur in a positive way to the similarity evaluation, and on the features of each item that are not owned by the other, which should concur negatively to the whole similarity value assigned to them [10]. Thus, plausible similarity parameters are:

- n , the number of features owned by i' but not by i'' (*residual* of i' wrt i'');
- l , the number of features owned both by i' and by i'' ;
- m , the number of features owned by i'' but not by i' (*residual* of i'' wrt i').

A novel similarity function that expresses the degree of similarity between i' and i'' based on the above parameters, developed to overcome some limitations of other functions in the literature (e.g., Tverski's, Dice's and Jaccard's), is:

$$sf(i', i'') = sf(n, l, m) = 0.5 \frac{l+1}{l+n+2} + 0.5 \frac{l+1}{l+m+2} \quad (1)$$

It takes values in $]0, 1[$, to be interpreted as the degree of similarity between the two items. A complete overlapping of the two items tends to the limit of 1

as long as the number of common features grows. The full-similarity value 1 is never reached, and is reserved to the exact identification of items, i.e. $i' = i''$ (in the following, we assume $i' \neq i''$). Conversely, in case of no overlapping the function will tend to 0 as long as the number of non-shared features grows. This is consistent with the intuition that there is no limit to the number of different features owned by the two descriptions, which contribute to make them ever different. Since each of the two terms refers specifically to one of the two clauses under comparison, a weight could be introduced to give different importance to either of the two.

3 Similarity Criteria

The main contribution of this paper is in the exploitation of the formula in various combinations that can assign a similarity degree to the different clause constituents. In FOL formulæ, terms represent specific objects; unary predicates represent term properties and n -ary predicates express relationships. Hence, two levels of similarity between first-order descriptions can be defined: the *object* level, concerning similarities between terms in the descriptions, and the *structure* one, referring to how the nets of relationships in the descriptions overlap.

Example 1. Let us consider, as a running example throughout the paper, the following toy clause (a real-world one would be too complex):

$$C : h(a) :- p(a, b), p(a, c), p(d, a), r(b, f), o(b, c), q(d, e), t(f, g), \\ \pi(a), \phi(a), \sigma(a), \tau(a), \sigma(b), \tau(b), \phi(b), \tau(d), \rho(d), \pi(f), \phi(f), \sigma(f).$$

3.1 Object Similarity

Consider two clauses C' and C'' . Call $A' = \{a'_1, \dots, a'_n\}$ the set of terms in C' , and $A'' = \{a''_1, \dots, a''_m\}$ the set of terms in C'' . When comparing a pair of objects $(a', a'') \in A' \times A''$, two kinds of object features can be distinguished: the properties they own as expressed by unary predicates (*characteristic features*), and the roles they play in n -ary predicates (*relational features*). More precisely, a *role* can be seen as a couple $R = (\text{predicate}, \text{position})$ (written compactly as $R = \text{predicate}/\text{arity.position}$), since different positions actually refer to different roles played by the objects. For instance, a characteristic feature could be **male**(X), while relational features in a **parent**(X,Y) predicate are the ‘parent’ role (*parent/2.1*) the ‘child’ role (*parent/2.2*).

Two corresponding similarity values can be associated to a' and a'' : a *characteristic similarity*,

$$\text{sf}_c(a', a'') = \text{sf}(n_c, l_c, m_c)$$

based on the set P' of properties related to a' and the set P'' of properties related to a'' , for the following parameters:

$n_c = |P' \setminus P''|$ number of properties owned by a' in C' but not by a'' in C''
(*characteristic residual* of a' wrt a'');

$l_c = |P' \cap P''|$ number of common properties between a' in C' and a'' in C'' ;
 $m_c = |P'' \setminus P'|$ number of properties owned by a'' in C'' but not by a' in C'
 (characteristic residual of a'' wrt a').

and a *relational similarity*,

$$\text{sf}_r(a', a'') = \text{sf}(n_r, l_r, m_r)$$

based on the *multisets* R' and R'' of roles played by a' and a'' , respectively, for the following parameters:

$n_r = |R' \setminus R''|$ how many times a' plays in C' role(s) that a'' does not play in C'' (*relational residual* of a' wrt a'');
 $l_r = |R' \cap R''|$ number of times that both a' in C' and a'' in C'' play the same role(s);
 $m_r = |R'' \setminus R'|$ how many times a'' plays in C'' role(s) that a' does not play in C' (*relational residual* of a'' wrt a').

Overall, we can define the *object similarity* between two terms as

$$\text{sf}_o(a', a'') = \text{sf}_c(a', a'') + \text{sf}_r(a', a'')$$

Example 2. Referring to clause C , the set of properties of a is $\{\pi, \phi, \sigma, \tau\}$, for b it is $\{\sigma, \tau\}$ and for c it is $\{\phi\}$. The multiset of roles of a is $\{p/2.1, p/2.1, p/2.2\}$, for b it is $\{p/2.2, r/2.1, o/2.1\}$ and for c it is $\{p/2.2, o/2.2\}$.

3.2 Structural Similarity

When checking for the structural similarity of two formulæ, many objects can be involved, and hence their mutual relationships represent a constraint on how each of them in the former formula can be mapped onto another in the latter. The structure of a formula is defined by the way in which n -ary *atoms* (predicates applied to a number of terms equal to their arity) are applied to the various objects to relate them. This is the most difficult part, since relations are specific to the first-order setting and are the cause of indeterminacy in mapping (parts of) a formula into (parts of) another one. In the following, we will call *compatible* two FOL (sub-)formulæ that can be mapped onto each other without yielding inconsistent term associations (i.e., a term in one formula cannot correspond to different terms in the other formula).

Given an n -ary literal, we define its *star* as the multiset of n -ary predicates corresponding to the literals linked to it by some common term (a predicate can appear in multiple instantiations among these literals). The *star similarity* between two compatible n -ary literals l' and l'' having stars S' and S'' , respectively, can be computed for the following parameters:

$n_s = |S' \setminus S''|$ how many more relations l' has in C' than l'' has in C'' (*star residual* of l' wrt l'');

$l_s = |S' \cap S''|$ number of relations that both l' in C' and l'' in C'' have in common;
 $m_s = |S'' \setminus S'|$ how many more relations l'' has in C'' than l' has in C' (*star residual* of l'' wrt l').

by taking into account also the object similarity values for all pairs of terms included in the association θ that map l' onto l'' of their arguments in corresponding positions:

$$\text{sf}_s(l', l'') = \text{sf}(n_s, l_s, m_s) + C^s(\{\text{sf}_o(t', t'')\}_{t'/t'' \in \theta})$$

where C^s is a composition function (e.g., the average).

Then, Horn clauses can be represented as a graph in which atoms are the nodes, and edges connect two nodes *iff* they share some term, as described in the following. In particular, we will deal with *linked* clauses only (i.e. clauses whose associated graph is connected). Given a clause C , we define its *associated graph* G_C , where the edges to be represented form a Directed Acyclic Graph (DAG), *stratified* in such a way that the head is the only node at level 0 and each successive level is made up by nodes not yet reached by edges that have at least one term in common with nodes in the previous level. In particular, each node in the new level is linked by an incoming edge to each node in the previous level having among its arguments at least one term in common with it.

Example 3. In the graph G_C , the head represents the 0-level of the stratification. Then directed edges may be introduced from $h(X)$ to $p(X, Y)$, $p(X, Z)$ and $p(W, X)$, which yields level 1 of the stratification. Now the next level can be built, adding directed edges from atoms in level 1 to the atoms not yet considered that share a variable with them: $r(Y, U)$ – end of an edge starting from $p(X, Y)$ –, $o(Y, Z)$ – end of edges starting from $p(X, Y)$ and $p(X, Z)$ – and $q(W, W)$ – end of an edge starting from $p(W, X)$. The third level of the graph includes the only remaining atom, $s(U, V)$ – having an incoming edge from $r(Y, U)$.

Now, all possible paths starting from the head and reaching *leaf* nodes are univoquely determined, which reduces the amount of indeterminacy in the comparison. Given two clauses C' and C'' , we define the *intersection* between two paths $p' = \langle l'_1, \dots, l'_{n'} \rangle$ in $G_{C'}$ and $p'' = \langle l''_1, \dots, l''_{n''} \rangle$ in $G_{C''}$ as the pair of longest compatible initial subsequences of p' and p'' :

$p' \cap p'' = (p_1, p_2) = (\langle l'_1, \dots, l'_k \rangle, \langle l''_1, \dots, l''_k \rangle)$ s.t.
 $\forall i = 1, \dots, k : l'_1, \dots, l'_i$ compatible with $l''_1, \dots, l''_i \wedge$
 $(k = n' \vee k = n'' \vee l'_1, \dots, l'_{k+1}$ incompatible with $l''_1, \dots, l''_{k+1})$
 and the two residuals as the incompatible trailing parts:

$$p' \setminus p'' = \langle l'_{k+1}, \dots, l'_{n'} \rangle \quad p'' \setminus p' = \langle l''_{k+1}, \dots, l''_{n''} \rangle$$

Hence, the *path similarity* between p' and p'' , $\text{sf}_s(p', p'')$, can be computed by applying (1) to the following parameters:

$n_p = |p' \setminus p''| = n' - k$ is the length of the trail incompatible sequence of p' wrt p'' (*path residual* of p' wrt p'');

$l_p = |p_1| = |p_2| = k$ is the length of the maximum compatible initial sequence of p' and p'' ;
 $m_p = |p'' \setminus p'| = n'' - k$ is the length of the trail incompatible sequence of p'' wrt p' (*path residual* of p'' wrt p').

by taking into account also the star similarity values for all pairs of literals associated by the initial compatible sequences:

$$\text{sf}_p(p', p'') = \text{sf}(n_p, l_p, m_p) + C^p(\{\text{sf}_s(l'_i, l''_i)\}_{i=1, \dots, k})$$

where C^p is a composition function (e.g., the average).

Example 4. In C , the star of $p(a, b)$ is the multiset $\{p/2, p/2, r/2, o/2\}$, while that of $p(a, c)$ is $\{p/2, p/2, o/2\}$. The paths in C (ignoring the head that, being unique, can be univoquely matched) are $\{ \langle p(a, b), r(b, f), t(f, g) \rangle, \langle p(a, b), o(b, c) \rangle, \langle p(a, c), o(b, c) \rangle, \langle p(d, a), q(d, e) \rangle \}$.

Note that no single criterion is by itself neatly discriminant, but their cooperation succeeds in assigning sensible similarity values to the various kinds of components, and in distributing on each kind of component a proper portion of the overall similarity, so that the difference becomes ever clearer as long as they are composed one atop the previous ones.

4 Clause Similarity

Now, similarity between two (tuples of) terms reported in the head predicates of two clauses, according to their description reported in the respective bodies, can be computed based on their generalization. In particular, one would like to exploit their *least general generalization*, i.e. the most specific model for the given pair of descriptions. Unfortunately, such a generalization is not easy to find: either classical θ -subsumption is used as a generalization model, and then one can compute Plotkin's least general generalization [13], at the expenses of some undesirable side-effects concerning the need of computing its reduced equivalent (and also of some counter-intuitive aspects of the result), or, as most ILP learners do, one requires the generalization to be a subset of the clauses to be generalized. In the latter option, that we choose for the rest of the work, the θ_{OI} generalization model [5], based on the Object Identity assumption, represents a supporting framework with solid theoretical foundations to be exploited.

Given two clauses C' and C'' , call $C = \{l_1, \dots, l_k\}$ their least general generalization, and consider the substitutions θ' and θ'' such that $\forall i = 1, \dots, k : l_i \theta' = l'_i \in C'$ and $l_i \theta'' = l''_i \in C''$, respectively. Thus, a formula for assessing the overall similarity between C' and C'' , called *formulæ similitudo* and denoted fs , can be computed according to the amounts of common and different literals:

$n = |C'| - |C|$ how many literals in C' are not covered by its least general generalization with respect to C'' (*clause residual* of C' wrt C'');

$l = |C| = k$ maximal number of literals that can be put in correspondence between C' and C'' according to their least general generalization;
 $m = |C''| - |C|$ how many literals in C'' are not covered by its least general generalization with respect to C' (*clause residual* of C'' wrt C').

and of common and different objects:

$n_o = |terms(C')| - |terms(C)|$ how many terms in C' are not associated by its least general generalization to terms in C'' (*object residual* of C' wrt C'');
 $l_o = |terms(C)|$ maximal number of terms that can be put in correspondence in C' and C'' as associated by their least general generalization;
 $m_o = |terms(C'')| - |terms(C)|$ how many terms in C'' are not associated by its least general generalization to terms in C' (*object residual* of C'' wrt C').

by taking into account also the star similarity values for all pairs of literals associated by the least general generalization:

$$sf(C', C'') = sf(n, l, m) \cdot sf(n_o, l_o, m_o) + C^c(\{sf_s(l'_i, l''_i)\}_{i=1, \dots, k})$$

where C^c is a composition function (e.g., the average). This function evaluates the similarity of two clauses according to the composite similarity of a maximal subset of their literals that can be put in correspondence (which includes both structural and object similarity), smoothed by adding the overall similarity in the number of overlapping and different literals and objects between the two (whose weight in the final evaluation should not overwhelm the similarity coming from the detailed comparisons, hence the multiplication).

In particular, the similarity formula itself can be exploited for computing the generalization. The path intersections are considered by decreasing similarity, adding to the partial generalization generated thus far the common literals of each pair whenever they are compatible [6]. The proposed similarity framework proves actually able to lead towards the identification of the proper sub-parts to be put in correspondence in the two descriptions under comparison, as shown indirectly by the portion of literals in the clauses to be generalized that is preserved by the generalization. More formally, the compression factor (computed as the ratio between the length of the generalization and that of the shortest clause to be generalized) should be as high as possible. Interestingly, on the document dataset (see section 6 for details) the similarity-driven generalization preserved on average more than 90% literals of the shortest clause, with a maximum of 99,48% (193 literals out of 194, against an example of 247) and just 0,006 variance. As a consequence, one would expect that the produced generalizations are least general ones or nearly so. Noteworthy, using the similarity function on the document labelling task leads to runtime savings that range from 1/3 up to 1/2, in the order of hours.

5 Related Works

Few works faced the definition of similarity or distance measures for first-order descriptions. [4] proposes a distance measure based on probability theory applied

to the formula components. Compared to that, our function does not require the assumptions and simplifying hypotheses to ease the probability handling, and no *a-priori* knowledge of the representation language is required. It does not require the user to set weights on the predicates' importance, and is not based on the presence of 'mandatory' relations, like for the *G1* subclause in [4]. *KGB* [1] uses a similarity function, parameterized by the user, to guide generalization; our approach is more straightforward, and can be easily extended to handle negative information in the clauses. In *RIBL* [3] object similarity depends on the similarity of their attributes' values and, recursively, on the similarity of the objects related to them, which poses the problem of indeterminacy. [17] presents an approach for the induction of a distance on FOL examples, that exploits the truth values of whether each clause covers the example or not as features for a distance on the space $\{0,1\}^k$ between the examples. [12] organizes terms in an importance-related hierarchy, and proposes a distance between terms based on interpretations and a level mapping function that maps every simple expression on a natural number. [14] presents a distance function between atoms based on the difference with their lgg, and uses it to compute distances between clauses. It consists of a pair where the second component allows to differentiate cases where the first component cannot.

As pointed out, we focus on the identification and exploitation of similarity measures for first-order descriptions in the clustering task. Many research efforts on data representation, elements' similarity and grouping strategies have produced several successful clustering methods (see [9] for a survey). The classical strategies can be divided in bottom-up and top-down. In the former, each element of the dataset is considered as a cluster. Successively, the algorithm tries to group the clusters that are more similar according to the similarity measure. This step is performed until the number of clusters the user requires as a final result is reached, or the minimal similarity value among clusters is greater than a given threshold. In the latter approach, known as hierarchical clustering, at the beginning all the elements of the dataset form a unique cluster. Successively, the cluster is partitioned into clusters made up of elements that are more similar according to the similarity measure. This step is performed until the number of clusters required by the user as a final result is reached. A further classification is based on whether an element can be assigned (NotExclusive or Fuzzy Clustering) or not (Exclusive or Hard Clustering) to more than one cluster. Also the strategy exploited to partition the space is a criterion used to classify the clustering techniques: in Partitive Clustering a representative point (centroid, medoid, etc.) of the cluster in the space is chosen; Hierarchical Clustering produces a nested series of partitions by merging (Hierarchical Agglomerative) or splitting (Hierarchical Divisive) clusters, Density-based Clustering considers the density of the elements around a fixed point.

Closely related to data clustering is Conceptual Clustering, a Machine Learning paradigm for unsupervised classification which aims at generating a concept description for each generated class. In conceptual clustering both the inherent structure of the data and the description language, available to the learner, drive

cluster formation. Thus, a concept (regularity) in the data could not be learned by the system if the description language is not powerful enough to describe that particular concept (regularity). This problem arises when the elements simultaneously describe several objects whose relational structures change from one element to the other. First-Order Logic representations allow to overcome these problems. However, most of the clustering algorithms and systems work on attribute-value representation (e.g., CLUSTER/2 [11], CLASSIT [8], COBWEB [7]). Other systems such as LABYRINTH [18] can deal with structured objects exploiting a representation that is not powerful enough to express the dataset in a lot of domains. There are few systems that cluster examples represented in FOL (e.g., AUTOCLASS-like [15], KBG [1]), some of which still rely on propositional distance measures (e.g., TIC [2]).

6 Experiments on Clustering

The proposed similarity framework was tested on the conceptual clustering task, where a set of items must be grouped into homogeneous classes according to the similarity between their first-order logic description. In particular, we adopted the classical K-means clustering technique. However, since first-order logic formulae do not induce an euclidean space, it was not possible to identify/build a *centroid* prototype for the various clusters according to which the next distribution in the loop would be performed. For this reason, we based the distribution on the concept of *medoid* prototypes, where a medoid is defined as the observation that actually belongs to a cluster and that has the minimum average distance from all the other members of the cluster. As to the stop criterion, it was set as the moment in which a new iteration outputs a partition already seen in previous iterations. Note that it is different than performing the same check on the set of prototypes, since different prototypes could yield the same partition, while there cannot be several different sets of prototypes for one given partition. In particular, it can happen that the last partition is the same as the last-but-one, in which case a fixed point is reached and hence a single solution has been found and has to be evaluated. Conversely, when the last partition equals a previous partition, but not the last-but-one one, a loop is identified, and one cannot focus on a single minimum to be evaluated.

Experiments on Conceptual Clustering were run on a real-world dataset¹ containing 353 descriptions of scientific papers first page layout, belonging to 4 different classes: Elsevier journals, Springer-Verlag Lecture Notes series (SVLN), Journal of Machine Learning Research (JMLR) and Machine Learning Journal (MLJ). The complexity of such a dataset is considerable, and concerns several aspects of the dataset: the journals layout styles are quite similar, so that it is not easy to grasp the difference when trying to group them in distinct classes; moreover, the 353 documents are described with a total of 67920 literals, for an average of more than 192 literals per description (some descriptions are made up of more than 400 literals); last, the description is heavily based on a *part_of*

¹ <http://lacam.di.uniba.it:8000/systems/inthelex/index.htm#datasets>

relation that increases indeterminacy. A short example of paper description (with predicate names slightly changed for the sake of brevity) is:

```

observation(d) :- num_pages(d,1), page_1(d,p1), page_w(p1,612.0), page_h(p1,792.0), last_page(p1), frame(p1,f4),
t_text(f4), w_medium_large(f4), h_very_very_small(f4), center(f4), middle(f4), frame(p1,f2), t_text(f2), w_large(f2),
h_small(f2), center(f2), upper(f2), frame(p1,f1), t_text(f1), w_large(f1), h_large(f1), center(f1), lower(f1), frame(p1,f6),
t_text(f6), w_large(f6), h_very_small(f6), center(f6), middle(f6), frame(p1,f12), t_text(f12), w_medium(f12), h_very_very_small(f12),
left(f12), middle(f12), frame(p1,f10), t_text(f10), w_large(f10), h_small(f10), center(f10), upper(f10), frame(p1,f3),
t_text(f3), w_large(f3), h_very_small(f3), center(f3), upper(f3), frame(p1,f9), t_text(f9), w_large(f9), h_medium(f9),
center(f9), middle(f9), on_top(f4,f12), to_right(f4,f12), to_right(f6,f4), on_top(f4,f6), on_top(f10,f4), to_right(f10,f4),
on_top(f2,f4), to_right(f2,f4), to_right(f1,f4), on_top(f4,f1), on_top(f3,f4), to_right(f3,f4), to_right(f9,f4), on_top(f4,f9),
on_top(f2,f12), to_right(f2,f12), on_top(f2,f6), valign_center(f2,f6), on_top(f10,f2), valign_center(f2,f10), on_top(f2,f1),
valign_center(f2,f1), on_top(f3,f2), valign_center(f2,f3), on_top(f2,f9), valign_center(f2,f9), on_top(f12,f1), to_right(f1,f12),
on_top(f6,f1), valign_center(f1,f6), on_top(f10,f1), valign_center(f1,f10), on_top(f3,f1), valign_center(f1,f3), on_top(f9,f1),
valign_center(f1,f9), on_top(f6,f12), to_right(f6,f12), on_top(f10,f6), valign_center(f6,f10), on_top(f3,f6), valign_center(f6,f3),
on_top(f9,f6), valign_center(f6,f9), on_top(f10,f12), to_right(f10,f12), on_top(f3,f12), to_right(f3,f12), on_top(f9,f12),
to_right(f9,f12), on_top(f3,f10), valign_center(f10,f3), on_top(f10,f9), valign_center(f10,f9), on_top(f3,f9), valign_center(f3,f9).

```

Since the class of each document in the dataset is known, we performed a supervised clustering: after hiding the correct class to the clustering procedure, we provided it with the ‘anonymous’ dataset, asking for a partition of 4 clusters. Then, we compared each outcoming cluster with each class, and assigned it to the best-matching class according to precision and recall. In practice, we found that for each cluster the precision-recall values were neatly high for one class, and considerably low for all the others; moreover, each cluster had a different best-matching class, so that the association and consequent evaluation became straightforward.

The clustering procedure was run first on 40 documents randomly selected from the dataset, then on 177 documents and lastly on the whole dataset, in order to evaluate its performance behaviour when tackling increasingly large data. Results are reported in Table 1: for each dataset size it reports the number of instances in each cluster and in the corresponding class, the number of matching instances between the two and the consequent precision (Prec) and recall (Rec) values, along with the overall number of correctly split documents in the dataset. Compound statistics, shown below, report the average precision and recall for each dataset size, along with the overall accuracy, plus some information about runtime and number of description comparisons to be carried out.

The overall results show that the proposed method is highly effective since it is able to autonomously recognize the original classes with precision, recall and purity (Pur) well above 80% and, for larger datasets, always above 90%. This is very encouraging, especially in the perspective of the representation-related difficulties (the lower performance on the reduced dataset can probably be explained with the lack of sufficient information for properly discriminating the clusters, and suggests further investigation). Runtime refers almost completely to the computation of the similarity between all couples of observations: computing each similarity takes on average about 2sec, which can be a reasonable time

Table 1. Experimental results

Instances	Cluster	Class	Intersection	Prec (%)	Rec (%)	Total Overlapping
40	8	Elsevier (4)	4	50	100	35
	6	SVLN (6)	5	83,33	83,33	
	8	JMLR (8)	8	100	100	
	18	MLJ (22)	18	100	81,82	
177	30	Elsevier (22)	22	73,33	100	164
	36	SVLN (38)	35	97,22	92,11	
	48	JMLR (45)	45	93,75	100	
	63	MLJ (72)	62	98,41	86,11	
353	65	Elsevier (52)	52	80	100	326
	65	SVLN (75)	64	98,46	85,33	
	105	JMLR (95)	95	90,48	100	
	118	MLJ (131)	115	97,46	87,79	
Instances	Runtime	Comparisons	Avg Runtime (sec)	Prec (%)	Rec (%)	Pur (%)
40	25'24"	780	1,95	83,33	91,33	87,5
177	9h 34' 45"	15576	2,21	90,68	94,56	92,66
353	39h 12' 07"	62128	2,27	91,60	93,28	92,35

considering the descriptions complexity and the fact that the prototype has no optimization in this preliminary version. Also the semantic perspective is quite satisfactory: an insight of the clustering outcomes shows that errors are made on very ambiguous documents (the four classes have a very similar layout style), while the induced cluster descriptions highlight interesting and characterizing layout clues. Preliminary comparisons on the 177 dataset with other classical measures report an improvement with respect to both Jaccard’s, Tverski’s and Dice’s measures up to +5,48% for precision, up to + 8,05% for recall and up to + 2,83% for purity.

7 Conclusions

Knowledge extraction concerns the ability to identify valid, potentially useful and understandable patterns from large data collections. Such a task becomes more difficult if the domain of application requires a First-Order Logic representation language, due to the problem of indeterminacy in mapping portions of descriptions onto each other. Nevertheless, the ability to assess similarity between first-order descriptions has many applications, ranging from description selection to flexible matching, from instance-based learning to clustering.

This paper deals with Conceptual Clustering, and proposes a framework for Horn Clauses similarity assessment. Experimental results on real-world datasets prove that, endowing classical clustering algorithms with this framework, considerable effectiveness can be reached. Future work will concern fine-tuning of the similarity computation methodology, and a more extensive experimentation.

References

- [1] G. Bisson. Conceptual clustering in a first order logic representation. In *ECAI '92: Proceedings of the 10th European conference on Artificial intelligence*, pages 458–462. John Wiley & Sons, Inc., 1992.
- [2] H. Blockeel, L. De Raedt, and J. Ramon. Top-down induction of clustering trees. In J. Shavlik, editor, *Proceedings of the 15th International Conference on Machine Learning*, pages 55–63. Morgan Kaufmann, 1998.
- [3] W. Emde and D. Wettschereck. Relational instance based learning. In L. Saitta, editor, *Proc. of ICML-96*, pages 122–130, 1996.
- [4] F. Esposito, D. Malerba, and G. Semeraro. Classification in noisy environments using a distance measure between structural symbolic descriptions. *IEEE Transactions on PAMI*, 14(3):390–402, 1992.
- [5] Floriana Esposito, Nicola Fanizzi, Stefano Ferilli, and Giovanni Semeraro. A generalization model based on oi-implication for ideal theory refinement. *Fundam. Inform.*, 47(1-2):15–33, 2001.
- [6] S. Ferilli, T.M.A. Basile, N. Di Mauro, M. Biba, and F. Esposito. Similarity-guided clause generalization. In *Proc. of AI*IA-2007*, LNAI, page 12. Springer, 2007 (To appear).
- [7] D. H. Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2(2):139–172, 1987.
- [8] J. H. Gennari, P. Langley, and D. Fisher. Models of incremental concept formation. *Artificial Intelligence*, 40(1-3):11–61, 1989.
- [9] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [10] Dekang Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.
- [11] R. S. Michalski and R. E. Stepp. Learning from observation: Conceptual clustering. In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 331–363. Springer: Berlin, 1984.
- [12] S. Nienhuys-Cheng. Distances and limits on herbrand interpretations. In D. Page, editor, *Proc. of ILP-98*, volume 1446 of *LNAI*, pages 250–260. Springer, 1998.
- [13] G. D. Plotkin. A note on inductive generalization. *Machine Intelligence*, 5:153–163, 1970.
- [14] J. Ramon. *Clustering and instance based learning in first order logic*. PhD thesis, Dept. of Computer Science, K.U.Leuven, Belgium, 2002.
- [15] J. Ramon and L. Dehaspe. Upgrading bayesian clustering to first order logic. In *Proceedings of the 9th Belgian-Dutch Conference on Machine Learning*, pages 77–84. Department of Computer Science, K.U.Leuven, 1999.
- [16] C. Rouveirol. Extensions of inversion of resolution applied to theory completion. In *Inductive Logic Programming*, pages 64–90. Academic Press, 1992.
- [17] M. Sebag. Distance induction in first order logic. In N. Lavrač and S. Džeroski, editors, *Proc. of ILP-97*, volume 1297 of *LNAI*, pages 264–272. Springer, 1997.
- [18] K. Thompson and P. Langley. Incremental concept formation with composite objects. In *Proceedings of the sixth international workshop on Machine learning*, pages 371–374. Morgan Kaufmann Publishers Inc., 1989.

Finding Composite Episodes

Ronnie Bathoorn and Arno Siebes

Institute of Information & Computing Sciences
Utrecht University
P.O. Box 80.089, 3508TB Utrecht, The Netherlands
{ronnie, arno}@cs.uu.nl

Abstract. Mining frequent patterns is a major topic in data mining research, resulting in many seminal papers and algorithms on item set and episode discovery. The combination of these, called composite episodes, has attracted far less attention in literature, however. The main reason is that the well-known frequent pattern explosion is far worse for composite episodes than it is for item sets or episodes. Yet, there are many applications where composite episodes are required, e.g., in developmental biology where sequences containing gene activity sets over time are analyzed.

This paper introduces an effective algorithm for the discovery of a small, descriptive set of composite episodes. It builds on our earlier work employing MDL for finding such sets for item sets and episodes. This combination yields an optimization problem. For the best results the components descriptive power has to be balanced. Again, this problem is solved using MDL.

keywords: composite episodes, MDL

1 Introduction

Frequent pattern mining is a major area in data mining research. Many seminal papers and algorithms have been written on the discovery of patterns such as item sets and episodes. However the combination of these two, called composite episodes, has attracted far less attention in the literature. Such composite episodes [1] are episodes of the form

$$\{A, B\} \rightarrow \{C, D\} \rightarrow \{E\}.$$

There are applications where one would like to discover frequent composite episodes. In developmental biology one has data sets that consist of time series where at each time point sets of events are registered. In one type of data, these events are the active genes at that moment in the development. In another type of data, the events are the morphological characters that occur for the first time at that moment in the development. For both types of data, frequent composite episodes would yield important insight in the development of species.

The main reason why there has been little attention to the discovery of composite episodes in the literature is that the frequent pattern explosion is worse for composite episodes than it is for both frequent item sets and for frequent episodes. In other words, the number of frequent patterns quickly explodes. For example, if $\{A, B\} \rightarrow \{C, D\}$ is frequent, then so are $\{A\}$, $\{A\} \rightarrow \{C\}$, $\{A\} \rightarrow \{D\}$, $\{A\} \rightarrow \{C, D\}$, $\{B\}$,

$\{B\} \rightarrow \{C\}$, $\{B\} \rightarrow \{D\}$, $\{B\} \rightarrow \{C, D\}$, $\{A, B\} \rightarrow \{C\}$, and $\{A, B\} \rightarrow \{D\}$. So, clearly an A Priori like property holds, but the number of results will simply swamp the user.

In related work [2] so called Follow-Correlation Item set-Pairs are extracted. These are patterns of the form $\langle A^m, B^n \rangle$ meaning: B likely occurs n times after A occurs m times. Patterns of this form only describe the interaction between two subsequent item sets and their complexity lies somewhere between item sets and composite episodes. And unlike our method this method does not offer a solution to restrict the number of patterns generated for low minimal support values. Other related work, item set summarization [3], does offer a method to restrict the number of item sets. However, there is no straightforward generalization to composite episodes.

In earlier work [4] we showed that MDL can be used to select interesting patterns that give a good description of the database. This paper extends on our earlier work on the use of MDL to select interesting item sets and episodes, we propose a method that reduces the number of generated patterns before it starts combining item sets into composite episodes. This reduces the number of generated patterns dramatically, while still discovering the important patterns.

Briefly the method works as follows: using a reduced set of item sets as building blocks for the patterns in the time sequences, we limit the number of possible patterns for a given dataset. MDL is used to select the item sets extracted from the data that contribute the most to the compression of that data as shown in [5]. Then the reduced set of patterns is used to encode the database after which episodes are extracted from this encoded database. Finally MDL is used again to reduce this set of episodes [6].

Simply running these two stages independently after each other, however, doesn't necessarily produce the best results. A too selective first stage will limit our abilities to select good composite sequences in the second stage. The selectivity of both stages has to be balanced for optimal results. We again use MDL to achieve this balance.

The rest of this papers is structured as follows. In Section 2 we will give some definitions of the concepts used in the rest of this paper. Section 3 introduces our 2-Fold MDL compression method used to extract composite episodes. Section 4 introduces the dataset we used in the experiments. The experiments and their results are discussed in Section 5. Section 6 contains our conclusions.

2 Composite Episode Patterns

Episodes are partially ordered sets of events that occur frequently in a time sequence. Mannila described 3 kind of episodes in [1], parallel episodes which can be seen as item sets, serial episodes and composite episodes as shown in figure 1.

The data and the composite episodes can be formalized as follows: For item sets x_i and x_j in a sequence x , we will use the notation $x_i \preceq x_j$ to denote that x_i occurs before x_j in x .

Definition 1. Given a finite set of events \mathcal{I} ,

1. A sequence s over \mathcal{I} is an ordered set of item sets

$$s = \{(is_i, i)\}_{i \in \{1, \dots, n\}},$$

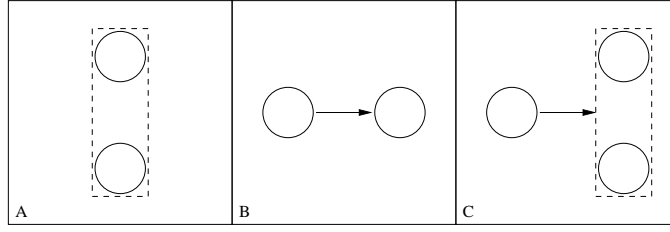


Fig. 1. (a) Item set, (b) episode, (c) composite episode.

in which the $is_i \subseteq \mathcal{I}$. If $1 \leq i \leq j \leq n$,
then $(is_i, i) \preceq (is_j, j)$.

2. An item set is is a set of events that happen together:

$$is = (e_1, \dots, e_j)$$

where j is the size of the item set.

3. A composite episode ep is a sequence of item sets.

$$ep = (is_1, \dots, is_k)$$

$$is_i = (e_1, \dots, e_{i_l})$$

where i_l is the size of the i^{th} item set.

The database db consists of a set of sequences of item sets, i.e., it consists of composite episodes. In this database, we want to find the frequent composite episodes. To define these, we need the notion of an occurrence of a composite episode. Note that, because of our application, we do not allow gaps in occurrences.

Definition 2. 1. Let x be composite episodes and y be a sequence. Let I be the set of composite episodes and $\Phi : I \rightarrow I$ an injective mapping. x occurs in y , denoted by $x \subseteq y$, iff

$$(a) \forall x_i \in x : x_i \subseteq \Phi(x_i)$$

$$(b) \forall x_i, x_j \in x:$$

$$i. \Phi(x_i) \preceq \Phi(x_j) \Leftrightarrow x_i \preceq x_j$$

$$ii. \exists y_k \in y : \Phi(x_i) \preceq y_k \preceq \Phi(x_j) \Leftrightarrow$$

$$\exists x_k \in x : \Phi(x_k) = y_k \wedge x_i \preceq x_k \preceq x_j.$$

The mapping Φ is called the occurrence.

2. length of an occurrence o is time interval between the first (t_s) and the last (t_e) event in the occurrence

$$length(o) = t_e(o) - t_s(o)$$

3. support of an episode is the number of occurrences of the episode in the database.

So, $(\{A, B\}, \{C\})$ occurs once in $\{A, B, C\}, \{C, D\}$, while $(\{A\}, \{B\})$ doesn't.

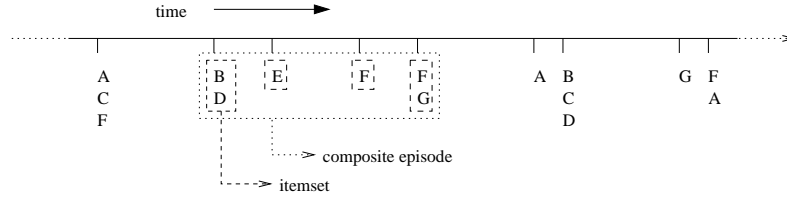


Fig. 2. Example sequence with a composite episode occurrence highlighted.

2.1 Pattern Explosion

As noted before, the number of composite episodes quickly explodes. In a composite episode of length 10 containing 3 possible events we have 7 possible item sets. This leads to $\sum_{i=1}^{10} 7^i = 329554456$ possible composite episodes. More in general, with n events and a sequence of length k we have as number of possible composite episodes:

$$\sum_{i=1}^k (2^n - 1)^i$$

Clearly, if the number of frequent item sets is not the maximum, then the number of possibly frequent composite episodes also goes down.

While the growth remains exponential, the fewer (frequent) item sets we consider, the fewer composite episodes we have to consider. This is exactly the power of our approach: by dramatically reducing the number of item sets to consider, the number of of composite episodes to consider becomes manageable.

2.2 Item set MDL

The basic building blocks of our database are the items \mathcal{I} , e.g., the items for sale in a shop. A transaction $t \in \mathcal{P}(\mathcal{I})$ is a set of items, e.g. representing the items a client bought at that store. A database db over \mathcal{I} is a bag of transactions, e.g., the different sale transactions on a given day. An item set $I \in \mathcal{I}$ occurs in a transaction $t \in db$ iff $I \subseteq t$. The support of I in db is the number of transactions in the database in which I occurs.

We will now give a quick summary on how MDL can be used to select a small and descriptive set of item sets, using the Krimp algorithm which was introduced in [5]. This is a shortened version of the description given in [4].

The key idea of our compression based approach is the code table, a code table has item sets on the left-hand side and a code for each item set on its right-hand side. The item sets in the code table are ordered descending on 1) item set length and 2) support. The actual codes on the right-hand side are of no importance: their lengths are. To explain how these lengths are computed we first have to introduce the coding algorithm. A transaction t is encoded by Krimp by searching for the first item set c in the code table for which $c \subseteq t$. The code for c becomes part of the encoding of t . If $t \setminus c \neq \emptyset$, the algorithm continues to encode $t \setminus c$. Since we insist that each code table contains at least all singleton item sets, this algorithm gives a unique encoding to each

(possible) transaction. The set of item sets used to encode a transaction is called its cover. Note that the coding algorithm implies that a cover consists of non-overlapping item sets. The length of the code of an item in a code table CT_i depends on the database we want to compress; the more often a code is used, the shorter it should be. To compute this code length, we encode each transaction in the database db . The frequency of an item set $c \in CT$ is the number of transactions $t \in db$ which have c in their cover. The relative frequency of $c \in CT_i$ is the probability that c is used to encode an arbitrary $t \in db$. For optimal compression of db , the higher $P(c)$, the shorter its code should be. In fact, from information theory [7] we have the optimal code length for c as:

$$l_{CT_i}(c) = -\log(P(c|db)) = -\log\left(\frac{freq(c)}{\sum_{d \in CT_i} freq(d)}\right) \quad (1)$$

The length of the encoding of a transaction is now simply the sum of the code lengths of the item sets in its cover. Therefore the encoded size of a transaction $t \in db$ compressed using a specified code table CT_i is calculated as follows:

$$L_{CT_i}(t) = \sum_{c \in cover(t, CT_i)} l_{CT_i}(c) \quad (2)$$

The size of the encoded database is the sum of the sizes of the encoded transactions, but can also be computed from the frequencies of each of the elements in the code table:

$$L_{CT_i}(db) = \sum_{t \in db} L_{CT_i}(t) = - \sum_{c \in CT_i} freq(c) \cdot \log\left(\frac{freq(c)}{\sum_{d \in CT_i} freq(d)}\right) \quad (3)$$

Finding the Right Code Table To find the optimal code table using MDL, we need to take into account both the compressed database size as described above as well as the size of the code table. (Otherwise, the code table could grow without limits and become even larger than the original database!) For the size of the code table, we only count those item sets that have a non-zero frequency. The size of the right-hand side column is obvious; it is simply the sum of all the different code lengths. For the size of the left-hand side column, note that the simplest valid code table consists only of the singleton item sets. This is the *standard encoding* (st) which we use to compute the size of the item sets in the left-hand side column. Hence, the size of the code table is given by:

$$L(CT) = \sum_{c \in CT: freq(c) \neq 0} l_{st}(c) + l_{CT}(c) \quad (4)$$

In [5] we defined the optimal set of (frequent) item sets as that one whose associated code table minimizes the total compressed size:

$$L(CT) + L_{CT}(db) \quad (5)$$

The algorithm starts with a valid code table (generally only the collection of singletons) and a sorted list of candidates. These candidates are assumed to be sorted

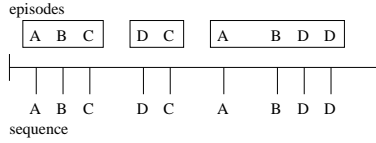


Fig. 3. Example of a sequence cover.

descending on 1) support and 2) item set length. Each candidate item set is considered by inserting it at the right position in CT and calculating the new total compressed size. A candidate is only kept in the code table iff the resulting total size is smaller than it was before adding the candidate. For more details, please see [5].

2.3 Episode MDL

For episode mining the basic building blocks of our database db are item sets I , e.g. all the genes active at one point in time. Each transaction $t \in db$ is a sequence of item sets, e.g. a time sequence recording the activity of genes over time. An episode e occurs in a transaction t if all the item sets in e occur in t without gaps between them as described in Definition 2. Reducing a set of episodes using MDL follows the same steps as used in item set MDL. Thus we start with a codetable with two columns, it has an episode on it's left-hand side and a code for each episode on the right-hand side. The code table is used to cover all sequences in the database, an example of such a cover can be seen in Figure 3. The frequency with which the codes in the code table are used to cover all the sequences in the database determines their code size, the more a code is used the shorter its code. It is important to note that because of our application in developmental biology we do not allow overlap between the episodes in a cover, or gaps within the episodes. To determine the size of our episode code table we need to define a *standard encoding* for episodes l_{st_e} as well. As the length of an episode in the codetable we use the length of that episode as it would be when we encoded it using only episodes of length 1, this is called this episodes *standard encoding*. With this standard encoding the size of our episode code table CT_e becomes:

$$L(CT_e) = \sum_{c \in CT_e: freq(c) \neq 0} l_{st_e}(c) + l_{CT_e}(c) \quad (6)$$

Using the episodes in our code table to encode our database leads to the following database size:

$$L_{CT_e}(db) = \sum_{t \in db} L_{CT_e}(t) = - \sum_{c \in CT_e} freq(c) \cdot \log \left(\frac{freq(c)}{\sum_{d \in CT_e} freq(d)} \right) \quad (7)$$

More details on reducing frequent episode sets using MDL can be found in [6].

2.4 Combining item set and episode MDL

In our method for finding composite episodes we are combining item set and episode MDL. First we use a set of item sets to compress our database. Then we extract episodes from the encoded database that results from the item set compression. Using MDL we select the episodes that give a good compression of our *item set encoded* database. To determine which item sets and episodes are used in the compression we have to optimize L_{total} .

$$L_{total} = L(CT_i) + L(CT_e) + L_{CT_e}(enc(CT_i, db)) \quad (8)$$

where $enc(CT_i, db)$ is the database as encoded by item set code table CT_i . This last equation shows that to compute the total size we now need the item set code table CT_i as well as the episode code table CT_e plus the double encoded database.

3 2-Fold MDL compression

The basis of the algorithm used to find the composite episode patterns consists of 4 steps.

BASE(*data*, *min_sup*, *max_length*)

- 1 Find item sets for given *min_sup*
- 2 Compress the database using item set MDL
- 3 Find episodes in the encoded database with given *max_length*
- 4 Compress the database using episode MDL
- 5 **return** *composite episodes*

Fig. 4. Base algorithm.

In the first step we use a standard FP-growth algorithm [8] to find all the item sets for a given minimal support. Which minimal support to use is the subject of the next subsection.

The set of item sets is used in the second step where MDL is used to heuristically select those item sets that give the shortest description of the database. This results in a compressed database together with a code table used to obtain the compressed database as described in Section 2.2. This code table is a set of item sets selected from all frequent item sets and the encoded database is a copy of the original database in which all occurrences of code table elements are replaced by a code that represents this item set.

In the third step we get all frequent episodes from the encoded database that was generated in the previous step. The frequent episodes are extracted using a minimal occurrence episode discovery algorithm from [1]. Note that because we extract our

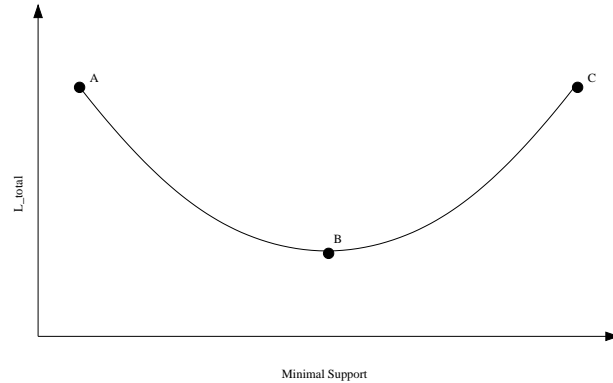


Fig. 5. Compressed Database size against minimal support values

episodes from the encoded database each event in the discovered episodes could now also be an item set, thus the episodes extracted are composite episodes.

And finally we use our method from Section 2.3 to get a set of episodes that give a good description of the database.

The output of our method consists of 2 codetables one from step 2 and one from step 4 together with the compressed database from step 4. Our MDL method enforces a loss-less compression of the database thus the 2 code tables can be used to decompress the database and generate the original dataset.

3.1 Compression Optimization

What is the right minimal support to use for extracting the item sets from the data? The minimal support limits the amount of episodes that could possibly be found. This can be seen as an optimization problem where we take L_{total} (equation 8) as the value to be optimized. We are interested in finding the minimal support that results in the lowest possible value of L_{total} .

Figure 5 shows the overall compression of the database for different minimal support levels. On the x-axis we have the minimal support used in the extraction of the item sets. Changes in the compression are caused by the interaction of item sets and episodes used in the compression.

At point 'A' in the graph the minimal support is set to 1, which means that all possible item sets will be extracted from the database. As the item sets are extracted before the episodes this means there is a strong bias towards the use of large item sets. Additionally the reduced set of item sets generated with the use of MDL is used in the encoding of our database before we proceed with the extraction of episodes. This will lower the probability of finding long episodes as they have to be build up of large item sets. As all item sets used in the encoding of our database are substituted by a single code this makes it impossible to use subsets of these item sets.

Increasing the minimal support will lower the number and size of the found item sets and will increase the possibility of longer episodes being used in the compression.

After reaching a certain threshold no item sets are used anymore and the compression will be based solely on episodes. This point is reached at 'C'.

Because of this interaction between item sets and episodes, we expect the best compression of our database somewhere near point 'B' in the graph. This changes our problem of finding the best minimal support for our algorithm to an optimization problem where we optimize the compression of the database by varying the minimal support. Our base algorithm can be extended by putting it inside a loop that runs the method for all the minimal support values we are interested in. So now the entire algorithm becomes as can be seen in Figure 6.

```
2-FOLD(data, max_length, start, end)
1  best_result = BASE(data, start, max_length)
2  foreach minsup in [start + 1..end]
3    cur_result = BASE(data, minsup, max_length)
4    if (cur_result.mld_size < best_result.mld_size) then
5      best_result = cur_result
6  return best_result
```

Fig. 6. Complete algorithm.

Computing this optimal solution comes at the prize of having to do one run of the composite episode extraction for each minimal support value. But as these runs do not depend on each other they can be run on different processors or different computers all together. Making this algorithm well suited for parallel computation and cutting the runtime down to the runtime of one run of the composite episode extraction algorithm.

4 The data

For our experiments we use two datasets from the biological domain. The first dataset contains time sequences containing developmental information of 5 different species. In these time sequences the timing of the activity of 28 different genes are recorded. There are large differences in the time sequences in length as well as in the number of times the events are present in each. More background information on the biology involved in the analysis of developmental sequences can be found in [9]. The second dataset contains time sequences of 24 mammals. It records the start of 116 different morphological characters such as the forming of the optic lens.

The datasets currently produced by the biologists are so small that the codetable is very large in relation to the database hampering the compression. As the biologists are working on producing bigger datasets in the future, we used the following method to test our method on a bigger datasets. The gene activity dataset was enlarged by combining

dataset	#sequences	#events
gene activity	5 species	28 genes
morphological characters	24 species	116 characters

Table 1. dataset description

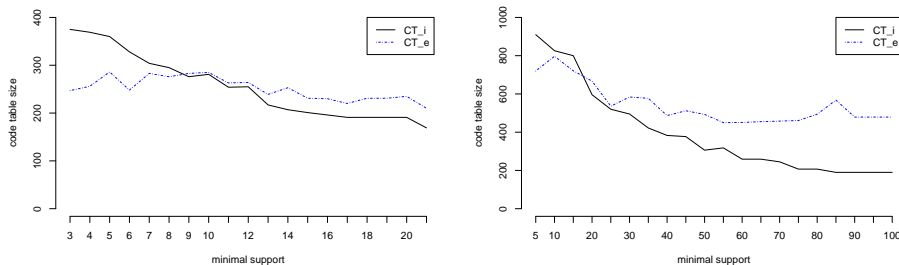


Fig. 7. Item set code table (ct_i) and episode code table (ct_e) sizes for the original (left) and the artificial (right) dataset.

time sequences of two randomly chosen species in a new time sequence by adding a reversed copy of the second time sequence to the back of the first. This recombination is used to preserve the types of patterns that can be found in the data but increases the number of patterns found as well as their frequency. The artificial dataset contains 10 of these combined time sequences, doubling the number of time sequences as well as the average length of these sequences.

5 Experimental Results

The first experiment was done on the original gene activity dataset and composite episodes were extracted for minimal support values ranging from 3 to 21. Figure 7 shows the size of the code tables for the different minimal support values. Here we can see that for lower minimal support levels the item set codetable is bigger than the episode code table and this is the other way around when the minimal support is increased.

In our experiments on the morphological characters dataset the algorithm was run multiple times for 7 different minimal support values ranging from 2 to 8. Where the minimal support is the minimal support for the item set extraction. The item sets are extracted using the implementation of fp-growth taken from [10]. The episodes were extracted using 3 different maximal episode lengths, 25, 35 and 45. It is important to note that the end result of our method is a set of composite episodes, the compression values in this experiment are only used to select the best minimal support for the item set discovery.

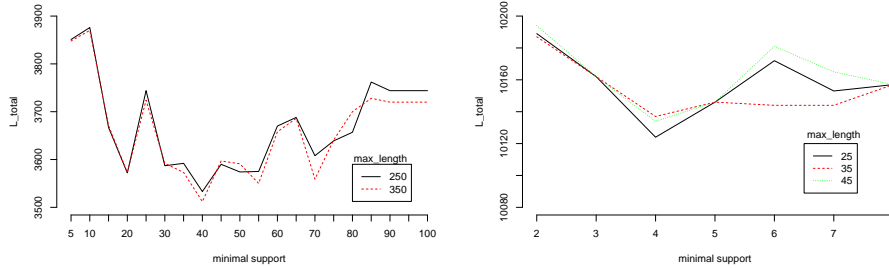


Fig. 8. Compressed Database size against minimal support values for the gene activity dataset (left) and the morphological dataset (right).

Figure 8 (left) shows the compressed database size as a function of the minimal support of the item set extraction for two maximal episode lengths. The compressed database size shown in the figure is the sum of the item set codetable, the episode codetable and the size of the encoded database. The figure looks very similar to figure 5 which was what we predicted based on the interaction of the item sets and the episodes. In figure 8 we can see that we reach the best compression for a minimal support of 4. The same experiment was done on the artificially enlarged dataset. With item sets being extracted for 20 different minimal support values ranging from 5 to 100. The episodes are extracted using 2 different maximal episode lengths, 250 and 350. The results are also shown in Figure 8 (right) we can see that we reach the best compression for a minimal support of 40.

To give an indication of the amount of reduction reached in the total number of patterns generated, only between 0.002% and 6.667% of the frequent item sets were selected by MDL. The reduction decreased for higher minimal support. As we showed in Section 2.1 this gives a tremendous reduction in the possible composite episodes. For the frequent episodes only between 0.33% and 1,7% of the episodes were selected by MDL as being interesting. Here the reduction was better for higher minimal support, due to the interaction between the item sets and the episodes in the total compression.

For the original dataset 2-Fold started with a set of 58 item sets from which it constructed 18 composite episodes. An example of such a composite episode: $\{hoxc10\} \rightarrow \{hoxd11, hoxd12\}$ which describes the temporal collinearity of hox genes that Biologists already know from their experiments.

6 Conclusions & Future Work

In this paper we show that it is possible to mine for the descriptive composite episodes from data. The 2-Fold algorithm uses MDL to keep the combinatorial explosion of potential patterns under control. 2-Fold uses MDL in three different ways. Firstly to mine for descriptive item sets. Secondly to mine for descriptive episodes. Thirdly to

balance the first two, to ensure the discovery of descriptive composite episodes. The experiments show first of all that MDL performs well in all three of its tasks. The number of composite episodes discovered is small enough that experts can still verify them. Moreover, the validity of the results we discovered has been verified by domain experts.

Acknowledgment

We would like to thank Matthijs van Leeuwen & Jilles Vreeken for their short introduction on item set MDL.

References

1. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery* **1** (1997) 259–289
2. Zhang, S., Zhang, J., Zhu, X., Huang, Z.: Identifying follow-correlation itemset-pairs. In: *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, Washington, DC, USA, IEEE Computer Society (2006) 765–774
3. Wang, C., Parthasarathy, S.: Summarizing itemset patterns using probabilistic models. In: *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, ACM Press (2006) 730–735
4. van Leeuwen, M., Vreeken, J., Siebes, A.: Compression picks item sets that matter. In Fürnkranz, J., Scheffer, T., Spiliopoulou, M., eds.: *PKDD*. Volume 4213 of *Lecture Notes in Computer Science*, Springer (2006) 585–592
5. Siebes, A., Vreeken, J., van Leeuwen, M.: Itemsets that compress. In: *SIAM 2006: Proceedings of the SIAM Conference on Data Mining*, Maryland, USA (2006) 393–404
6. Bathoorn, R., Koopman, A., Siebes, A.: Reducing the frequent pattern set. In Tsumoto, S., Clifton, C., Zhong, N., Wu, X., Liu, J., Wah, B., Cheung, Y.M., eds.: *ICDM '06: Proceedings of the 6th International Conference on Data Mining - Workshops*. Volume 6 of *ICDM workshops*, Los Alamitos, CA, USA, IEEE Computer Society (2006) 55–59
7. Grünwald, P. In: *Advances in Minimum Description Length. A tutorial introduction to the minimum description length principle*. MIT Press (2005)
8. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In Chen, W., Naughton, J., Bernstein, P.A., eds.: *2000 ACM SIGMOD Intl. Conference on Management of Data*, ACM Press (2000) 1–12
9. Welten, M.C.M., Verbeek, F.J., Meijer, A.H., Richardson, M.K.: Gene expression and digit homology in the chicken embryo wing. *Evolution & Development* **7** (2005) 18–28
10. Rácz, B., Bodon, F., Schmidt-Thieme, L.: On benchmarking frequent itemset mining algorithms. In: *Proceedings of the 1st International Workshop on Open Source Data Mining*, in conjunction with ACM SIGKDD. (2005)

Using Secondary Knowledge to Support Decision Tree Classification of Retrospective Clinical Data[★]

Dympna O’Sullivan¹, William Elazmeh², Szymon Wilk¹, Ken Farion³, Stan
Matwin^{1,3}, Wojtek Michalowski¹, and Morvarid Sehatkar¹

¹ University of Ottawa, Ottawa, Canada

`dympna,wilk,wojtek@telfer.uottawa.ca, mseha092@site.uottawa.ca`

² Institute of Computer Science, Polish Academy of Sciences, Warsaw, Poland
`stan@site.uottawa.ca`

³ University of Bristol, Bristol, United Kingdom

`elazmah@cs.bris.ac.uk`

⁴ Faculty of Medicine, University of Ottawa, Ottawa, Canada

`farion@cheo.on.ca`

Abstract. Retrospective clinical data presents many challenges for data mining and machine learning. The transcription of patient records from paper charts and subsequent manipulation of data often results in high volumes of noise as well as a loss of other important information. In addition, such datasets often fail to represent expert medical knowledge and reasoning in any explicit manner. In this research we describe applying data mining methods to retrospective clinical data to build a prediction model for asthma exacerbation severity for pediatric patients in the emergency department. Difficulties in building such a model forced us to investigate alternative strategies for analyzing and processing a retrospective data. This paper describes this process together with an approach to mining retrospective clinical data by incorporating formalized external expert knowledge (*secondary knowledge sources*) into the classification task. This knowledge is used to partition the data into a number of coherent sets, where each set is explicitly described in terms of the secondary knowledge source. Instances from each set are then classified in a manner appropriate for the characteristics of the particular set. We present our methodology and outline a set of experiential results that demonstrate some advantages and some limitations of our approach.

1 Introduction

In his book [1], Motulsky submits “*the human brain excels at finding patterns and relationships ...*”. Scientists have long exhibited an aptitude to learn and

[★] The support of the Natural Sciences and Engineering Research Council of Canada, the Canadian Institutes of Health Research and the Ontario Centres of Excellence is gratefully acknowledged.

generalize from observations leading them to develop refined methods for detecting patterns and identifying coherent conjectures drawn from experience. Since their early days, intelligent computer systems have inspired scientists with their promising potential of supporting such research in medical domains [2]. However, medical data features many difficult domain-specific characteristics and complex properties [3]. Incompleteness (missing data), incorrectness (noise), sparseness (non-representative values), and inexactness (inappropriate parameter selection) make up the short list of challenges faced by any machine learning technique applied in the medical domain [4]. A comprehensive overview of these and other challenges is presented in [5], where medical data is described as often being heterogeneous in source as well as in structure, and that the pervasiveness of missing values for technical and/or social reasons can create problems for automatic methods for classification and prediction. Furthermore, translating physicians' interpretations based on years of clinical experience to formal models represents a serious and complex challenge.

An important requirement of medical problem solving or decision support applications is interpretability for domain users [6]. Such a stipulation dramatically reduces the choice of machine learning models that can be applied to medical problem solving to those that can offer systematic justification and explanation of the prediction process. Such models include classifiers that estimate probabilities (probabilistic), classifiers that identify training examples similar to a test example (case-based), classifiers that produce rules that can be applied to a given test example (rule-based), and classifiers that describe decisions based on a selected set of attributes (tree-based). In this work we have chosen to focus our prediction efforts on tree-based classifiers. Decision tree classification models are especially useful in medical applications as a result of their simple interpretation but also as they are represented in the form typically used for describing clinical algorithms and practice guidelines. As such a tree-based classification model can easily be represented in a comprehensible and transparent format if required, without the need for computer implementation.

In this work, the clinical prediction task is centered on the domain of emergency pediatric asthma where the goal is to develop a classification model that can provide an early prediction of the severity of a child's asthma exacerbation. Asthma is the most common chronic disease in children (10% of Canadian population), and asthma exacerbations are one of the most common reasons for children to be brought to the emergency department [7]. The provision of computer-based decision support to emergency physicians treating asthma patients has been shown to increase the overall effectiveness of health care delivered in emergency departments [8, 9]. For a patient suffering from an asthma exacerbation, early identification of severity (*mild*, *moderate*, or *severe*) is a crucial part of the management and treatment process. Patients with a *mild* attack are usually discharged following a brief course of treatment (less than 4 hours) and resolution of symptoms, patients with a *moderate* attack receive more aggressive treatment over an extended observation in the emergency department (up to 12 hours), and patients with a *severe* attack receive maximal therapy before

ultimately being transferred to an in-patient hospital bed for ongoing treatment (after about 16 hours in the emergency department).

This paper discusses challenges, issues, and difficulties we face in developing a prediction model for early asthma exacerbation severity using a retrospective clinical data. Preliminary analysis of the data without preprocessing resulted in unacceptably low classification accuracy. These results forced a rethink of common methodologies for mining retrospective clinical data. Although not particularly complex, this data set is characterized by a fair amount of missing values such that standard methods of feature extraction and classifier tuning fail to produce acceptable performance. Furthermore, clinically-based “classifiers”, such as PRAM (section 3.1) cannot be applied due to the type of data being collected. We employ such a clinical classifier as an external method to evaluate the data which leads to the identification of data where PRAM can be used and “other”. We argue that such partitioning will ultimately improve the classification. Our investigations led us to develop a methodology for classification that involves identification and formalization of expert medical knowledge specific to the clinical domain. This knowledge is referred to as a secondary knowledge source and its incorporation allowed us to exploit implicit domain knowledge in the data for more fine-grained data analysis and processing. This paper demonstrates the usefulness of how to exploit this secondary knowledge to partition medical data to reduce its complexity. Our experimental evaluation demonstrates that with such partitioning a decision tree classifier is capable of overcoming some but not all complexities posed by this dataset. An added benefit is the ability to capture different regularities that should be in asthma data according to PRAM, thus in a sense, we “extend” its interpretation.

This paper is organized as follows. In Section 2 we describe the retrospectively collected asthma data used in this analysis. Section 3 outlines a methodology for identifying, formalizing and applying secondary knowledge sources with the purpose of harnessing and exploiting implicit domain knowledge. An experimental evaluation of this approach is outlined in Section 4, where our results display that the approach can be applied with some degree of success with some limitations. We conclude with a discussion in Section 5.

2 Retrospective Clinical Dataset

The dataset used in this study was developed as part of a retrospective chart study conducted in 2004 at the Children’s Hospital of Eastern Ontario (CHEO), Ottawa, Canada. The study includes patients who visited the hospital’s emergency department from 2001 to 2003 for treatment of an asthma exacerbation. To illustrate the underlying structure of the data, we present the workflow by which asthma patients are processed in the emergency department (Figure 1). The workflow shows that a patient is evaluated multiple times by multiple caregivers at variable time intervals. This information is documented on the patient chart with varying degrees of completeness. Furthermore, some aspects of evaluation are objective and therefore reliable measures of the patient’s status, however

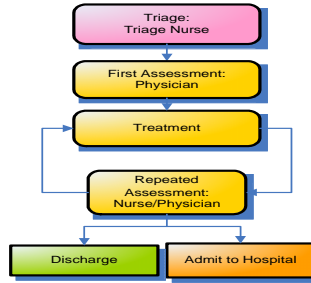


Fig. 1. Asthma Assessment Workflow in the Emergency Department at CHEO

other aspects can be quite subjective and less reliably correlated with the patient’s status. In preparing the final dataset, patient information was divided into three subcategories for each record; historical and environmental information, information collected during the triage assessment and information collected at a reassessment approximately 2 hours after triage. The final dataset consisted of 362 records and each record was reviewed by a physician and assigned to one of two classes (*mild* or *moderate/severe*) using predefined criteria related to the duration and extent of treatment required, the final disposition (i.e., discharged or admitted to hospital), and the possible need for additional visits for ongoing symptoms. In this way, the assigned severity group was used as a gold standard for creation and evaluation of a prediction model.

The dynamic nature of asthma exacerbations and the collection of assessments over time would lend itself naturally to a temporal representation for analysis of data. However, inconsistencies in data recording meant it was not possible to incorporate a temporal aspect into the analysis. Further difficulties presented by the data were a significant number of missing values (for some attributes up to 98%), incorrectness, sparseness, and noise due to the variability with which information was recorded, and inexactness due to inappropriate parameter selections as well as the problem of “values as attributes” often encountered in medical data.

3 Secondary Knowledge Sources

Evidence-based medicine is a recent movement that has gained prominence in current clinical practice as a methodology for supporting clinical decision making. The practice of evidence based medicine involves integrating individual clinical expertise with the best available external clinical evidence from systematic research [10]. Individual clinical expertise refers to the proficiency and judgment that individual clinicians acquire through clinical experience and external clinical evidence describes clinically relevant research usually evaluated using randomized control trials. In practice evidence based medicine is applied in a number of

ways, including, through the use of clinical practice guidelines, specialty-specific literature and clinical scoring systems.

In this research, we utilize external clinical evidence to support the classification task. The incorporation of a secondary knowledge source into classification leads us to define a three step approach to mining retrospective clinical data. In the first step relevant medical evidence is identified, for example in the form of a clinical practice guideline for the particular clinical domain. The second step is to formalize the medical evidence so it can be applied to available data. The third step involves developing a framework that makes use of the evidence to support the automatic classification task. The advantage of integrating such knowledge is that it allows for more effective and natural organization of information along existing and important data characteristics. As such secondary knowledge can be viewed as a proxy for an expert built classifier and may be incorporated to improve the predictive accuracy of the automatic classification task.

3.1 Secondary Knowledge Sources for Pediatric Asthma

The secondary knowledge source identified as relevant for our retrospective asthma data is the Preschool Respiratory Assessment Measure (PRAM) asthma index [11]. PRAM provides a discriminative index of asthma severity for preschool children. It is based on five clinical attributes commonly recorded for pediatric asthma patients, *suprasternal indrawing*, *scalene retractions*, *wheezing*, *air entry* and *oxygen saturation*. PRAM is based on a 12 point scale (see Table 1) and is calculated using scores of 0, 1, 2, and 3. These scores are assigned to attributes depending on the presence or absence of values as well as observed increasing or decreasing values of attributes. PRAM has been clinically validated as a reliable and responsive measure of the severity of airway obstruction. A patient with a PRAM score of 4 or less is considered to have a *mild* exacerbation, a score between 5 and 8 corresponds to a *moderate* exacerbation, and a score of 9 or higher corresponds to a *severe* exacerbation.

In order to identify if the PRAM scoring system was appropriate secondary knowledge, the retrospective asthma dataset was analyzed for the presence of PRAM attributes. It was found that four of the five PRAM attributes were present in our data and values for these attributes may be collected twice for each record, once at triage and again at reassessment. The next step of our approach was to formalize the secondary knowledge source so that it could be applied to the classification task. This process is described in the next subsection.

3.2 Formalizing Secondary Knowledge Sources for Classification

The formalization of the secondary knowledge source involved determining a mapping from the set of attributes outlined by PRAM to a subset of attributes from the retrospective asthma data and an associated assignment of scores for attribute values. This was necessary as not all attributes required to calculate the PRAM score were present in the retrospective asthma data, and for some other attributes a 1:1 mapping did not exist. Specifically, the retrospective data

Table 1. PRAM Scoring System

Signs	0	1	2	3
Suprasternal indrawing	absent		present	
Scalene retractions	absent		present	
Wheezing	absent	expiratory	inspiratory and expiratory	Audible without stethoscope /absent with no air entry
Air entry	normal	decreased bases	widespread decrease	absent/minimal
Oxygen saturation	$\geq 95\%$	92-95%	$< 92\%$	

did not contain an attribute corresponding with “Suprasternal Indrawing”, and “Wheezing” was captured using two attributes in the retrospective data, inspiratory wheezing and expiratory wheezing. Also, the PRAM scoring system describes “Air Entry” using four values (normal, decreased bases, widespread decrease and absent/minimal), whereas our data defined air entry as either good (i.e., normal) or reduced. Therefore the formalized mapping was developed in conjunction with a domain expert (emergency physician), and the rules devised for mapping attributes used by the PRAM system to attributes in our data and their corresponding score assignments are shown in Table 2.

3.3 Building a Classifier by a Secondary Knowledge Source

The final step of our approach was to use secondary knowledge to build a model for predicting asthma severity. In the retrospective asthma data a decision (class label) is recorded for each patient along with clinical and historical information. This class is the final outcome for the patient as recorded in the patient chart (not the result of the assessment at the 2-hour point) and indicates whether the patient has suffered a *mild* or *moderate/severe* exacerbation. Using the attributes, values and associated scores mapped from the PRAM scoring system we calculated a PRAM score for each patient in the dataset. This score had possible values between 0 and 12, where a score of less than 5 indicated a *mild* exacerbation and a score of greater than 5 indicated a *moderate/severe* exacerbation. (In our data the *moderate* and *severe* categories outlined by PRAM are collapsed into one group, *moderate/severe*). The score is then compared with the class label for each record in the dataset and the set of patients who comply with the PRAM scoring system are identified. The assignment of PRAM scores allows for the dataset to be partitioned into instances for which all PRAM attributes were present and thus a complete and correct PRAM score could be calculated and instances for which only a partial or no PRAM score could be calculated due

Table 2. Mapping PRAM attributes and scores

Attribute(s)	Value(s)	Score
Oxygen Saturation	Greater than 95%	0
Oxygen Saturation	Greater than 92% and Less than 95%	1
Oxygen Saturation	Greater than 88% and Less than 92%	2
Oxygen Saturation	Less than 88%	3
Air Entry	Good	0
Air Entry (class = mild)	Reduced	1
Air Entry (class = other)	Reduced	3
Retractions AND Air Entry	Absent AND Good	0
Retractions AND Air Entry	Absent AND Reduced	1
Retractions AND Air Entry	Absent AND "Missing"	2
Retractions	Present	2
Expiratory AND Inspiratory Wheeze	Absent AND Absent	0
Expiratory AND Inspiratory Wheeze	Present AND Absent	1
Expiratory AND Inspiratory Wheeze	Present AND Present	2
Expiratory AND Inspiratory Wheeze	Absent AND Present	Undefined

to the absence of values for the PRAM attributes. PRAM attributes may be collected at two stages in the asthma workflow (triage and reassessment), however analysis of our data demonstrated that such attributes were more likely to be collected at reassessment (there were many missing values for triage attributes) and as such the dataset was partitioned using the larger set of reassessed values. This resulted in a dataset with 147 instances for which the PRAM score was complete and correct, 206 instances where only a partial or no PRAM score could be calculated due to missing values and 9 instances for which the score calculated by PRAM and the class label completely disagreed. These 9 cases were considered outliers and deleted from the dataset for evaluation.

4 Experimental Evaluation

4.1 Experimental Design

Our evaluation reports results from a number of experiments involving the retrospective asthma dataset where each experiment involved building a decision tree using the J48 decision tree classifier in Weka[12] to classify data. The first experiment involved building a classifier on the entire dataset prior to any application of secondary knowledge. These results serve as a baseline for classifier performance upon which to evaluate all subsequent results. In the next experiment secondary knowledge in the form of the PRAM scoring system was applied to partition the dataset into two sets, one containing PRAM complete and correct instances and one containing PRAM partial or incomplete instances. The purpose of this experiment is to demonstrate that the incorporation of secondary knowledge into the classification tasks allows for enhanced representation of data

which results in reducing the complexity of the retrospective clinical dataset for classification. In the final experiment we applied feature selection to the complete original data set twice, once using automatic feature selection (available in Weka), and a second time by manually selecting combinations of expertly selected attributes and removing the remaining attributes. The function of this experiment was to show that neither automatic or expert feature selection can identify and reduce complexities in the data as efficiently as a classifier that incorporates secondary or expert medical knowledge, selects important features and partitions data into sets of similar characteristics.

For each experiment we report classifier performance in terms of percentages of Sensitivity (Sens) and Specificity (Spec), Predictive Accuracy (Acc) and Area Under the Curve (AUC) on the positive class. Sensitivity (the true positive rate) measures how often the classifier finds a set of positive examples. For instance in this research we consider *moderate/severe* to be the critical/positive class, therefore the sensitivity of *moderate/severe* measures how often the classifier correctly identifies patients suffering *moderate/severe* asthma exacerbations. Specificity (1 - false positive rate) measures how often what the classifier finds, is indeed what it was looking for. Therefore the specificity of the positive class (*moderate/severe*) measures how often what the classifier predicts is indeed a patient with a *moderate/severe* asthma exacerbation. Analyzing the trade-off between sensitivity and specificity is common in medical domains and is analogous to Receiver Operating Characteristics (ROC) analysis [13, 14] used in machine learning [15].

In addition we report accuracy and AUC where accuracy is the rate at which the classifier classifies patients (in both classes) correctly while AUC represents the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance [16]. Hence AUC measures the classifier's ability to discriminate the positive class from the negative class. In our experiments, we aim to analyze decision tree performance by measuring its ability to discriminate each positive patient with a *moderate/severe* asthma exacerbation. For a given classifier and positive class, an ROC curve [15, 16] plots the true positive rate against the rate of false positives produced by the classifier on the test data. The points along the ROC curve are produced by varying the classification threshold from most positive classification to most negative classification and the AUC of a classifier is the area under the ROC curve [16]. For this reason as well as the relatively small size of the dataset we evaluate the classifier for each patient in the dataset using leave-one-out cross-validation.

4.2 Classifying the Entire Dataset

The first experiment involved building a decision tree on the original dataset of 362 instances. The results of this experiment are shown in the first row of Table 3 and demonstrate that the retrospective clinical dataset is complex and that good classification accuracy is difficult to achieve without performing some degree of data preprocessing. We include these results as a baseline by which to measure subsequent classifier performance.

4.3 Classifying PRAM and non-PRAM Sets

In this experiment the dataset was partitioned by applying the formalized mapping from PRAM scoring system to attributes from the retrospective asthma dataset. This resulted in the dataset being partitioned into those that were PRAM complete and correct (PRAM set) and those that were either PRAM partial or incomplete (non-PRAM set). A decision tree was built for each set and the results are shown in the last two rows of Table 3. Also included for reference purposes are the results for the entire dataset in the first row.

Table 3. Decision Trees built on PRAM and non-PRAM sets

Set	Size	Sens	Spec	Acc	AUC
Entire	362	73	63	69	69
PRAM Set	147	93	96	95	98
Non-PRAM Set	206	89	53	74	77

From Table 3 we observe that splitting the dataset into different sets based on formalized secondary knowledge increases classification accuracy of the PRAM set. For the non-PRAM set classification improves in terms of Sensitivity, Accuracy and AUC. In particular sensitivity on the PRAM set increases by 20% from the baseline. In addition a large gain in AUC from the baseline reflects the increased probability that a positive example is ranked higher than a negative example. In fact, when the decision tree is supplemented with secondary knowledge (the PRAM set) we gain an increase in AUC, and when secondary knowledge cannot be so easily applied (non-PRAM set), the performance only improves marginally on that of the baseline. These results demonstrate that the incorporation of formalized secondary knowledge sources can help with classification in such domains unraveling the concept to be learned by the decision tree and thus reducing the overall complexity of the dataset by exploiting domain knowledge implicitly present in the data. The results represent an overall improvement on previous research into classification of clinical data with tree-based classifiers [8, 9].

However from the results in Table 3 we also observe a decrease of 10% in specificity between the Non-PRAM set and the baseline. This performance is inadequate in terms of achieving a balance between high sensitivity and high specificity. We note however that in terms of the problem domain high sensitivity and low specificity on the positive class translates to the fact that the classifier is very accurate in identifying *moderate/severe* patients and recommending they are kept for an extended time in the emergency department, however at the same time the classifier is overconservative in recommending that *mild* patients are kept for longer than usual stays. Such direction of classification a less serious error than one occurring in the opposite situation.

4.4 Automatic and Expert-Driven Feature Selection

It is acceptable to state that a reduction in dimensionality of data can reduce the complexity of underlying concepts that it may represent. The purpose of this experiment is to demonstrate that data complexity in this domain requires more than dimensionality reduction to reduce its complexity. We compare results obtained from applying automatic and expert-driven feature selection methods to those obtained by partitioning the data according to PRAM secondary knowledge. Automatic feature selection is based on standard methods used by the data mining community and are available in the Weka software. The expert-driven feature selection methods are based on selecting attributes observed to be useful to classification from our repeated experiments and by an expert and those outlined by the PRAM scoring system. 10 methods of feature automatic selection were applied to the dataset where each was used in conjunction with a decision tree for classification. The results for the best four methods are shown in rows 1-4 of Table 4. Comparing the results for automatic feature selection to those for the baseline as outlined in Table 3 we can conclude it is not successful in reducing the complexity of the dataset. In general results do not display any improvement in classification except in the case where a wrapper using a Naive Bayes classifier for optimization is used for feature selection. Here we note an increase in AUC, however this is at the expense of a large decrease in specificity. In applying expert feature selection, we built one classifier using all data records of attributes collected during the reassessment only and another classifier using only the attributes that were mapped from the PRAM scoring system while still using all instances available in the dataset. The results for these two experiments are shown in rows 5-6 of Table 4. Again comparing these results to those outlined for the baseline in Table 3 we observe no significant improvement.

However, by comparing the results from Table 4 to those for classification on the PRAM and non-PRAM sets in Table 3 a number of important conclusions can be drawn. Partitioning data into different sets for classification based on secondary knowledge results in much improved classification that of using either automatic or expert feature selection. Augmenting the developed classifier with external knowledge allows for more effective classification by exploiting underlying domain knowledge in the dataset and by organizing data according to these concepts. Such classification accuracy cannot be captured by a classification model developed on the data alone. The partitioning of data does not reduce the dimensionality of the dataset like traditional methods for classification such as feature selection, however it manages to reduce the complexity of the dataset by using secondary knowledge to identify more coherent sets into which data more naturally fits.

The intention is to use the classification results from the PRAM and non-PRAM sets from Table 3 to implement a prediction model for asthma severity. This can be achieved in a number of ways. One option is to develop a metaclassifier that could learn to direct new instances to either the model built on the PRAM set or the model built on the non-PRAM set. For such a metaclassifier values of PRAM attributes alone may be sufficient to make the decision or it

Table 4. Automatic and Expert Feature Selection

Feature Selection Mode	Mode	Size	Sens	Spec	Acc	AUC
Information Gain	Automatic	362	72	63	68	69
Chi-squared	Automatic	363	72	63	68	69
Combinatorial	Automatic	362	72	65	69	71
Wrapper with Naive Bayes	Automatic	362	71	60	70	77
On All Attributes	Expert	362	72	66	70	73
On Only PRAM Attributes	Expert	362	77	78	70	71

may be necessary to develop a method by which unseen patients can be related to the sets (PRAM and non-PRAM) we identify in the dataset. Alternatively the predictions from both sets could be combined to perform the prediction task. One option is to use a voting mechanism, another is to build these classifiers in a manner that produce rankings of the severity of the exacerbation. With such a methodology the classifier with the highest ranking provides a better insight into the condition. However, such an approach introduces additional issues in terms of interpretations and calibrations of ranks and probabilities. Such a study remains as part of our future research directions.

5 Discussion

We have introduced an approach to mining complex retrospective clinical data by incorporating secondary knowledge to supplement the classification task by reducing the complexity of the dataset. The methodology involves identification of a secondary knowledge source suitable for the clinical domain, formalization of the knowledge to analyze and organize data according to the underlying principle of the secondary knowledge, and incorporation of the secondary knowledge into the chosen classification model. In this research we concentrated on classifying information using a decision tree to satisfy the requirement that classification should be easily interpreted by domain users. From our experimental results we draw a number of conclusions. Firstly we have demonstrated that domain knowledge is implicit in the data as the dataset partitions naturally into two sets for classification with the application of a formalized mapping from the PRAM scoring system. This is in spite of the fact that the mapping was inexact; our dataset only contained four of the five attributes outlined by PRAM and some attribute values had slightly different representations. In such a way the application of secondary knowledge reduces the complexity of the dataset by allowing for the exploitation of underlying domain knowledge to supplement data analysis, representation and classification. As outlined, this approach is more successful than traditional methods for reducing data complexity such as feature selection which fail to capture a measure of the expert knowledge implicit in the retrospectively collected data. A further advantage of the approach was demonstrated by the ability of the secondary knowledge to help identify outlier examples in the data.

However, the results are still somewhat disappointing in terms of achieving a balance acceptable in medical practice between high sensitivity and high specificity in the non-PRAM set. We believe that a high proportion of missing values in this set is causing difficulties for the classification model. This issue remains an open problem for future research. In other future work we are interested in further investigating attributes used by the PRAM system and to test whether all attributes used by PRAM are necessary for enhanced classification.

References

1. Motulsky, H.: *Intuitive Biostatistics*. Oxford University Press, New York (1995)
2. Ledley, R.S., Lusted, L.B.: Reasoning foundations of medical diagnosis. *Science* **130** (1959) 9–21
3. Mullins, I.M., Siadaty, M.S., Lyman, J., Scully, K., Garrett, C.T., Miller, W.G., Muller, R., Robson, B., Apte, C., Weiss, S., Rigoutsos, I., Platt, D., Cohen, S., Knaus, W.A.: Data mining and clinical data repositories: Insights from a 667,000 patient data set. *Computers Biology and Medicine* **36**(12) (2006) 1351–77
4. Magoulas, G.D., Prentza, A.: Machine learning in medical applications. *Lecture Notes in Computer Science* **2049** (2001) 300–307
5. Cios, K.J., Moore, G.W.: Uniqueness of medical data mining. *A. I. in medicine* **26**(1-2) (2002) 1–24
6. Lavrac, N.: Selected techniques for data mining in medicine. *Artificial Intelligence in Medicine* **16**(1) (1999) 3–23
7. Lozano, P., Sullivan, S., Smith, D., Weiss, K.: The economic burden of asthma in us children: estimates from the national medical expenditure survey. *The Journal of allergy and clinical immunology* **104**(5) (1999) 957–63
8. Kerem, E., Tibshirani, R., Canny, G., Bentur, L., Reisman, J., Schuh, S., Stein, R., Levison, H.: Predicting the need for hospitalization in children with acute asthma. *Chest* **98** (1990) 1355–1361
9. Lieu, T.A., Quesenberry, C.P., Sorel, M.E., Mendoza, G.R., Leong, A.B.: Computer-based models to identify high-risk children with asthma. *American Journal of Respiratory and Critical Care Medicine* **157**(4) (1998) 1173–80
10. Sackett, D., Rosenberg, W., Muir Gray, J., Haynes, R., Richardson, W.: Evidence based medicine: what it is and what it isn't. *British Medical Journal* (1996)
11. Chalut, D.S., Ducharme, F.M., Davis, G.M.: The preschool respiratory assessment measure (pram): A responsive index of acute asthma severity. *Pediatrics* **137**(6) (2000) 762–768
12. Witten, I.H., Frank, E.: *Data mining: Practical machine learning tools and techniques*. (2005)
13. Sox, H.C.J., Blatt, M.A., Higgins, M.C., Marton, K.I.: *Medical Decision Making*. (Boston, 1998)
14. Faraggi, D., Reiser, B.: Computer-based models to identify high-risk children with asthma. *American Journal of Respiratory and Critical Care Medicine* **157**(4) (1998) 1173–80
15. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. *The Third International Conference on Knowledge Discovery and Data Mining* (1997) 34–48
16. Fawcett, T.: Roc graphs: Notes and practical considerations for data mining researchers (2003) Technical Report HPL-2003-4, HP Labs.

Evaluating a Trading Rule Mining Method based on Temporal Pattern Extraction

Hidenao Abe¹, Satoru Hirabayashi², Miho Ohsaki³, and Takahira Yamaguchi⁴

¹Department of Medical Informatics, Shimane University, School of Medicine
abe@med.shimane-u.ac.jp

²Graduate School of Science and Technology, Keio University
and_joy@ae.keio.ac.jp

³Faculty of Engineering, Doshisha University
mohsaki@mail.doshisha.ac.jp

⁴Faculty of Science and Technology, Keio University
yamaguti@ae.keio.ac.jp

Abstract. In this paper, we present an evaluation of the integrated temporal data mining environment for trading dataset from the Japanese stock market. Temporal data mining is one of key issues to get useful knowledge from databases. However, users often face difficulties during such temporal data mining process for data pre-processing method selection/construction, mining algorithm selection, and post-processing to refine the data mining process as shown in other data mining processes. To get more valuable rules for experts from a temporal data mining process, we have designed an environment which integrates temporal pattern extraction methods, rule induction methods and rule evaluation methods with visual human-system interface. After implementing this environment, we have done a case study to mine temporal rules from a Japanese stock market database for trading. The result shows the availability to find out useful trading rules based on temporal pattern extraction

1 Introduction

In recent years, KDD (Knowledge Discovery in Databases) [3] has been widely known as a process to extract useful knowledge from databases. In the research field of KDD, ‘Temporal (Time-Series) Data Mining’ is one of important issues to mine useful knowledge such as patterns, rules, and structured descriptions for a domain expert. However, huge numerical temporal data such as stock market data, medical test data, and sensor data have been only stored to databases.

Besides, many temporal mining schemes such as temporal pattern extraction methods and frequent itemset mining methods have been proposed to find out useful knowledge from numerical temporal databases. Although each method can find out partly knowledge of each suggested domains, there is no systematic framework to utilize each given numerical temporal data through whole of the KDD process.

To above problems, we have developed an integrated temporal data mining environment, which can apply numerical temporal data to find out valuable

knowledge systematically. The environment consists of temporal pattern extraction, mining, mining result evaluation support system to attempt numerical temporal data from various domains.

In this paper, we present an evaluation of the integrated temporal data mining environment with Japanese stock market data. Then, we discuss about the availability of the temporal rule mining process based on temporal pattern extraction.

2 Related Work

Many efforts have been done to analyze temporal data at the field of pattern recognitions. Statistical methods such as autoregressive model and ARIMA (AutoRegressive Integrated Moving Average) have been developed to analyze temporal data, which have linearity, periodicity, and equalized sampling rate. As signal processing methods, Fourier transform, Wavelet, and fractal analysis method have been also developed to analyze such well formed temporal data. These methods based on mathematic models restrict input data, which are well sampled. However, temporal data include ill-formed data such as clinical test data of chronic disease patients, purchase data of identified customers, and financial data based on social events. To analyze these ill-formed temporal data, we take another temporal data analysis method such as DTW (Dynamic Time Wrapping)[1], temporal clustering with multiscale matching [5], and finding Motif based on PAA (Piecewise Approximation Aggregation) [6].

To find out useful knowledge to decide orders for stock market trading, many studies have done. For example, temporal rule induction methods such as Das's framework [2] have been developed. Frequent itemset mining methods are also often attempt to the domain [15]. Although they analyze the trend of price movement, many trend analysis indices such as moving average values, Bolinger band signals, MACD signals, RSI and signals based on balance table are often never considered. In addition, these studies aim not to find out decision support knowledge, which directly indicates orders for stock market trading, but useful patterns to think better decision by a domain expert. Therefore, the decision support of trading order is still costly task even if a domain expert uses some temporal data analysis methods. The reason of this problem is that decision criteria of trading called anomaly are obtained from very complex combination of many kinds of indices related to the market by domain experts.

3 An integrated temporal data mining environment

Our temporal data mining environment needs temporal data as input. Output rules are if-then rules, which have temporal patterns or/and ordinal clauses, represented in $A=x$, $A<=y$, and $A>z$. Combinations of extracted patterns and/or ordinal clauses can be obtained as if-then rules by a rule induction algorithm. Fig. 1 illustrates a typical output it-then rule visualized with our temporal data mining environment.

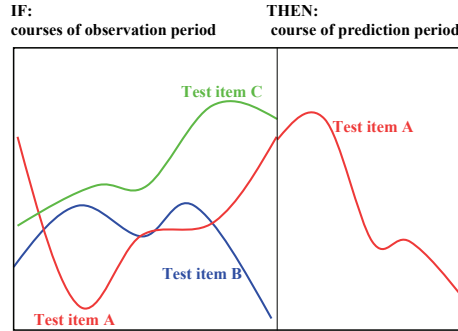


Fig. 1. Typical output if-then rule, which consists of patterns both its antecedent and its consequent.

To implement the environment, we have analyzed temporal data mining frameworks [2, 10]. Then, we have identified procedures for pattern extraction as data pre-processing, rule induction as mining, and evaluation of rules with visualized rule as post-processing of mined result. The system provides these procedures as commands for users. At the same time, we have designed graphical interfaces, which include data processing, validation for patterns on elemental sequences, and rule visualization as graphs. Fig. 2 shows us a typical system flow of this temporal data mining environment.

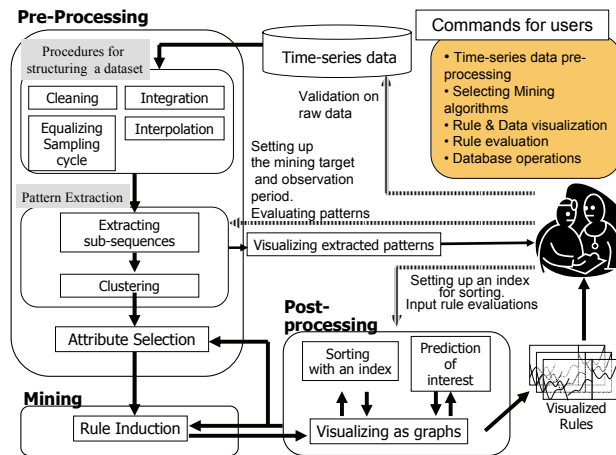


Fig. 2. A system flow view of the integrated temporal data mining environment.

3.1 Details of procedures to mine temporal rules

We have identified procedures for temporal data mining as follows:

Data pre-processing

- pre-processing for data construction
- temporal pattern extraction
- attribute selection

Mining

- rule induction

Post-processing of mined results

- visualizing mined rule
- rule selection
- supporting rule evaluation

Other database procedures

- selection with conditions
- join

As data pre-processing procedures, pre-processing for data construction procedures include data cleaning, equalizing sampling rate, interpolation, and filtering irrelevant data. Since these procedures are almost manual procedures, they strongly depend on given temporal data and a purpose of the mining process. Temporal pattern extraction procedures include determining the period of sub-sequences and finding representative sequences with a clustering algorithm such as K-Means, EM clustering and the temporal pattern extraction method developed by Ohsaki et al. [12]. Attribute selection procedures are done by selecting relevant attributes manually or using attribute selection algorithms [7].

At mining phase, we should choose a proper rule induction algorithm with some criterion. There are so many rule induction algorithms such as Version Space [9], AQ15 [8], C4.5 rule [13], and any other algorithm. To support this choice, we have developed a tool to construct a proper mining application based on constructive meta-learning called CAMLET. However, we have taken PART [4] implemented in Weka [16] in the case study to evaluate improvement of our pattern extraction algorithm.

To predict class of a test dataset with learned a classification model, the system should formally predict pattern symbols of the test dataset using some classification learning method L based on the training dataset as shown in Figure 3.

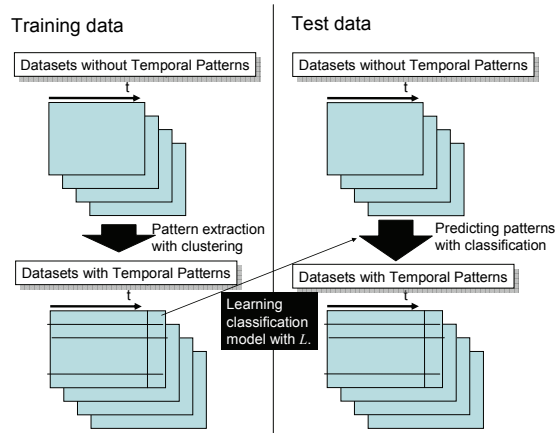


Fig. 3. The process to obtain a test dataset based on temporal patterns of a training dataset using classification learning algorithm.

To validate mined rules correctly, users need readability and ease for understand about mined results. We have taken 39 objective rule evaluation indexes to select mined rules [11], visualizing and sorting them depended on users' interest. Although these two procedures are passive support from a viewpoint of the system, we have also identified active system reaction with prediction of user evaluation based on objective rule evaluation indexes and human evaluations.

Other database procedures are used to make target data for a data mining process.

Since the environment has been designed based on open architecture, these procedures have been able to develop separately. To connect each procedure, we have only defined input/output data format.

4 Evaluating Temporal Rule Mining Performances with the Integrated Temporal Data Mining Environment

After implementing the integrated temporal data mining environment described in Section 3, we have done a case study on Japanese stock market database.

In this case study, we firstly gathered temporal price data and its trend index values through Kaburobo SDK [17]. Then, using the environment, we evaluated the performance of if-then rules based on temporal patterns. Finally, with regarding to the results, we discuss about the availability of our temporal rule mining based on temporal pattern extraction.

4.1 Description about Temporal Datasets

Using Kaburobo SDK, we got four price values, trading volume, and 13 trend index values as shown in Table 1. Excepting DMI, volume ratio, and momentum, the trend indices are defined as trading signals: buy and sell. The attribute values of these indices are converted from 1.0 to -1.0. Thus, 0 means nothing to do (or hold on the stock) for these attributes.

Table 1. The description about attributes from Kaburobo SDK.

Attribute name	Description	
R A W	opening	opening price of the day (O_t)
	high	Highest price of the day (H_t)
	low	Lowest price of the day (L_t)
	closing	Closing price of the day (C_t)
	Volume	Volume of the day (V_t)
T R E N D	Moving Average	Buy: if $SMA - LMA < 0 \cap SMA_{t-1} - LMA_{t-1} > 0$, Sell: if $SMA - LMA > 0 \cap SMA_{t-1} - LMA_{t-1} < 0$ Where $SMA = (C_t + C_{t-1} + \dots + C_{t-13})/13$, and $LMA = (C_t + C_{t-1} + \dots + C_{t-25})/26$
	Bolinger Band	Buy: if $C_t \geq (MA + 2\sigma) \times 0.05$, Sell: if $C_t \leq (MA - 2\sigma) \times 0.05$ where $MA = (C_t + C_{t-1} + \dots + C_{t-24})/25$
	Envelope	Buy: if $C_t \geq MA + (MA \times 0.05)$, Sell: if $C_t \leq MA - (MA \times 0.05)$
	HLband	Buy: if $C_t < LowLine_{-10days}$, Sell: if $C_t > HighLine_{-10days}$
I N D I C E S	MACD	Buy: if $MACD - AvgMACD_{-9days} > 0 \cap MACD_{t-1} - AvgMACD_{(t-1)-9days} < 0$ Sell: if $MACD - AvgMACD_{-9days} < 0 \cap MACD_{t-1} - AvgMACD_{(t-1)-9days} > 0$ Where $MACD = EMA_{12days} - EMA_{26days}$, $EMA = EMA_{t-1} + (2/range + 1)(C_{t-1} - EMA_{t-1})$
	DMI	Buy: if $PDI - MDI > 0 \cap PDI_{t-1} - MDI_{t-1} < 0$, Sell: if $PDI - MDI < 0 \cap PDI_{t-1} - MDI_{t-1} > 0$ Where $PDI = \sum_{i=t-1}^t (H_i - H_{i-1}) \times \sum_{i=t-1}^t TR_i \times 100$, $MDI = \sum_{i=t-1}^t (L_i - L_{i-1}) \times \sum_{i=t-1}^t TR_i \times 100$ $TR_i = \max\{(H_i - C_{i-1}), (C_{i-1} - L_i), (H_i - L_i)\}$
	volumeRatio	$VR_t = \left(\sum_{i=t-25}^t V_i + \sum_{i=t-25}^t V_i \right) / \left(\sum_{i=t-25}^t V_i + \sum_{i=t-25}^t V_i \right) \times 100$
	RSI	$RSI_t = 100 - 100 / \left(1 + \frac{\sum_{i=t-13}^t (C_{i+1} - C_i)}{\sum_{i=t-13}^t (C_i - C_{i+1})} \right)$
	Momentum	$M_t = C_t - C_{t-10}$
	Ichimoku1	Buy: if $C_{t-1} < RL_{t-9days} \cap C_t > RL_{t-9days}$, Sell: if $C_{t-1} > RL_{t-9days} \cap C_t < RL_{t-9days}$ Where $RL_{t-9days} = \text{average}(\max(H_i) + \min(L_i))$ ($i = t-8, t-7, \dots, t$)
	Ichimoku2	Buy: if $C_{t-1} < RL_{t-26days} \cap C_t > RL_{t-26days}$, Sell: if $C_{t-1} > RL_{t-26days} \cap C_t < RL_{t-26days}$ Where $RL_{t-26days} = \text{average}(\max(H_i) + \min(L_i))$ ($i = t-25, t-24, \dots, t$)
	Ichimoku3	Buy: if $RL_{(t-2)-26days} < RL_{(t-2)-9days} \cap RL_{(t-1)-26days} > RL_{(t-1)-9days} \cap RL_{(t-1)-26days} < RL_{t-26days}$ Sell: if $RL_{(t-2)-26days} > RL_{(t-2)-9days} \cap RL_{(t-1)-26days} < RL_{(t-1)-9days} \cap RL_{(t-1)-26days} > RL_{t-26days}$
	Ichimoku4	Buy: if $C_t > AS1_{t-26} \cap C_t > AS2_{t-26}$, Sell: if $C_t < AS1_{t-26} \cap C_t < AS2_{t-26}$ Where $AS1_t = \text{median}(RL_{t-9days} - RL_{t-26days})$, $AS2_t = (\max(H_t) - \min(L_t)) / 2$ ($i = t-51, t-50, \dots, t$)

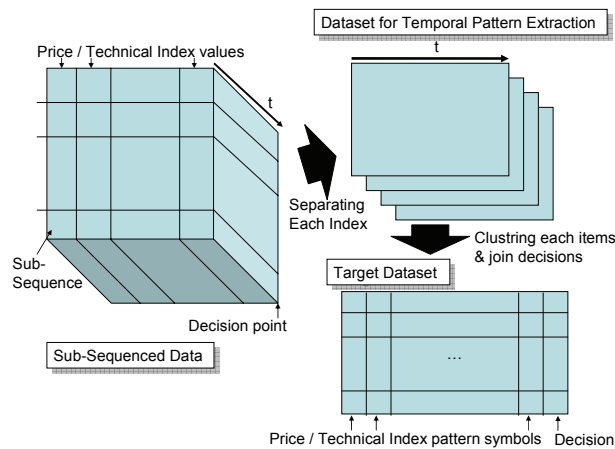
We obtained temporal data consists of the above mentioned attributes about five financial companies and four telecommunication companies as follows: Credit Saison (Saison), Orix, Mitsubishi Tokyo UFJ Financial Group (MUFJFG), Mitsui Sumitomo Financial Group (MSFG), Mizuho Financial Group (MizuhoFG), NTT, KDDI, NTT Docomo, and Softbank. The period, which we have collected from the temporal stock data, is from 5th January 2006 to 31st May 2006. For each day, we have made decisions as the following: the decision is if the closing value rises 5% within 20 days then 'buy', otherwise if the closing value falls 5% within 20 days then 'sell', otherwise 'hold'. We set these decisions as the class attribute to each target instance. Table 2 shows the class distributions about the nine stocks for the period.

Table 2. The distributions of decisions of the nine stocks during the five months.

Finance	Buy	sell	Telecom.	buy	sell
Saison	37	53	NTT	27	32
Orix	43	40	KDDI	42	39
MUFJFG	0	50	NTTdocomo	19	29
MSFG	6	27	Softbank	23	69
MizuhoFG	38	31			

For each gathered temporal data of the nine stocks, the system extracted temporal patterns for each attribute. Then, the symbols of each pattern and the decision of each day joined as each instance of the target dataset as illustrated in Figure 4.

Fig. 4. An illustration of the process to obtain target datasets from temporal data



4.2 Mining results of the nine temporal stock data

To extract temporal patterns, we have used K-Means and EM algorithm, which are implemented in Weka. Then, to predict temporal pattern of each test dataset, we have used Boosted C4.5 [14], which is also implemented in Weka.

As shown in Table 3, predicting temporal patterns for test dataset are succeeded, because the accuracies of the nine dataset are satisfactory high scores as a classification task.

Table 3. Accuracies (%) of re-substitution on the two temporal pattern extraction with K-Means and EM algorithm.

Finance	K-Means	EM	Telecom.	K-Means	EM
Saison	90.1	88.9	NTT	84.8	90.9
Orix	88.9	84.8	KDDI	86.9	78.8
MUFJFG	90.9	93.9	NTTdocomo	80.8	85.9
MSFG	96.0	90.9	Softbank	93.9	89.9
MizuhoFG	92.9	83.8			

Table 4 shows accuracies (%) of cross stock evaluation on the two temporal pattern extraction algorithms. The cross stock evaluation uses different stocks as training dataset and test dataset. Stocks in rows mean training datasets, and columns mean test datasets. As shown in this table, bolded accuracies go beyond 50%, which means that the mined rules work better than just predicting sell or buy. The result shows the performance of our temporal rules depends on the similarity of trend values rather than the field of each stock.

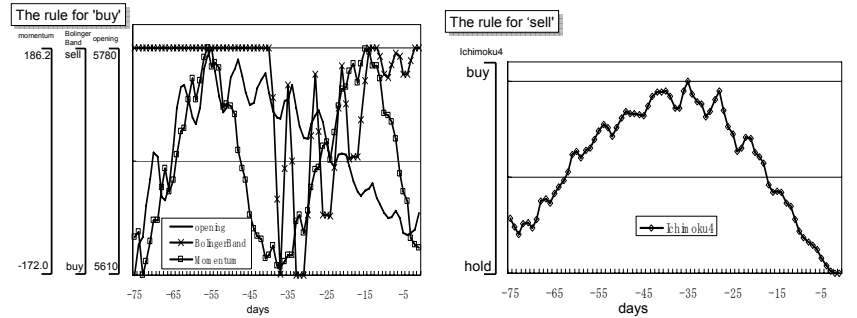
Table 4. Accuracies (%) of cross stock evaluation with temporal patterns using K-Means and EM algorithm.

K-Means	Saison	MUFJFG	MSFG	MizuhoFG	Orix	NTT	KDDI	NTTdocomo	Softbank
Saison		44.4	28.3	31.3	40.4	29.3	35.4	22.2	49.5
MUFJFG	46.5		44.4	30.3	42.4	32.3	39.4	29.3	55.6
MSFG	44.4	24.2		38.4	31.3	28.3	27.3	29.3	22.2
MizuhoFG	46.5	31.3	33.3		29.3	22.2	20.2	22.2	58.6
Orix	38.4	50.5	27.3	31.3		32.3	39.4	19.2	30.3
NTT	14.1	50.5	27.3	31.3	14.1		39.4	37.4	6.1
KDDI	12.1	44.4	56.6	27.3	31.3	41.4		55.6	16.2
NTTdocomo	26.3	40.4	52.5	33.3	23.2	30.3	20.2		8.1
Softbank	44.4	28.3	18.2	45.5	34.3	40.4	30.3	26.3	

EM	Saison	MUFJFG	MSFG	MizuhoFG	Orix	NTT	KDDI	NTTdocomo	Softbank
Saison		46.5	28.3	31.3	38.4	51.5	65.7	21.2	32.3
MUFJFG	31.3		51.5	31.3	38.4	29.3	41.4	22.2	46.5
MSFG	23.2	58.6		34.3	31.3	43.4	32.3	30.3	29.3
MizuhoFG	35.4	31.3	34.3		31.3	42.4	38.4	43.4	20.2
Orix	41.4	29.3	39.4	34.3		37.4	21.2	28.3	25.3
NTT	41.4	21.2	20.2	42.4	44.4		33.3	23.2	39.4
KDDI	61.6	59.6	50.5	28.3	27.3	42.4		28.3	37.4
NTTdocomo	27.3	42.4	29.3	52.5	25.3	30.3	19.2		28.3
Softbank	52.5	45.5	27.3	31.3	41.4	33.3	43.4	19.2	

Figure 5 shows an example of temporal rules. These rules are obtained from the training dataset with EM algorithm temporal pattern extraction for Saison. As shown in Table 3, the rule set of Saison works the best to KDDI test dataset.

Fig. 5. An example of rule for 'buy' and rule for 'sell'.



4.3 Discussion about the temporal rule mining

The prediction of decisions for each dataset works correctly with regarding to the result of Table 3, predicting temporal patterns of test dataset with a classification learning algorithm. However, mined rules based on temporal patterns are rather over fitting to each training dataset as shown in Table 4. One of the solutions to avoid over fitting will be to mine a temporal rule set from a training dataset, which consists of multiple stocks.

With regarding to Figure 5, our temporal rule mining system can find out adequate combinations of trend index patterns for each stock. To learn adequate trend index pattern combinations is very costly work for trading beginners. Thus, our temporal rule mining can support traders who want to know the adequate combinations of trend indices for each stock.

5 Conclusion

We have designed and implemented a temporal data mining environment, which integrates temporal pattern extraction, rule induction, and rule evaluation.

As the result of the case study on the nine Japanese stock datasets, this environment mines valuable temporal rules to predict different stock decisions based on temporal pattern extraction. The result also indicated the availability to support stock traders to learn adequate combinations of trend index patterns.

In future, we will evaluate trading result with the predictions of decisions by each mined temporal rule set on the stock trading simulator included in Kaburobo SDK.

Although we have not tried to select proper algorithms for the temporal pattern extraction procedure, the attribute selection procedure and the mining procedure, it is also able to connect subsystems for selecting each proper algorithm to this environment.

References

1. Berndt, D. J. and Clifford, J. "Using dynamic time wrapping to find patterns in time series", in *Proc. of AAAI Workshop on Knowledge Discovery in Databases* (1994) pp.359-370
2. Das, G., King-Ip, L., Heikki, M., Renganathan, G., and Smyth, P., "Rule Discovery from Time Series", in *Proc. of International Conference on Knowledge Discovery and Data Mining* (1998) pp.16-22
3. Fayyad, U. M., Piatetsky-Shapiro, G., and Smyth, P., "From Data Mining to Knowledge Discovery: An Overview", *Advances in Knowledge Discovery and Data Mining*, AAAI Press / The MIT Press, CA (1996) pp.1-34
4. Frank, E., Witten, I. H., "Generating accurate rule sets without global optimization", in *Proc. of the Fifteenth International Conference on Machine Learning* (1998) pp.144-151
5. Hirano, S. and Tsumoto, S., "Mining Similar Temporal Patterns in Long Time-Series Data and Its Application to Medicine", in *Proc. of the 2002 IEEE International Conference on Data Mining* (2002) pp.219-226
6. Lin, J., Keogh, E., Lonardi, S., and Patel, P., "Finding Motifs in Time Series", in *Proc. of Workshop on Temporal Data Mining* (2002) pp.53-68
7. Liu, H., and Motoda, H., "Feature selection for knowledge discovery and data mining", Kluwer Academic Publishers (1998)
8. Michalski, R., Mozetic, I., Hong, J., and Lavrac, N., "The AQ15 Inductive Learning System: An Overview and Experiments", *Reports of Machine Learning and Inference Laboratory*, MLI-86-6, George Mason University, (1986)
9. Mitchell, T. M., "Generalization as Search", *Artificial Intelligence*, 18(2) (1982) pp.203-226
10. Ohsaki, M., Sato, Y., Yokoi, H., and Yamaguchi T., "A Rule Discovery Support System for Sequential Medical Data - In the Case Study of a Chronic Hepatitis Dataset -, *ECML/PKDD-2003 Workshop on Discovery Challenge*, (2003) pp.154-165
11. Ohsaki, M., Kitaguchi, S., Okamoto, K., Yokoi, H., and Yamaguchi, T., "Evaluation of Rule Interestingness Measures with a Clinical Dataset on Hepatitis", in *Proc. of ECML/PKDD 2004*, LNAI3202 (2004) pp.362-373
12. Ohsaki, M., Abe, H., Kitaguchi, S., Kume, S., Yokoi, H., and Yamaguchi, T., "Development and Evaluation of an Integrated Time-Series KDD Environment - A Case Study of Medical KDD on Hepatitis-", *Joint Workshop of Vietnamese Society of Artificial Intelligence, SIGKBS-JSAI, ICS-IPSI and IEICE-SIGAI on Active Mining* (2004) No.23.
13. Quinlan, J. R., "Programs for Machine Learning", Morgan Kaufmann (1992)
14. Quinlan, J. R., "Bagging, Boosting and C4.5", *AAAI/IAAI*, 1 (1996) pp. 725-730
15. Raymond, W., and Ada, F., "Mining top-K frequent itemsets from data streams", *Data Mining and Knowledge Discovery*, 13(2) (2006) pp.193-217
16. Witten, I. H. and Frank, E., "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann, San Francisco (2000)
17. KabuRobo: [<http://www.kaburobo.jp>] (in Japanese)

Discriminant Feature Analysis for Music Timbre Recognition

Xin Zhang¹, Zbigniew W. Ras^{1,2}

¹ Computer Science Department, University of North Carolina, Charlotte, N.C., USA,

² Polish-Japanese Institute of Information Technology, 02-008 Warsaw, Poland
{xinzhang, ras}@uncc.edu

Abstract. The high volume of digital music recordings in the internet repositories has brought a tremendous need for automatic recommendation system based on content data to help users to find their favorite music items. Music instrument identification is one of the important subtasks of content-based automatic indexing, for which the authors have developed novel new temporal features and implemented a high dimensional sound feature database with all the low-level MPEG7 descriptors as well as popular features in the literature. This paper presents development details of these new features and evaluates them among other 300 features in the database by a logistic discriminant analysis for improving music instrument identification efficiency of rule-based classifiers.

Keywords: Automatic Indexing, Music Information Retrieval, MPEG7, Timbre Estimation, Logistic Discriminant Analysis, Feature Selection, and Machine Learning.

1 Introduction

The state of art technologies in network and computer storage boost the fast growing of music repositories throughout the Internet, which brought the need for music intelligence techniques to help users to find their favorite music items.

Aiming to provide efficient music recommendations, data mining of different representations of musical files (e.g., music recording, MIDI file, and music notes) involves very different techniques respectively: research in MIDI files and music notes tackles problems in text mining, where intuitively Music Information Retrieval (MIR) needs not to be considered; knowledge discovery in digital recordings requires MIR of sound features from musical sound signals, since digital recordings contain only sound signals unless manually labeled with semantic descriptions (e.g., author, title, and company). Timbre identification is one of the important subtasks for mining digital recordings, where timbre is a quality of sound that distinguishes one music instrument

from another. Researchers in this area have investigated a number of acoustical features to build computational model for timbre identification. In this paper, authors focus on developing automated indexing solutions for digital recordings based on MIR techniques of instruments and their types.

The real use of timbre-based grouping of music is very nicely discussed in [3]. Methods in research on automatic musical instrument sound classification go back to last few years. Next, the paper reviews these algorithms in two categories: monophonic and polyphonic, which are defined by the total instruments within the target sounds.

For monophonic sounds, a number of acoustic features have been explored by researchers in this field ([1] and [4]). Some of them are quite successful on certain simple sound data (monophonic, short, of limited instrument types). The challenges for instrument identification in this type of sounds are at least in these areas: a digital multimedia file normally contains a huge amount of data, where subtle changes of sound amplitude in time can be critical for human perception system, thus the data-driven timbre identification process demands lots of information to capture and describes the patterns among those subtle changes; since after the dimensional approach to timbre description was proposed by [3], so far there is no standard parameterization used as a classification basis. Researchers in the area have been explored different statistical parameters to describe patterns and properties of the spectrums of music sounds to distinguish different timbre, such as Tristimulus parameters [10], brightness [5], and irregularity [16], etc. Based on recent research performed in this area, MPEG-7 standard provides a set of low-level temporal and spectral sound features where some of them are in a form of vector or matrix of a large size. Flattening and summarizing these features for traditional classifiers intuitively increases the number of features and losses some potentially useful information. Therefore, in this paper, authors are proposing new features which are sufficient in musical timbre signatures and suitable in format for machine learning classifiers. Authors compare them against popular features in the literature to improve the classification efficiency by logistic regression studies.

For polyphonic sounds, different methods have been investigated by various researchers and mathematicians, such as Independent Component Analysis (ICA) ([6] and [15]), Factorial Hidden Markov Models (HMM) ([9] and [13]), and Harmonic Sources Separation Algorithms ([2], [18], [7], and [20]). ICA requires multiple channels of different sound sources. Most often, HMM works well for sound sources separation, where fundamental frequency range is small and the variation is subtle. Harmonic Sources Separation Algorithms can be used to isolate sound sources within a single channel, where efficient solution in one channel can be intuitively applied to other channels and therefore facilitates more types of sound recordings (e.g., mono-channel and stereo with two or more channels). The authors also have interest to apply the proposed features to their harmonic sound source isolation system [18] in order to improve its isolation accuracy.

2 Audio Features in our research

In their previous work, authors implemented aggregation [21] to the MPEG7 spectral descriptors as well as other popular sound features in the literature. This section introduces new temporal features that have been implemented in author's database of music instruments, and then it describes the rest of the features in the database. The spectrum features have two different frequency domains: Hz frequency and Mel frequency. Frame size is carefully designed to be 120ms, so that the 0th octave G (the lowest pitch in our audio database) can be detected. The hop size is 40ms with a overlapping of 80ms. Since the sample frequency of all the music objects is 44,100Hz, the frame size is 5292. A hamming window is applied to all STFT transforms to avoid jittering in the spectrum.

2.1 Temporal features based on pitch

Pitch trajectories of instruments behaves very differently in time. The authors designed parameters to capture the power change in time.

Pitch Trajectory Centroid PC is used to describe the center of gravity of the power of the fundamental frequency during the quasi-steady state.

$$PC = \frac{\sum_{n=1}^{length(P)} n / length(P) \cdot P(n)}{\sum_{n=1}^{length(P)} P(n)} \quad (1.)$$

where P is the pitch trajectory in the quasi-steady state, n is the n^{th} frame.

Pitch Trajectory Spread PS is the RMS deviation of the Pitch Trajectory with respect to its gravity center.

$$PS = \sqrt{\frac{\sum_{n=1}^{length(P)} (n / length(P) - PC)^2 \cdot P(n)}{\sum_{n=1}^{length(P)} P(n)}} \quad (2.)$$

Pitch Trajectory Max Angle PM is an angle of the normalized power maximum vs. its normalized frame position along the trajectory in the quasi-steady state.

$$PM = \frac{MAX\{P(n)\} - P(0)}{\frac{1}{length(P)} \sum_{n=1}^{length(P)} P(n)} \bigg/ \frac{F(n) - F(0)}{length(P)} \quad (3.)$$

where $F(n)$ is the position of n^{th} frame in the steady state.

Harmonic Peak Relation is a vector to describe the relationship among the harmonic partials.

$$HR = \frac{1}{m} \sum_{i=1}^m H_j / H_0 \quad (4.)$$

where m is the total number of frames in the steady state, H_j is the j th harmonic peak in the i th frame.

2.2 Aggregation features

MPEG7 descriptors can be categorized into two types: temporal and spectral. The authors applied aggregation among all the frames per music object for all the following instantaneous spectral features.

MPEG7 Spectrum Centroid [22] describes the center-of-gravity of a log-frequency power spectrum. It economically indicates the pre-dominant frequency range. Coefficients under 62.5Hz have been grouped together for fast computation.

MPEG7 Spectrum Spread is the Root of Mean Square value of the deviation of the Log frequency power spectrum with respect to the gravity center in a frame [22]. Like Spectrum Centroid, it is an economic way to describe the shape of the power spectrum.

MPEG7 Harmonic Centroid is computed as the average over the sound segment duration of the instantaneous Harmonic Centroid within a frame [22]. The instantaneous Harmonic Spectral Centroid is computed as the amplitude in linear scale weighted mean of the harmonic peak of the spectrum.

MPEG7 Harmonic Spread is computed as the average over the sound segment duration of the instantaneous harmonic spectral spread of frame [22]. The instantaneous harmonic spectral spread is computed as the amplitude weighted standard deviation of the harmonic peaks of the spectrum with respect to the instantaneous harmonic spectral centroid.

MPEG7 Harmonic Variation is defined as the mean value over the sound segment duration of the instantaneous harmonic spectral variation [22]. The instantaneous harmonic spectral variation is defined as the normalized correlation between the amplitude of the harmonic peaks of two adjacent frames.

MPEG7 Harmonic Deviation is computed as the average over the sound segment duration of the instantaneous Harmonic Spectral Deviation in each frame [22]. The instantaneous Harmonic Spectral Deviation is computed as the spectral deviation of the log amplitude components from a spectral envelope.

MPEG7 Harmonicity Rate is the proportion of harmonics in the power spectrum. It describes the degree of harmonicity of a frame [22]. It is computed by the normalized correlation between the signal and a lagged representation of the signal.

MPEG7 Fundamental Frequency is the frequency that best explains the periodicity of a signal [22]. The ANSI definition of psycho-acoustical terminology says that “pitch is an auditory attribute of a sound according to which sounds can be ordered on a scale from low to high”.

MPEG7 Upper Limit of Harmonicity describes the frequency beyond which the spectrum cannot be considered harmonic [22]. It is calculated based on the power spectrum of the original and a comb-filtered signal.

Tristimulus and similar parameters describe the ratio of the amplitude of a harmonic partial to the total harmonic partials [10]. They are first modified tristimulus parameter, power difference of the first and the second tristimulus parameter, grouped tristimulus of other harmonic partials, odd and even tristimulus parameters.

$$Tr_1 = A_1^2 / \sum_{n=1}^N A_n^2 \quad (5.)$$

$$h_{3,4} = \sum_{i=3}^4 A_i^2 / \sum_{j=1}^N A_j^2 \quad (6.)$$

$$h_{5,6,7} = \sum_{i=5}^7 A_i^2 / \sum_{j=1}^N A_j^2 \quad (7.)$$

$$h_{8,9,10} = \sum_{i=8}^{10} A_i^2 / \sum_{j=1}^N A_j^2 \quad (8.)$$

$$Od = \sqrt{\sum_{k=2}^L A_{2k-1}^2} / \sqrt{\sum_{n=1}^N A_n^2} \quad (9.)$$

$$Ev = \sqrt{\sum_{k=1}^M A_{2k}^2} / \sqrt{\sum_{n=1}^N A_n^2} \quad (10.)$$

$$\overline{fd}_m = \sum_{k=1}^5 A_k (\Delta f_k / (k f_1)) / \sum_{k=1}^5 A_k \quad (11.)$$

Brightness is calculated as the proportion of the weighted harmonic partials to the harmonic spectrum [5].

$$B = \sum_{n=1}^N n \cdot A_n / \sum_{n=1}^N A_n \quad (12.)$$

Transient, steady and decay duration in this research, the transient duration is considered as the time to reach the quasi-steady state of fundamental frequency. In this duration the sound contains more timbre information than pitch information that is highly relevant to the fundamental frequency. Thus differentiated harmonic descriptors values in time are calculated based on the subtle change of the fundamental frequency [19].

Zero crossing counts the number of times that the signal sample data changes signs in a frame [12] [14].

$$ZC_i = 0.5 \sum_{n=1}^N |sign(s_i[n]) - sign(s_i[n-1])| \quad (13.)$$

$$sign(x) = \begin{cases} 1, x \geq 0 \\ -1, x < 0 \end{cases} \quad (14.)$$

where S_i is the n^{th} sample in the i^{th} frame, N is the frame size.

Spectrum Centroid describes the gravity center of the spectrum [12] [17]

$$C_i = \frac{\sum_{k=1}^{N/2} f(k) |X_i(k)|}{\sum_{k=1}^{N/2} |X_i(k)|} \quad (15.)$$

where N is the total number of the FFT points, $X_i(k)$ is the power of the k^{th} FFT point in the i^{th} frame, $f(k)$ is the corresponding frequency of the FFT point.

Roll-off is a measure of spectral shape, which is used to distinguish between voiced and unvoiced speech [8]. The roll-off is defined as the frequency below which C percentage of the accumulated magnitudes of the spectrum is concentrated, where C is an empirical coefficient.

$$\sum_{k=1}^K |X_i(k)| \leq C \cdot \sum_{k=1}^K |X_i(k)| \quad (16.)$$

Flux is used to describe the spectral rate of change [12]. It is computed by the total difference between the magnitude of the FFT points in a frame and its successive frame.

$$F_i = \sum_{k=1}^{N/2} (|X_i(k)| - |X_{i-1}(k)|)^2 \quad (17.)$$

2.3 Statistical parameters

Also, to flatten the matrix data to suitable format for the classifiers, statistical parameters (e.g., maximum, minimum, average, distance of similarity, standard deviation) are applied to the power of each spectral band.

MPEG7 Spectrum Flatness describes the flatness property of the power spectrum within a frequency bin, which is ranged by edges in the following formula [22]. The value of each bin is treated as an attribute value in the database. Since the octave resolution in our research is 1/4, the total number of bands is 32.

MPEG7 Spectrum Basis Functions V_k are used to reduce the dimensionality by projecting the spectrum from high dimensional space to low dimensional space with compact salient statistical information [22]. x_i is a vector of power spectrum coefficients

in a frame t , which are transformed to Db scale and then normalized. N , the total number of frequency bins, is 32 in 1/4 octave resolution.

Mel frequency cepstral coefficients describes the spectrum according to the human perception system in the mel scale. They are computed by grouping the STFT points of each frame into a set of 40 coefficients by a set of 40 weighting curves with logarithmic transform and a discrete cosine transform (DCT).

2.4 MPEG7 temporal descriptors

The temporal descriptors in MPEG7 [22] have been applied directly into the feature database. **MPEG7 Spectral Centroid** is computed as the power weighted average of the frequency bins in the power spectrum of all frames in a sound segment with a Welch method. **MPEG7 Log Attack Time** is defined as the logarithm of the time duration between the time when the signal starts to the time it reaches its stable part, where the signal envelope is estimated by computing the local mean square value of the signal amplitude in each frame. **MPEG7 Temporal Centroid** is calculated as the time average over the energy envelope.

3 Discriminant Analysis for Feature Selection

Logistic regression model is a popular statistical approach of analyzing multinomial response variables, since it does not assume normally distributed conditional attributes which can be continuous, discrete, dichotomous or a mix of any of these; it can handle nonlinear relationships between the discrete responses and the explanatory attributes. It has been widely used to investigate the relationship between decision attribute and conditional attributes, using the most economical model. An ordinal response logit model has a form:

$$\log\left(\frac{\Pr(Y=i|x)}{\Pr(Y=k+1|x)}\right) = \alpha_i + \beta_i x, \quad i=1, \dots, k \quad (18.)$$

where the $k+1$ possible responses have no natural ordering and $\alpha_1, \dots, \alpha_k$ are k intercept parameters, β_1, \dots, β_k are k vectors of parameters, and Y is the response. For details, see [11]. The system fits a common slopes cumulative model which is a parallel lines regression model based on the cumulative probabilities of the response categories. The significance of an attribute is calculated with the likelihood ratio or chi-square difference test by the Fisher's Score algorithm. A final model is selected, where adding another variable would not improve the model significantly.

4 Experiments

The authors implemented a feature database of 1569 music recording sound objects of 74 instruments, which can play music notes, from McGill University Master Samples CD collection, which has been widely used for research on musical instrument recognition all over the world. The authors discriminated instrument types on different levels of a classification tree. The tree consists of three levels: the top level (e.g., aerophone, chordophone, and idiophone), the second level (e.g., lip-vibrated, side, reed, composite, simple, rubbed, shaken, and struck), and the third level (e.g., piano, violin, and flute.). All classifiers were 10-fold cross validation with a split of 90% training and 10% testing. We used WEKA for all classifications and SAS LOGISTIC procedure for discriminant analysis. In each experiment, a 99% confidence level is used. Feature extraction was implemented in .NET C++ with connection to MS SQL Server.

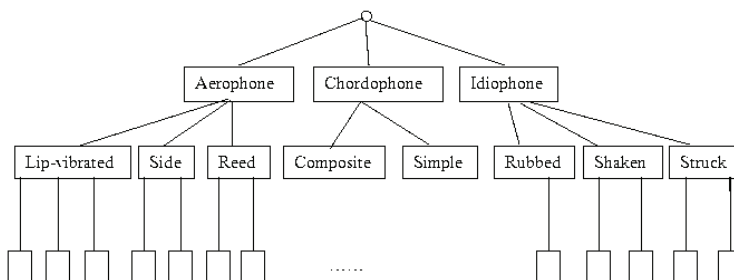


Figure 1. Music instrument classification tree.

For classification on the first level in the music instrument family tree, the selected feature set was stored in List I: { PeakRelation8, PeakRelation16, PeakRelation24, MPEGFundFreq, MPEGHarmonicRate, MPEGULHarmonicity, MPEGHarmoVariation, MPEGHarmoDeviation, MPEGFlat3, MPEGFlat8, MPEGFlat18, MPEGFlat30, MPEGFlat36, MPEGFlat46, MPEGFlat55, MPEGFlat56, MPEGFlat66, MPEGFlat67, MPEGFlat76, MPEGFlat77, MPEGFlat83, MPEGFlat85, MPEGFlat94, MPEGFlat96, MPEGSpectrumCentroid, MPEGTC, MPEGBasis59, MPEGBasis200, TristimulusRest, ZeroCrossing, MFCCMaxBand1, MFCCMaxBand3, MFCCMaxBand5, MFCCMaxBand6, MFCCMaxBand7, MFCCMaxBand8, MFCCMaxBand10, MFCCMaxBand13, MFCCMinBand1, MFCCMinBand13, PitchSpread, MaxAngle}. Experiment was also performed on the rest of features after List I was removed from the whole feature set, which was stored in List II. In the table below, “All” stands for all the attributes used for classifier construction.

Table 1. Results of three groups of features at the top level of the music family tree.

	Precision			Recall		
	List I	All	List II	List I	All	List II
Idiophone	87.00%	91.10%	95.10%	82.10%	91.40%	94.80%
Chordophone	86.80%	91.30%	88.50%	88.60%	88.90%	84.70%
Aerophone	91.50%	91.30%	87.30%	91.80%	93.50%	90.90%

Table1 shows the precisions of the classifiers constructed with selected features at the family level. After the less significant features, elected by the logistic model, have been removed, the group of List I slightly improved the precision for aerophone instruments. However, the selected significant feature group (List I) significantly outperformed in precision for aerophone instruments and in recall for both chordophone and aerophone instruments.

For classification at the second level in the music instrument family tree, the selected feature set was stored in List I: {PeakRelation8, PeakRelation16, PeakRelation30, MPEGFundFreq, MPEGHarmonicRate, MPEGULHarmonicity, MPEGHarmoDeviation, MPEGFlat3, MPEGFlat11, MPEGFlat14, MPEGFlat18, MPEGFlat22, MPEGFlat26, MPEGFlat36, MPEGFlat44, MPEGFlat46, MPEGFlat58, MPEGFlat67, MPEGFlat81, MPEGFlat82, MPEGFlat83, MPEGFlat85, MPEGFlat93, MPEGFlat94, MPEGFlat95, MPEGSpectrumCentroid, MPEGTC, MPEGBasis50, MPEGBasis57, MPEGBasis59, MPEGBasis69, MPEGBasis73, MPEGBasis116, MPEGBasis167, MPEGBasis206, Tristimulus1, TristimulusRest, TristimulusBright, ZeroCrossing, SpectrumCentroid2, RollOff, MFCCMaxBand1, MFCCMaxBand3, MFCCMaxBand4, MFCCMaxBand6, MFCCMaxBand7, MFCCMaxBand9, MFCCMinBand2, MFCCMinBand5, MFCCMinBand10, MFCCMinBand13, MFCCAvgBand10, MFCCAvgBand11, PitchSpread, MaxAngle}. Experiment was also performed on List II obtained by removing List I from the whole feature set.

Table 2. Results of three groups of features at the second level of the music family tree.

	Precision			Recall		
	List I	All	List II	List I	All	List II
Lip-Vibrated	83.80%	84.40%	77.30%	84.70%	88.80%	82.30%
Side	74.30%	73.20%	66.40%	75.70%	64.00%	64.00%
Reed	77.10%	78.30%	70.50%	78.40%	80.10%	70.50%
Composite	84.50%	86.20%	84.90%	86.70%	84.30%	83.90%
Simple	71.20%	74.10%	72.20%	67.20%	80.00%	72.80%
Rubbed	85.30%	82.10%	75.00%	78.40%	86.50%	73.00%
Shaken	79.20%	91.00%	89.50%	64.80%	92.00%	87.50%
Struck	78.20%	86.30%	85.40%	80.40%	79.00%	77.60%

Table2 shows the precisions of the classifiers constructed with the selected features, all features, and the rest of the features after selection at the second level of the instrument

family tree. After the less significant features, elected by the logistic model, have been removed, the group of List I improved the precision for side and rubbed instruments and recall for the side, composite, and struck instruments. Also, the selected significant feature group (List I) significantly outperformed in precision for lip-vibrated, side, reed, and rubbed instruments and in recall for all the types except for simple and shaken instruments.

For classification at the third level in the music instrument family tree, the selected feature set was stored in List I: { MPEGTristimulusOdd, MPEGFundFreq, MPEGULHarmonicity, MPEGHarmoVariation, MPEGFlatness6, MPEGFlatness14, MPEGFlatness27, MPEGFlatness35, MPEGFlatness43, MPEGFlatness52, MPEGFlatness63, MPEGFlatness65, MPEGFlatness66, MPEGFlatness75, MPEGFlatness76, MPEGFlatness79, MPEGFlatness90, MPEGFlatness91, MPEGSpectrumCentroid, MPEGSpectrumSpread, MPEGBasis41, MPEGBasis42, MPEGBasis69, MPEGBasis87, MPEGBasis138, MPEGBasis157, MPEGBasis160, MPEGBasis170, MPEGBasis195, TristimulusBright, TristimulusEven, TristimulusMaxFd, ZeroCrossing, SpectrumCentroid2, Flux, MFCCMaxBand2, MFCCMaxBand3, MFCCMaxBand6, MFCCMaxBand7, MFCCMaxBand9, MFCCMaxBand10, MFCCMinBand1, MFCCMinBand2, MFCCMinBand3, MFCCMinBand6, MFCCMinBand7, MFCCMinBand10, MFCCAvgBand1, MFCCAvgBand12, SteadyEnd, Length}. Experiment was also performed on List II obtained by removing List I from the whole feature set.

Table 3. Results of three groups of features at the third level of the music family tree.

	Precision		Recall			
	List I	All	List I	All	List I	All
flute	92.90%	67.70%	70.40%	89.70%	72.40%	65.50%
tubularbells	86.70%	60.00%	52.40%	72.20%	66.70%	61.10%
tuba	85.70%	81.80%	85.70%	90.00%	90.00%	90.00%
electricbass	83.10%	87.50%	89.10%	80.60%	83.60%	85.10%
Trombone	80.60%	80.00%	76.30%	69.20%	82.10%	74.40%
marimba	79.20%	89.50%	90.60%	71.40%	89.50%	86.50%
piano	78.50%	82.40%	83.00%	81.60%	78.40%	74.40%
frenchhorn	78.00%	83.70%	82.90%	87.70%	88.90%	84.00%
bassflute	77.40%	75.50%	71.20%	68.30%	61.70%	61.70%
altoFlute	76.70%	82.80%	78.60%	79.30%	82.80%	75.90%
doublebass	75.40%	60.80%	60.00%	75.40%	54.40%	52.60%
piccolo	74.50%	69.20%	62.00%	71.70%	67.90%	58.50%
ctrumpet	72.00%	68.90%	69.10%	83.10%	78.50%	72.30%
violin	71.00%	75.00%	77.10%	78.00%	72.70%	76.50%
oboe	70.30%	71.00%	35.90%	81.30%	68.80%	43.80%
vibraphone	69.30%	91.40%	85.70%	73.20%	90.10%	93.00%
bassoont	68.80%	66.70%	45.50%	61.10%	55.60%	27.80%
cello	67.00%	63.20%	63.50%	61.50%	62.50%	68.80%
saxophone	66.70%	51.70%	53.60%	46.70%	50.00%	50.00%

Table3 shows statistics of the precisions of the classifiers constructed with the selected features, all features, and the rest of the features after selection in the bottom level of the family tree for some instruments in the experiment. The overall accuracy of all the features was slightly better than that of the selected features. The computing time for List I, All, and List II is 7.33, 61.59, and 54.31 seconds respectively.

5 Conclusion and future work

A large number of attributes is generated in a table during fattening the features into a single value attributes for classical classifiers by statistical and other feature design methods. Some of the derived attributes may not significantly contribute to the classification models, or sometimes may distract the classification. In the light of the results from the experiments, we conclude that attributes have different degree of influence on the classification performance for different instrument families. The new temporal features related to harmonic peaks significantly improved the classification performance when added into the database with all other features. However, the new features were not suitable to replace the MPEG7 harmonic peak related features and Tristimulus parameters as the logistic studies shows. We also noticed that classifications at a higher level of granularity tended to use more features for correct prediction than those at the lower level. This may especially benefit a cooperative query answering system to choose suitable features for classifiers at different levels.

Acknowledgments. This work is supported by the National Science Foundation under grant IIS-0414815.

6 References

- [1] Balzano, G.J. (1986). What are musical pitch and timbre? *Music Perception - an interdisciplinary Journal*, 3, pp. 297-314.
- [2] Bay, M. and Beauchamp, J.W. (2006). Harmonic source separation using prestored spectra, *ICA 2006, LNCS 3889*, pp. 561 – 568, 2006.
- [3] Bregman, A.S. (1990). Auditory scene analysis, the perceptual organization of sound, MIT Press.
- [4] Cadoz, C. (1985). Timbre et causalite, Unpublished paper, Seminar on Timbre, Institute de Recherche et Coordination Acoustique / Musique, Paris, France, April 13-17.
- [5] Fujinaga, I., McMillan, K. (2000) Real time Recognition of Orchestral Instruments, *International Computer Music Conference*, pp. 141-143.
- [6] Kinoshita, T., Sakai, S., and Tanaka, H. (1999). Musical sound source identification based on frequency component adaptation, in *Proceedings of IJCAI Workshop on Computational Auditory Scene Analysis (IJCAI-CASA '99)*, Stockholm, Sweden, July-August, pp. 18–24.

- [7] Kitahara, T., Goto, M., Komatani, K., Ogata, T., and Okuno, H.G. (2007) Instrument Identification in Polyphonic Music: Feature Weighting to Minimize Influence of Sound Overlaps. *EURASIP Journal on Advances in Signal Processing*, Volume 2007, Article ID 51979.
- [8] Lindsay, A. T., and Herre, J. (2001) MPEG-7 and MPEG-7 Audio—An Overview, *J. Audio Eng. Soc.*, vol. 49, July/Aug, pp. 589–594.
- [9] Ozerov, A., Philippe, P., Gribonval, R., and Bimbot, F. (2005) One microphone singing voice separation using source adapted models, in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 90-93.
- [10] Pollard, H.F. and Jansson, E.V. (1982). A tristimulus Method for the specification of Musical Timbre. *Acustica*, 51, pp. 162-171
- [11] Cessie, S. le and Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression, *Applied Statistics*, vol. 41, no. 1, pp. 191-201.
- [12] Scheirer, E. and Slaney, M. (1997). Construction and Evaluation of a Robust Multi-feature Speech/Music Discriminator, in *Proc. IEEE int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*.
- [13] Smith, J.O. and Serra, X. (1987). PARSHL: An Analysis/Synthesis Program for Non Harmonic Sounds Based on a Sinusoidal Representation. In *Proc. Int. Computer Music Conf.*, Urbana-Champaign, Illinois, pp. 290-297.
- [14] Tzanetakis, G. and Cook, P. (2002). Musical Genre Classification of Audio Signals, *IEEE Trans. Speech and Audio Processing*, July, vol. 10, pp. 293–302.
- [15] Vincent, E. (2006). Musical source separation using time-frequency source priors, *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 91–98.
- [16] Wieczorkowska, A. (1999). Classification of musical instrument sounds using decision trees, in *the 8th International Symposium on Sound Engineering and Mastering, ISSEM'99*, pp. 225-230.
- [17] Wold, E., Blum, T., Keislar, D., and Wheaton, J. (1996). Content-Based Classification, Search and Retrieval of Audio, *IEEE Multimedia, Fall*, pp. 27–36.
- [18] Zhang, X., Marasek, K. and Ras, Z. W. (2007). Maximum Likelihood Study for Sound Pattern Separation and Recognition. In *Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering (MUE 2007)*. April 26–28, Seoul, Korea, pp. 807-812.
- [19] Zhang, X. and Ras, Z.W. (2006). Differentiated Harmonic Feature Analysis on Music Information Retrieval For Instrument Recognition, *proceeding of IEEE International Conference on Granular Computing*, May 10-12, Atlanta, Georgia, pp. 578-581.
- [20] Zhang, X. and Ras, Z.W. (2006). Sound Isolation by Harmonic Peak Partition For Music Instrument Recognition, *Fundamenta Informaticae Journal Special issue on Tilings and Cellular Automata*, IOS Press, pp. 612-628.
- [21] Zhang, X. and Ras, Z.W. (2007). Analysis of Sound Features for Music Timbre Recognition. Invited paper. In *Proceedings of the IEEE CS International Conference on Multimedia and Ubiquitous Engineering (MUE 2007)*. April 26–28, Seoul, Korea, pp. 3-8.
- [22] ISO/IEC JTC1/SC29/WG11 (2002). MPEG-7 Overview. <http://mpeg.telecomitalia.com/standards/mpeg-7/mpeg-7.htm>

Discovery of frequent graph patterns that consist of the vertices with the complex structures

Tsubasa Yamamoto¹, Tomonobu Ozaki² and Takenao Okawa¹

¹ Graduate School of Engineering, Kobe University

² Organization of Advanced Science and Technology, Kobe University

1-1 Rokkodai, Nada, Kobe 657-8501, JAPAN

{yamamoto@cs25.scitec., tozaki@cs., ohkawa@kobe-u.ac.jp

Abstract. In some applications, the data can be represented naturally in a special kind of graphs that consist of vertices holding a set of (structured) data such as item sets, sequences and so on. One of the typical examples is metabolic pathway. Metabolic pathway is represented in a graph structured data in which each vertex corresponds to an enzyme described by a set of various kinds of properties such as amino acid sequence, label and so on. We call this kind of complex graphs *multi-structured graphs*. In this paper, we propose an algorithm named FMG for mining frequent patterns in multi-structured graphs. In FMG, the external structure will be expanded by general graph mining algorithm, while the internal structure will be enumerated by some algorithm suitable for its structure. In addition, a pruning technique is introduced to exclude uninteresting patterns. The preliminary experimental results with real data show the effectiveness of the proposed algorithm.

1 Introduction

Graphs are widely used to represent complicated structures such as proteins, LSI circuits, hyperlinks in WWW, XML data and so on. Discovering frequent sub-graphs from a graph database is one of the most important problems of the graph mining[3]. In recent years, several efficient algorithms of graph mining have been developed[1, 6, 9, 10]. However, since the structure of data is becoming complex more and more, these algorithms might not be sufficient in some application domains. One typical example of such complex database is KEGG* , which is a database of metabolic pathways. Metabolism denotes total chemical reactions in the body of organisms. The chemical reaction is risen up by some enzyme and it translates a compound into another compound. Pathway is the large network of these reactions, so it can be regarded as a graph structured data. In addition, the vertices in the pathway consist of several types of data such as compounds, enzymes, genes and so on. Therefore, a metabolic pathway can be naturally represented in a graph that consist of vertices holding a set of (structured) data such as item sets, sequences and so on. Since most of graph mining algorithms

* <http://www.genome.ad.jp/kegg/>

consider the element of vertices as an item, they can not discover frequent sub-graphs in which sub-patterns of vertex elements are considered. Because such kind of complex graph data is expected to be going to rapidly increase, it is important to establish a flexible technique that can inclusively treat such kind of data.

In this paper, as one of the techniques to deal with such kind of complex graphs, we propose a new frequent graph mining algorithm named **FMG**. While we describe it in detail later, the target of FMG is special kind of graphs, called *multi-structured graphs*, that consist of vertices holding a set of (structured) data such as item sets, sequences and so on. Given a database of multi-structured graphs, FMG will discover frequent graph patterns that consist of vertices with several complex structures. In order to enumerate frequent patterns completely, while the external structures will be expanded in the manner of general graph mining algorithm, the internal structures, *i.e.* vertices, will be enumerated by some algorithm suitable for those structures. In addition, FMG employs several novel optimization techniques to exclude uninteresting patterns.

The rest of this paper is organized as follows. In section 2, we introduce some basic notations and define our data mining problem. In section 3, our frequent pattern miner FMG is proposed. Preliminary experimental results with pathways in KEGG are reported in section 4. After describing related work in section 5, we conclude this paper in section 6.

2 Preliminaries

A *multi-structured graph* G consists of a set of vertices $V(G)$ and a set of edges $E(G)$. Each vertex $v \in V(G)$ consists of a length n list of attributes of plural kinds of structured data. We denote the list of v as $list(v) = [elm_1^v, \dots, elm_n^v] \in [dom(A_1), \dots, dom(A_n)]$ where $dom(A_i)$ denotes the domain of structure of i th attribute A_i . We show an example of multi-structured graph in Fig. 1. For example, for $v_{13} \in V(G_1)$, $list(v_{13})$ is $[\{a, b, c\}, \langle AACCC \rangle]$ and the domain of the first and second attributes are item sets and sequences, respectively. A spanning tree of a graph is considered with depth first search for numbering the vertices. The first visited vertex is called root, while the last visited vertex is called rightmost vertex. The path from the root to the rightmost vertex in the spanning tree is called rightmost path. In G_1 , if we set v_{11} to the root, then the rightmost vertex is v_{14} and the rightmost path becomes $v_{11}-v_{13}-v_{14}$. We denote the edge $e_{ij} \in E(G)$ between i th and j th vertices. Edge labels are not considered in this paper.

For example, from an attribute whose domain is graph, we can extract several classes of pattern, such as paths, trees and graphs. So, as a bias, we have to give the class of pattern \mathcal{P}_{A_i} to be extracted from each attribute A_i . Given two patterns $p, q \in \mathcal{P}_{A_i}$, $p \preceq q$ denotes that p is more general than or equals to q . Given a vertex v with $list(v) = [elm_1^v, \dots, elm_n^v]$ and a list of patterns $lp = [p_1, \dots, p_n] \in [\mathcal{P}_{A_1}, \dots, \mathcal{P}_{A_n}]$, if all p_i cover its corresponding elm_i^v , then we say that P covers v and denote it as $lp \prec v$. Note that, we assume that each

combination of \mathcal{P}_{A_i} and $dom(A_i)$ gives the definitions of the *cover relation* and the *generality relation*.

For two multi-structured graphs G and G' , a subgraph isomorphism, denoted as $G \subseteq G'$, is an injective function $f : V(G) \Rightarrow V(G')$ such that (1) $\forall v \in V(G) v \prec f(v)$, and (2) $\forall e_{ij} \in E(G) f(e_{ij}) \in E(G')$. If there exists a subgraph isomorphism from G to G' , G is called subgraph of G' and G' is called a supergraph of G . Let $D = \{G_1, G_2, \dots, G_M\}$ be a database of multi-structured graphs. The *support* of a multi-structured subgraph pattern P , hereafter *graph pattern* in short, is defined as follows.

$$sup_D(P) = \frac{\sum_{G \in D} O_P(G)}{M} \quad \text{where } O_P(G) = \begin{cases} 1 & (P \subseteq G) \\ 0 & (\text{otherwise}) \end{cases}$$

Given a user defined threshold σ , a graph pattern P is called *frequent* in D if $sup_D(P) \geq \sigma$ holds. The problem discussed in this paper is stated formally as follows : Given a database D of multi-structured graphs and a positive number $\sigma (0 < \sigma \leq 1)$ called the *minimum support*, find all frequent graph patterns P such that $sup_D(P) \geq \sigma$.

3 Mining Frequent Multi-structured Subgraph Patterns

In this section, we propose a frequent multi-structured subgraph pattern miner named FMG. Throughout this section, we use the database shown in Fig. 1 as a running example for explaining the behavior of FMG.

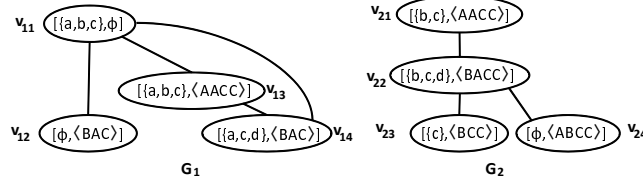


Fig. 1. Target database

FMG employs a kind of the pattern-growth approach [7, 9]. Initial patterns for FMG are multi-structured graphs with one vertex. That is to say, they are the set of all sub-patterns of the vertex element whose size is 1. For example, initial patterns taken from G_1 are $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$, $\langle A \rangle$, $\langle B \rangle$ and $\langle C \rangle$. FMG employs two kinds of procedures for expansion. The first one is for expanding internal structures, *i.e.* sub-patterns in a vertex, and the second one is for overall structure or topology of graph patterns. In this paper, the overall structure of graph pattern is referred to as external structure. By applying these two kinds

of expansions to the initial patterns repeatedly, all frequent graph patterns are to be enumerated.

In the following subsections, we describe internal and external expansions in detail. After that, some optimization techniques are introduced.

3.1 Expansion of Internal Structures

While we describe it in detail later, FMG employs general graph mining algorithm for expanding external structures. This constrains the enumeration strategy for internal structures, *i.e.* patterns in a vertex. The expansion of the internal structure in FMG is limited only in the rightmost vertex of the graph pattern. If not, a lot of duplicated patterns will be generated.

As described before, an internal structure of a multi-structured graph consists of a list of attributes of plural kinds of structured data such as item sets, sequences and so on. In order to enumerate *single* sub-patterns within each attribute efficiently, FMG employs existing algorithms. In addition to the single patterns in a vertex, we need to consider the combinations of patterns taken from different attributes. To avoid the duplications, the enumerations have to obey the ordering in the attribute list. Consider the case where an attribute A_0 is ahead of another attribute A_1 . In this case, given a single pattern $p \in \mathcal{P}_{A_1}$, then we avoid the generation of patterns $q \in [\mathcal{P}_{A_0}, \mathcal{P}_{A_1}]$ because it is against the order.

3.2 Expansion of External Structures

Since the external structure of multi-structured graph is a graph structure, FMG employs the enumeration strategy for the external structure in the manner of general graph mining algorithm[1, 9]. To put it concretely, the only vertices on the rightmost path are extended by adding an edge and, if necessary, a vertex.

In graph mining, it is important to check the isomorphism of a graph. In order to check the isomorphism and to identify the canonical form of a graph, the concept of code words for simple graphs is introduced in [1, 9]. The core idea underlying a canonical form is to construct a code word that uniquely identifies a graph up to isomorphism and symmetry. In FMG, the similar code words for the graph patterns are employed. The code word of a multi-structured graph pattern P is in the form as follows.

$$code(P) = l(list(v)) (i_d [-i_s] l(list(v)))^m$$

In this code word, i_s is the index of the source vertex, and i_d is the index of the destination vertex, respectively. The index of the source vertex is smaller than that of the destination vertex with respect to an edge. m is the total number of edges. $[-i_s]$ is a negative number. $l(list(v))$ is the string description of $list(v)$.

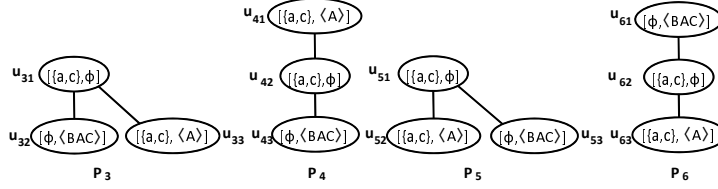


Fig. 2. Four isomorphic graph patterns

ϕ is defined to be the lexicographically smallest string. The code words of graph patterns P_3 – P_6 in Fig. 2 are shown below.

$$\begin{aligned}
 code(P_3) &= [\{a, c\}, \phi] \quad 2 - 1 [\phi, \langle BAC \rangle] \quad 3 - 1 [\{a, c\}, \langle A \rangle] \\
 code(P_4) &= [\{a, c\}, \langle A \rangle] \quad 2 - 1 [\{a, c\}, \phi] \quad 3 - 2 [\phi, \langle BAC \rangle] \\
 code(P_5) &= [\{a, c\}, \phi] \quad 2 - 1 [\{a, c\}, \langle A \rangle] \quad 3 - 1 [\phi, \langle BAC \rangle] \\
 code(P_6) &= [\phi, \langle BAC \rangle] \quad 2 - 1 [\{a, c\}, \phi] \quad 3 - 2 [\{a, c\}, \langle A \rangle]
 \end{aligned}$$

The canonical form of a graph pattern, or canonical pattern, is determined to be lexicographically smallest code word in the set of isomorphic patterns. In Fig. 2, $code(P_6)$ is the lexicographically smallest among the set of isomorphic graph patterns P_3 – P_6 . Thus, P_6 can be identified as a canonical pattern.

The canonical form of the simple graph pattern satisfies the anti-monotone property, *i.e.* no canonical pattern will be generated by expanding non-canonical patterns. Thus, we need not to expand non-canonical patterns. On the other hand, in case of mining multi-structured graphs, while no canonical pattern will be generated by expanding external structures, some canonical patterns can be generated from non-canonical patterns by expanding those internal structures. For example, in Fig. 3, P_7 is non-canonical pattern, while P_8 and P_9 are canonical patterns respectively. Since the expansion of the internal structures is limited in the rightmost vertex, P_8 is not generated from P_9 . On the other hand, P_8 is generated by the internal expansion of the rightmost vertex of P_7 .

Therefore, we cannot prune non-canonical patterns immediately. Non-canonical patterns have to be expanded in the internal structure until they become canonical.

3.3 Pruning based on the Internal Closedness

Since all sub-patterns of a frequent graph pattern are also frequent, it is easy to imagine that the number of frequent graph patterns grows exponentially. To

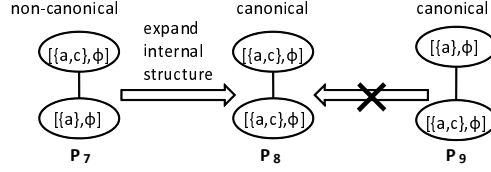


Fig. 3. The internal expansion from non-canonical pattern to canonical pattern

alleviate the explosion of frequent patterns, we introduce a pruning technique based on the *internal closedness*.

The set of all occurrences of P in G is denoted as $emb_G(P)$. Furthermore, we define the occurrence of P in a database D as $Emb^D(P) = \cup_{G \in D} emb_G(P)$.

Suppose a graph pattern P and another graph pattern P' which is obtained by expanding internal structure from P . If, for each occurrence $occ_p \in Emb^D(P)$, there is at least one corresponding occurrence $occ_{p'} \in Emb^D(P')$, i.e. $occ_{p'}$ overlaps occ_p completely, then we denote it as $OM_D(P, P')$. If $OM_D(P, P')$, $sup_D(P) = sup_D(P')$ holds by definition. Because P is a subgraph of P' and they have the same support value, P can be regarded as redundant. In addition, for each graph pattern Q obtained from P by expanding external structure, a graph pattern Q' such that $sup_D(Q) = sup_D(Q')$ can be obtained from Q by expanding external structure. Therefore, the expansion of external structure is not to be applied to P . In other words, P will be pruned. We call this pruning *internal closedness pruning*.

3.4 Introducing Monotone Constraints

The ability to handle monotone constraints will be incorporated into FMG. In this paper, we divide monotone constraints into two kinds. The first one is called *external* monotone constraints which give the restrictions for external structures. The requirement of the minimum number of vertices is an example of this kind of constraint. The second constraint is called *internal* monotone constraint. This constrains the patterns in a vertex, e.g. the minimum length of a sequence in a vertex.

As similar to the traditional top-down graph mining algorithms, it doesn't influence the generation of candidate patterns whether a certain pattern does not satisfy the given external monotone constraints in FMG. On the other hand, internal monotone constraint can be utilized for the effective pruning. In FMG, the expansion of the external structure does not be applied when the internal structure in the rightmost vertex does not satisfy the internal monotone constraints. In other words, the external structure will be expanded only after the internal structure in the rightmost vertex satisfies the internal monotone constraints. By employing the above enumeration strategy, we can avoid the generation of

graph patterns which have non-rightmost vertices that do not satisfy the internal monotone constraints. Note that, no graph pattern satisfying the constraints can be enumerated by expanding graph patterns having non-rightmost vertices that do not satisfy the internal monotone constraints.

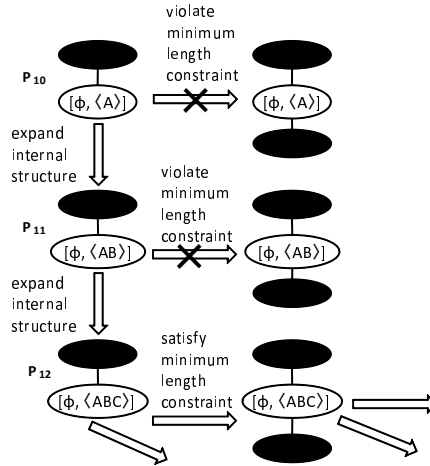


Fig. 4. Expanding with internal monotone constraint

We will explain the pruning based on internal monotone constraint with an example shown in Fig. 4. We assume that given internal monotone constraint is that the length of sequential pattern is more than 2. No pattern will be enumerated by expanding the external structure of P_{10} because the sequential pattern $\langle A \rangle$ violates the constraint. Thus, the only internal expansions will be applied to P_{10} and P_{11} will be obtained. Again, because P_{11} does not satisfy the internal monotone constraint, external expansion will not be applied. For P_{12} and its successors, both of internal and external expansions are allowed. As explained, introducing internal monotone constraint enables to reduce the number of patterns to be generated by expanding the external structure of the pattern.

On the other hand, suppose that we enumerate a pattern by expanding external structure of P_{10} . Then such pattern has a vertex that does not satisfy internal monotone constraint. In FMG, internal expansion will be applied to the rightmost vertex only. Therefore, no pattern satisfying the constraints can be obtained from such patterns. This shows that the pruning based on the internal monotone constraint does not affect the completeness.

3.5 Frequent Multi-structured Subgraph Pattern Miner

We show the pseudo code of FMG in Algorithm 1 and 2. In these algorithms, line 3–8 in `expand/6` corresponds to the pruning based on the internal monotone constraint. If a frequent graph pattern g violates the internal monotone constraint, the external expansion will not be applied to g . On the other hand, regardless of the external monotone constraint, the external expansion will be applied only to the canonical patterns which satisfy internal monotone constraint. The applicability of the internal closedness pruning is examined for the graph patterns which satisfy both of internal and external monotone constraints in line 10–21 in `expand/6`.

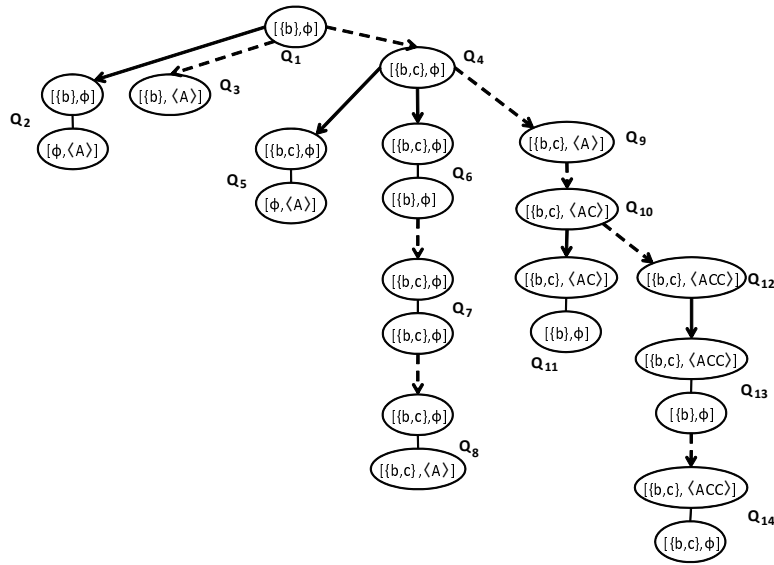


Fig. 5. A part of search space of FMG

In Fig. 5, we show a part of search space of FMG for a database D in Fig. 1 under the conditions that the minimum support σ is 2, minimum number of items in each vertex is 2, and minimum number of vertices is 2. The last two conditions are internal and external monotone constraints, respectively. In this figure, dashed lines denote the internal expansions, while solid lines indicate the external expansions. Q_1 is one of the initial patterns. Q_2 and Q_3 are not to be enumerated because of the pruning based on the internal monotone constraint. While Q_{11} can be obtained by applying external expansion of Q_{10} , FMG does not generate Q_{11} because the condition for the internal closedness pruning holds between Q_{10} and Q_{12} .

Algorithm 1 FMG($\sigma, \mathcal{D}, C_{in}, C_{ex}$)

1: **Input:** a multi-structured graph database \mathcal{D} , minimum support σ , internal monotone constraint C_{in} , external monotone constraint C_{ex} .
2: **Output:** a set of frequent subgraphs \mathcal{FG} that satisfies constraints.
3: $\mathcal{FG} := \emptyset$;
4: $\mathcal{SD} :=$ a set of initial patterns of \mathcal{D} ;
5: **for all** $g \in \mathcal{SD}$ **do**
6: call $\text{expand}(g, \sigma, \mathcal{D}, \mathcal{FG}, C_{in}, C_{ex})$;
7: **end for**
8: **return** \mathcal{FG} ;

Algorithm 2 $\text{expand}(g, \sigma, \mathcal{D}, \mathcal{FG}, C_{in}, C_{ex})$

1: **Input:** a multi-structured graph pattern g , minimum support σ , a multi-structured graph database \mathcal{D} , internal monotone constraint C_{in} , external monotone constraint C_{ex} .
2: **if** $\text{sup}_{\mathcal{D}}(g) \geq \sigma$ **then**
3: **if** g violates C_{in} **then**
4: $\mathcal{P}_{in} :=$ a set of patterns obtained by expanding the internal structure of g ;
5: **for all** $p \in \mathcal{P}_{in}$ **do**
6: call $\text{expand}(p, \sigma, \mathcal{D}, \mathcal{FG}, C_{in}, C_{ex})$;
7: **end for**
8: **else**
9: $\mathcal{P}_{in} :=$ a set of patterns obtained by expanding the internal structure of g ;
10: $bool := false$;
11: **if** g satisfies C_{ex} **then**
12: $bool := true$;
13: **for all** $p_{in} \in \mathcal{P}_{in}$ **do**
14: **if** $OM_{\mathcal{D}}(g, p_{in})$ **then**
15: $bool := false$;
16: **end if**
17: **end for**
18: **if** $bool \wedge g$ is canonical **then**
19: $\mathcal{FG} := \mathcal{FG} \cup \{g\}$;
20: **end if**
21: **end if**
22: $\mathcal{P}_{ex} := \emptyset$;
23: **if** $bool \wedge g$ is canonical **then**
24: $\mathcal{P}_{ex} :=$ a set of patterns obtained by expanding the external structure of g ;
25: **end if**
26: $\mathcal{P} := \mathcal{P}_{in} \cup \mathcal{P}_{ex}$;
27: **for all** $p \in \mathcal{P}$ **do**
28: call $\text{expand}(p, \sigma, \mathcal{D}, \mathcal{FG}, C_{in}, C_{ex})$;
29: **end for**
30: **end if**
31: **end if**

4 Experimental Results

In order to assess the effectiveness of the proposed algorithm, we implement the prototype of FMG in Java and conduct some preliminary experiments with real world data obtained from KEGG database. We mine several pathways for different organisms. The vertex of the pathway consists of enzymes, links to other pathway and compounds. Enzymes consist of an amino acid sequence and the enzyme number. Links are treated as an item. Compounds consist of a set of links to other pathway and its label. Those are treated as an item sets. As a result, we define the vertex element list as [item, number, item set, sequence]. We use the pathway database of RIBOFLAVIN METABOLISM which consists of pathways of 100 organisms. The average number of vertices is 21.03, and the average number of edges is 19.51.

The external and internal constraints used in the experiments are that the minimum number of vertices in the pattern is 5 and the minimum length of the sequence in a vertex is 7. All experiments were done on a PC(Intel Pentium IV, 2GHz) with 2GB of main memory running Windows XP. The experimental results are shown in Table 1.

Table 1. Experimental results

σ [%]	patterns	time[sec]	closed	constraint
80	4	1308	286	5202
50	189	2083	3489	12008
30	346	4183	13035	120432
20	2396	9636	25065	502342

While changing the minimum support value, we measured the number of patterns discovered (patterns in Table 1), execution time (time), number of patterns pruned by internal closedness pruning (closed) and number of patterns pruned by monotone constraints (constraint). Both the number of the patterns and execution time exponentially increase as the minimum support decreases. On the other hand, pruning based on the internal monotone constraint seems to be more promising than the internal closedness pruning.

We conducted another experiments by using FMG without internal monotone constraint. However, the result was not able to be obtained within 24 hours. This result shows the effectiveness of the pruning based on the internal monotone constraints.

We show an example of extracted pattern in Fig. 6. In this pattern, the vertex denoted as C00255 is riboflavin, the most important chemical compound in the riboflavin metabolism. At the same time, the sequential patterns of the enzyme are extracted. Thus, we can discover a special kind of patterns with FMG.

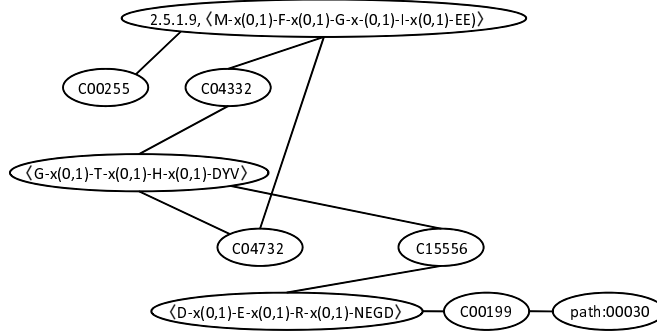


Fig. 6. An example of extracted pattern

5 Related Work

Recently, a lot of graph mining algorithms have been developed and applied to several real world problems[1, 5, 6, 9]. However, since most of these algorithms do not take the internal structure into account, they might fail to discover some meaningful patterns which will be found by FMG.

DAG Miner[2] differs from these traditional graph miners and it is one of the most related studies to FMG. DAG Miner extracts frequent patterns from directed acyclic graphs of which each vertex has an item sets. In DAG miner, all of frequent item sets will be found in advance, and then, by using these item sets, a restricted form of frequent DAG patterns called pyramid patterns will be mined. In contrast to DAG miner, FMG enumerates internal and external structures simultaneously.

On the other hand, mining algorithms for complex tree-structured patterns have been proposed. FAT-miner[4] is an algorithm for discovering frequent tree patterns that consists of vertices holding a set of attributes. pFreqT[8] mines frequent subtrees in which each vertex forms a sequence. While these two miners handle the tree-structured data, the target of FMG is complex graphs. In addition, FMG permits the combination of various structured patterns in the vertex.

6 Conclusion

In this paper, we focus on the problem of frequent pattern discovery in complex graph structured databases. By combining several algorithms for mining (structured) data such as item sets, sequences and graphs, we propose an algorithm FMG for mining frequent patterns in multi-structured graphs. Through the preliminary experiments, we show the effectiveness of the proposed algorithm.

As one of the future works, we plan to exploit FMG in order to extract more meaningful patterns.

References

1. Borgelt, C.: On Canonical Forms for Frequent Graph Mining. Proc. of the 3rd International Workshop on Mining Graphs. (2005) 1–12
2. Chen, Y. L., Kao, H. M., Ko.: Mining DAG Patterns from DAG Databases. Web-Age Information Management: 5th International Conference. Lecture Notes in Computer Science (2004) 579–588
3. De Raedt, L., Washio, T. and Kok, J. N. (eds.): Advances in Mining Graphs, Trees and Sequences. Volume 124 Frontiers in Artificial Intelligence and Applications. IOS Press (2005)
4. Knijf, J. D.: FAT-miner: Mining Frequent Attribute Trees. Symposium on Applied Computing archive Proc. of the 2007 ACM symposium on Applied computing Seoul, Korea (2007) 417–422
5. Koyuturk, M., Grama, A., Szpankowski, W.: An efficient algorithm for detecting frequent subgraphs in biological networks. Bioinformatics **20** (2004) 200–207
6. Nijssen, S., Kok, J. N.: A Quickstart in Frequent Structure Mining can make a Difference. Proc. of 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2004) 647–652
7. Pei, J., Han, J., Mortazavi-Asl, B., Pnto, H., Chen, Q., Dayal, U., Hsu, M.: PrefixSpan : Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. In Proc. of ICDE2001 IEEE Press (2001) 215–224
8. Sato, I., Nakagawa, H.: Semi-structure Mining Method for Text Mining with a Chunk-Based Dependency Structure. The 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining. (2007) 777–784
9. Yan, X., Han, J.: gSpan: Graph-Based Substructure Pattern Mining. Proc. of the 2nd IEEE International Conference on Data Mining. (2002) 721–724
10. Yan, X., Han, J.: CloseGraph: Mining Closed Frequent Graph Patterns. Proc. of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2003) 286–295

Learning to order basic components of structured complex objects

Donato Malerba and Michelangelo Ceci

Dipartimento di Informatica, Università degli Studi di Bari
via Orabona, 4 - 70126 Bari - Italy
{malerba, ceci}@di.uniba.it

Abstract. Determining the ordering of basic components of structured complex objects can be a crucial problem for several applications. In this paper, we investigate the problem of discovering partial or total orders among basic components by resorting to a data mining approach which acquires the domain specific knowledge from a set of training examples. The input of the learning method is the description of user-defined “chains” of basic components. The output is a logical theory that defines two predicates, *first/1* and *succ/2*, useful for consistently reconstructing all chains in new structured complex objects. The proposed method resorts to an ILP approach in order to exploit possible relations among basic components. We describe an application of the proposed method to learning the reading order of layout components extracted from document images. Determining the reading order enables the reconstruction of a single textual element from texts associated to multiple layout components and makes both information extraction and content-based retrieval of documents more effective. Experimental results show the effectiveness of the proposed method.

1 Introduction

Inductive learning has mostly concentrated on learning to classify. However, there are many applications in which it is desirable to order rather than classify instances [6, 8]. They include ordering information retrieval results on the basis of the relevance of retrieved documents with respect to a given query [4, 13] and ranking database query results [5]. Mathematically speaking, the problem of learning to order objects can be seen from two different perspectives [12]: 1) identifying a *total ordering* of a set of objects (ranking); 2) identifying several *partial orderings* of objects.

In the literature, several methods have been proposed for learning total or partial orderings [6, 8, 9, 4, 13, 12]. However, most of them assume that training data are represented in a single relational database table, such that each row (or tuple) represents an independent example (an object) and columns correspond to object properties. This single-table assumption seems to be quite restrictive for some applications. In particular, in the document image understanding domain, an ordering (specifically, a reading order) should be learned from the descriptions

of basic objects, the layout components, which are spatially correlated. The representation of the different relations between basic objects requires several tables which cannot be suitably handled under the single-table assumption.

In order to overcome limitations imposed by the single table assumption, in this paper we investigate the problem of learning to identify a partial ordering of basic components of complex objects by resorting to an inductive logic programming (ILP) approach [19, 14, 20]. More precisely, we resort to the application of the ILP system ATRE [16] to learn a logical theory which defines two predicates, *first/1* and *succ/2*, useful for consistently reconstructing partial orderings (in form of chains) in new structured complex objects.

The paper is organized as follows. The problem of learning to order objects is formally defined in Section 2. The machine learning system ATRE applied to the problem of learning the logical theory is introduced in Section 3, while the application of the learned theory in order to reconstruct a partial order relationship is reported in Section 4. Finally, the application to the document image processing domain is illustrated in Section 5, where experimental results are also reported and commented.

2 Problem Definition

In order to formalize the learning problem, some useful definition are necessary.

Definition 1. Partial Order [10]

Let A be a set of basic components of a complex object, a partial order P over A is a relation $P \in A \times A$ such that P is

1. reflexive $\forall s \in A \Rightarrow (s, s) \in P$
2. antisymmetric $\forall s_1, s_2 \in A: (s_1, s_2) \in P \wedge (s_2, s_1) \in P \Leftrightarrow s_1 = s_2$
3. transitive $\forall s_1, s_2, s_3 \in A: (s_1, s_2) \in P \wedge (s_2, s_3) \in P \Rightarrow (s_1, s_3) \in P$

When P satisfies the antisymmetric, the transitive and the irreflexive ($\forall s \in A \Rightarrow (s, s) \notin P$) properties, it is called a weak partial order over A .

Definition 2. Total Order

Let A be a set of basic components of a complex object, a partial order T over the set A is a total order iff $\forall s_1, s_2 \in A: (s_1, s_2) \in T \vee (s_2, s_1) \in T$

Definition 3. Complete chain, Chain reduction

Let A be a set of basic components of a complex object, let D be a weak partial order over A , let $B = \{a \in A | (\exists b \in A \text{ s.t. } (a, b) \in D \vee (b, a) \in D)\}$ be the subset of elements in A related to any element in A itself. If $D \cup \{(a, a) | a \in B\}$ is a total order over B then D is a complete chain over A .

Furthermore, $C = \{(a, b) \in D | \neg \exists c \in A \text{ s.t. } (a, c) \in D \wedge (c, b) \in D\}$ is the reduction of the chain D over A .

Example 1. Let $A = \{a, b, c, d, e\}$. $D = \{(a, b), (a, c), (a, d), (b, c), (b, d), (c, d)\}$ is a complete chain over A , then $C = \{(a, b), (b, c), (c, d)\}$ is its reduction.

Indeed, for our purposes it is equivalent to deal with complete chains or their reduction. Henceforth, for the sake of simplicity, the term *chain* will denote the reduction of a complete chain. By resorting to definitions above, it is possible to formalize the ordering induction problem as follows:

Given :

- A description $DesTO_i$ in the language L of the set of n training complex objects $TrainingObjs = \{TP_i \in \Pi | i = 1..n\}$ (where Π is the set of complex objects).
- A description $DesTC_i$ in the language L of the set TC_i of chains (over $TP_i \in TrainingObjs$) for each $TP_i \in TrainingObjs$.

Find :

An intensional definition T in the language L of a chain over a generic complex object $O \in \Pi$ such that T is complete and consistent with respect to all training chains descriptions $DesTC_i, i = 1..n$.

In this problem definition, we refer to the intensional definition T as a first order logic theory. The fact that T is complete and consistent with respect to all training chains descriptions can be formally described as follows:

Definition 4 (Completeness and Consistency).

Let:

- T be a logic theory describing chains instances expressed in the language L .
- E^+ be the set of positive examples for the chains instances.
($E^+ = \bigcup_{i=1..n} (\bigcup_{TC \in TC_i} TC)$).
- E^- be the set of negative examples for the chains instances.
($E^- = \bigcup_{i=1..n} (TP_i \times TP_i) / E^+$).
- $DesE^+$ be the description of E^+ in L .
- $DesE^-$ be the description of E^- in L .

then T is complete and consistent with respect to all training chains descriptions iff $T \models DesE^+ \wedge T \not\models DesE^-$

This formalization of the problem permits to represent and identify distinct orderings on the same complex object and allows to avoid to include in the ordering basic components that should not be included.

3 ATRE: The Learning System

ATRE is an ILP system that can learn recursive theories from examples. The learning problem solved by ATRE can be formulated as follows:

Given

- a set of *concepts* C_1, C_2, \dots, C_r to be learned
- a set of *observations* O described in a language L_O
- a *background theory* BK

- a *language* of hypotheses L_H
- a user’s *preference criterion* PC

Find

A logical theory T expressed in the language L_H and defining the concepts C_1, C_2, \dots, C_r , such that T is complete and consistent with respect to O and satisfies the preference criterion PC .

The *completeness* property holds when the theory T explains all observations in O of the r concepts C_1, C_2, \dots, C_r , while the *consistency* property holds when the theory T explains no counter-example in O of any concept C_i . The satisfaction of these properties guarantees the correctness of the induced theory, with respect to the given observations O . The selection of the “best” theory is made on the basis of an inductive bias embedded in some heuristic function or expressed by the user of the learning system (preference criterion). Details on the inductive learning strategy implemented in ATRE are reported in [16].

In the context of the ordering induction problem, we identified two concepts to be learned, namely *first*/1 and *succ*/2. The former refers to the the first basic component of a chain, while the latter refers to the relation *successor* between two basic components in a chain. By combining the two concepts it is possible to identify a partial ordering of basic components of a complex object.

As to the representation languages, literals can be of the two distinct forms:
 $f(t_1, \dots, t_n) = \text{Value}$ (simple literal); $f(t_1, \dots, t_n) \in \text{Range}$ (set literal),
 where f and g are function symbols called *descriptors*, t_i ’s are *terms* (constants or variables) and Range is a closed interval of possible values taken by f .

4 Application of Learned Rules

Once rules have been learned, they can be applied to new complex objects in order to generate a set of ground atoms such as: $\{first(0) = true, succ(0, 1) = true, \dots, succ(4, 3) = true, \dots\}$ which can be used to reconstruct chains of (possibly logically labelled) layout components. In our approach, we propose two different solutions: 1) Identification of multiple chains of basic components. 2) Identification of a single chain of basic components.

By applying rules learned by ATRE, it is possible to identify:

- A *directed* graph $G = \langle V, E \rangle$ ¹ where V is the set of nodes representing all the basic components of a complex object and edges represent the existence of a *succ* relation between two basic components, that is, $E = \{(b_1, b_2) \in V^2 | succ(b_1, b_2) = true\}$
- A list of initial nodes $I = \{b \in V | first(b) = true\}$

Both approaches make use of G and I in order to identify chains.

¹ G is not a direct acyclic graph (dag) since it could also contain cycles.

Multiple chains identification This approach aims at identifying a (possibly empty) set of chains over the set of basic components of a complex object. It is based on two steps, the first of which aims at identifying the heads (first elements) of the possible chains, that is the set

$$Heads = I \cup \{b_1 \in V \mid \exists b_2 \in V (b_1, b_2) \in E \wedge \forall b_0 \in V (b_0, b_1) \notin E\}$$

This set contains both nodes for which *first* is true and nodes which occur as a first argument in a true *succ* atom and do not occur as a second argument in any true *succ* atom.

Once the set *Heads* has been identified, it is necessary to reconstruct the distinct chains. Intuitively, each chain is the list of nodes forming a path in G which begins with a node in *Heads* and ends with a node without outgoing edges. Formally, an extracted chain $C \subseteq E$ is defined as follows:

$$C = \{(b_1, b_2), (b_2, b_3), \dots, (b_k, b_{k+1})\}, \text{ such that } b_1 \in Heads, \forall i = 1..k : (b_i, b_{i+1}) \in E \text{ and } \forall b \in V (b_{k+1}, b) \notin E.$$

In order to avoid cyclic paths, we impose that the same node cannot appear more than once in the same chain. The motivation for this constraint is that the same layout component is generally not read more than once by the reader.

Single chain identification The result of the second approach is a single chain. Following the proposal reported in [6], we aim at iteratively evaluating the most promising node to be appended to the resulting chain. More formally, let $PREF_G : V \times V \rightarrow \{0, 1\}$ be a preference function defined as follows:

$$PREF_G(b_1, b_2) = \begin{cases} 1 & \text{if } b_1 = b_2 \text{ or a path connecting } b_1 \text{ and } b_2 \text{ exists in } G \\ 0 & \text{otherwise} \end{cases}$$

Let $\mu : V \rightarrow \mathbb{N}$ be the function defined as follows:

$$\mu(L, G, I, b) = countConnections(L, G, I, b) + outGoing(V/L, b) - inComing(V/L, b)$$

where

- $G = \langle V, E \rangle$ is the ordered graph
- L is a list of *distinct* nodes in G
- $b \in V/L$ is a candidate node
- $countConnections(L, G, I, b) = |\{d \in L \cup I \mid PREF_G(d, b) = 1\}|$ counts the number of nodes in $L \cup I$ from which b is reachable.
- $outGoing(V/L, b) = |\{d \in V/L \mid PREF_G(b, d) = 1\}|$ counts the number of nodes in V/L reachable from b .
- $inComing(V/L, b) = |\{d \in V/L \mid PREF_G(d, b) = 1\}|$ counts the number of nodes in V/L from which b is reachable.

Algorithm 1 fully specifies the method for the single chain identification. The rationale is that at each step a node is added to the final chain. Such a node is that for which μ is the highest. Higher values of μ are given to nodes which can be reached from I , as well as from other nodes already added to the chain, and have a high (low) number of outgoing (incoming) paths to (from) nodes in V/L . Indeed, the algorithm returns an ordered list of nodes which could be straightforwardly transformed into a chain.

Algorithm 1 Single chain identification algorithm

```

1: findChain ( $G = \langle V, E \rangle, I$ ) Output: L: chain of nodes
2:  $L \leftarrow \emptyset$ ;
3: repeat
4:    $best\_mu \leftarrow -\infty$ ;
5:   for all  $b \in V/L$  do
6:      $cc \leftarrow countConnections(L, G, I, b)$ ;
7:      $inC \leftarrow incoming(V/L, b)$ ;  $outG \leftarrow outGoing(V/L, b)$ ;
8:     if  $((cc \neq 0) \text{ AND } (inC \neq 0) \text{ AND } (outG \neq 0))$  then
9:        $\mu \leftarrow cc + outG - inC$ ;
10:      if  $best\_mu < \mu$  then
11:         $best\_b \leftarrow b$ ;  $best\_mu \leftarrow \mu$ ;
12:      end if
13:    end if
14:  end for
15:  if  $(best\_mu <> -\infty)$  then
16:     $L.add(best\_b)$ ;
17:  end if
18: until  $best\_mu = -\infty$ 
19: return L

```

5 The Application: Learning Reading Order of Layout Components

In this paper, we investigate an application to the document image understanding problem. More specifically, we are interested in determining the reading order of layout components in each page of a multi-page document. Indeed, the spatial order in which the information appears in a paper document may have more to do with optimizing the print process than with reflecting the logical order of the information contained. Determining the correct reading order can be a crucial problem for several applications. By following the reading order recognized in a document image, it is possible to cluster together text regions labeled with the same logical label into the same textual component (e.g., “introduction”, “results”, “method” of a scientific paper). In this way, the reconstruction of a single textual component is supported and advanced techniques for text processing can be subsequently applied. For instance, information extraction methods

may be applied locally to reconstructed textual components. Moreover, retrieval of document images on the basis of their textual contents is more effectively supported.

Several works on reading order detection have already been reported in the literature [22][11][18][21][1] [3]. A common aspect of all methods is that they strongly depend on the specific domain and are not “reusable” when the classes of documents or the task at hand change. There is no work, to the best of our knowledge, that handles the reading order problem by resorting to machine learning techniques, which can generate the required knowledge from a set of training layout structures whose correct reading order has been provided by the user. In previous works on document image understanding, we investigated the application of machine learning techniques to several knowledge-based document image processing tasks, such as classification of blocks [2], automatic global layout analysis correction [17], classification of documents into a set of pre-defined classes and logical labelling. Following this mainstream of research, herein we consider the problem of learning the reading order.

In this context the limitations posed by the single table assumption are quite restrictive for at least two reasons. First, layout components cannot be realistically considered independent observations, because their spatial arrangement is mutually constrained by formatting rules typically used in document editing. Second, spatial relationships between a layout component and a variable number of other components in its neighborhood cannot be properly represented by a fixed number of attributes in a table. Even more so, the representation of properties of the other components in the neighborhood, because different layout components may have different properties (e.g., the property “brightness” is appropriate for half-tone images, but not for textual components). Since the single-table assumption limits the representation of relationships (spatial or non) between examples, it also prevents the discovery of this kind of pattern, which can be very useful in document image mining.

For these reasons, the ILP approach proposed in this paper seems to be appropriate for the task at hand. In ATRE, training observations are represented by ground multiple-head clauses [15], called *objects*, which have a conjunction of simple literals in the head. The head of an object contains positive and negative examples for the concepts to be learned, while the body contains the description of layout components on the basis of geometrical features (e.g. width, height) and topological relations (e.g. vertical and horizontal alignments) existing among blocks, the type of the content (e.g. text, horizontal line, image) and the logic type of a block (e.g. title or authors of a scientific paper). Terms of literals in objects can only be constants, where different constants represent distinct layout components within a page. An example of object description generated for the document page in Figure 1 is the following:

```

object('tpami17_1-13', [class(p) = tpami,
first(0) = true, first(1) = false, ...
succ(0, 1) = true, succ(1, 2) = true, ..., succ(7, 8) = true, succ(2, 10) = false, ...],
[part_of(p, 0) = true, ...,
```

$height(0) = 83, height(1) = 11, \dots, width(0) = 514, width(1) = 207, \dots,$
 $type_of(0) = text, \dots, type_of(11) = hor_line,$
 $title(0) = true, author(1) = true, affiliation(2) = true, \dots, undefined(16) = true,$
 $x_pos_centre(0) = 300, x_pos_centre(1) = 299, \dots,$
 $y_pos_centre(0) = 132, y_pos_centre(1) = 192, \dots,$
 $on_top(9, 0) = true, on_top(15, 0) = true, \dots, to_right(6, 8) = true, \dots$
 $alignment(16, 8) = only_right_col, alignment(17, 5) = only_left_col, \dots$
 $class(p) = tpami, page(p) = first$].

The constant p denotes the whole page while the remaining integer constants (0, 1, ..., 17) identify distinct layout components. In this example, the block number 0 corresponds to the first block to read ($first(0) = true$), it is a textual component ($type_of(0) = text$) and it is logically labelled as ‘title’ ($title(0) = true$). Block number 1 (immediately) follows block 0 in the reading order ($succ(0, 1) = true$), it is a textual component and it includes information on the authors of the paper ($author(1) = true$).

As explained in the previous sections, ATRE learns a logical theory T defining the concepts $first/1$ and $succ/2$ such that T is complete and consistent with respect to the examples. This means that it is necessary to represent both positive and negative examples and the representation of negative examples for the concept $succ/2$ poses some feasibility problems due to their quadratic growth. In order to reduce the number of negative examples, we resort to sampling techniques. In our case, we sampled negative examples by limiting their number to 1000% of the number of positive examples. This way, it is possible to simplify the learning stage and to have rules that are less specialized and avoid overfitting.

In order to evaluate the applicability of the proposed approach to reading order identification, we considered a set of multi-page articles published in an international journal. In particular, we considered twenty-four papers, published as either regular or short articles, in the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), in the January and February issues of 1996. Each paper is a multi-page document; therefore, we processed 211 document images. Each document page corresponds to a 24bit TIFF color image.

Initially, document images are pre-processed in order to segment them, perform layout analysis, identify the membership class and map the layout structure of each page into the logical structure. Training examples are then generated by manually specifying the reading order. In all, 211 positive examples and 3,263 negative examples for the concept $first/1$ and 1,418 positive examples and 15,518 negative examples for the concept $succ/2$ are generated.

We evaluated the performance of the proposed approach by means of a 6-fold cross-validation: the dataset is first divided into 6 *folds* of equal size and then, for every fold, the learner is trained on the remaining folds and tested on it.

For each learning problem, statistics on precision and recall of the learned logical theory are recorded. In order to evaluate the ordering returned by the proposed approach, we resort to metrics used in information retrieval to evaluate the returned ranking of results [7]. Herein we consider the metrics valid for partial orders evaluation.

In this study, we apply such distance measures to chains. In particular $FD = F(L|_{L_1}, L_1)$ and $SFD = F'(L|_{L_1}, L_1)$ are used in the evaluation of single chain identification. While $IFD = F(L, L_1, \dots, L_k)$ and $ISFD = F'(L, L_1, \dots, L_k)$ are used in the evaluation of multiple chains identification.

Results reported in Table 1 show that the system has a precision of about 65% and a recall greater than 75%. Moreover, there is no significant difference in terms of recall between the two concepts, while precision is higher for rules concerning the *succ* concept. This is mainly due to the specificity of rules learned for the concept *first*: rules learned for the concept *first* cover (on average) fewer positive examples than rules learned for the concept *succ* (statistics are not reported here because of space constraints). We can conclude that the concept *first* appears to be more complex to learn than the concept *succ*, probably because of the lower number of training examples (one per page).

Experimental results concerning the reconstruction of single/multiple chains are reported in Table 2. We recall that the lower the distance value the better the reconstruction of the original chain(s). By comparing results in terms of the *footrule distance* measure (IFD vs FD), we note that the reconstruction of multiple chains shows better results than the reconstruction of single chains. Indeed, this result does not take into account the length of the lists. When considering the length of the lists (ISFD vs. SFD) the situation is completely different and the reconstruction of single chains outperforms the reconstruction of multiple chains.

Some examples of rules learned by ATRE are reported below:

1. $first(X1) = true \leftarrow x_pos_centre(X1) \in [55..177],$
 $y_pos_centre(X1) \in [60..121], height(X1) \in [98..138].$
2. $first(X1) = true \leftarrow title(X1) = true, x_pos_centre(X1) \in [293..341],$
 $succ(X1, X2) = true.$
3. $succ(X2, X1) = true \leftarrow affiliation(X1) = true, author(X2) = true,$
 $height(X1) \in [45..124].$
4. $succ(X2, X1) = true \leftarrow alignment(X1, X3) = both_columns,$
 $on_top(X2, X3) = true, succ(X1, X3) = true, height(X1) \in [10..15]$

They show that ATRE is particularly indicated for the task at hand since it is able to identify dependencies among concepts to be learned or even recursion.

Concept	<i>first/1</i>		<i>succ/2</i>		<i>Overall</i>	
	Precision %	Recall%	Precision%	Recall%	Precision%	Recall%
FOLD1	75.00	50.00	76.90	64.10	76.60	61.80
FOLD2	66.70	63.20	74.10	65.20	73.00	64.90
FOLD3	74.30	78.80	81.00	66.10	80.10	67.40
FOLD4	69.40	71.40	67.80	56.30	68.00	58.20
FOLD5	66.70	66.70	78.40	68.70	76.80	68.40
FOLD6	71.00	61.10	79.40	62.90	78.20	62.60
AVG	70.52%	65.20%	76.27%	63.88%	75.45%	63.88%

Table 1. Precision and Recall results shown per concept to be learned

6 Conclusions

In this paper, we present an ILP approach to the problem of inducing a partial ordering between basic components of complex objects. The proposed solution is based on learning a logical theory which defines two predicates *first/1* and *succ/2*. The learned theory should be able to express dependencies between the two target predicates. For this reason we used the learning system ATRE which is able to learn mutually recursive predicate definitions. In the recognition phase, learned predicate definitions are used to reconstruct reading order chains identifying two different modalities: single vs. multiple chains identification.

The proposed approach can be applied to several application domains. In this paper, it has been applied to a real-world problem, namely detecting the reading order between layout components extracted from images of multi-page documents. Results prove that learned rules are quite accurate and that the reconstruction phase significantly depends on the application at hand. In particular, if the user is interested in reconstructing the actual chain (e.g. text reconstruction for rendering purposes), the best solution is in the identification of single chains. On the contrary, when the user is interested in recomposing text such that sequential components are correctly linked (e.g. in information extraction), the most promising solution is the identification of multiple chains.

For future work we intend to extend our empirical investigation to other application domains as well as to synthetically generated datasets.

Acknowledgments

This work is in partial fulfillment of the research objectives of the project “D.A.M.A.” (Document Acquisition, Management and Archiving).

References

1. M. Aiello, C. Monz, L. Todoran, and M. Worring. Document understanding for a broad class of documents. *International Journal on Document Analysis and Recognition-IJDAR*, 5(1):1–16, 2002.
2. O. Altamura, F. Esposito, and D. Malerba. Transforming paper documents into XML format with WISDOM++. *IJDAR*, 4(1):2–17, 2001.

Concept	Multiple chains		Single chain	
	AVG. IFD%	AVG. ISFD%	AVG. FD%	AVG. SFD%
FOLD1	13.18	21.12	47.33	10.17
FOLD2	10.98	18.51	46.32	8.13
FOLD3	1.31	26.91	47.32	17.63
FOLD4	1.32	24.00	49.96	14.51
FOLD5	0.90	22.50	49.31	10.60
FOLD6	0.90	27.65	54.38	12.97
AVG	4.76%	23.45%	49.10%	12.33%

Table 2. Reading order reconstruction results

3. T. M. Breuel. High performance document layout analysis. In *Proceedings of the 2003 Symposium on Document Image Understanding (SDIUT '03)*, 2003.
4. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *WWW7: Proceedings of the seventh international conference on World Wide Web 7*, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.
5. S. Chaudhuri, G. Das, V. Hristidis, and G. Weikum. Probabilistic information retrieval approach for ranking of database query results. *ACM Trans. Database Syst.*, 31(3):1134–1168, 2006.
6. W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *J. Artif. Intell. Res. (JAIR)*, 10:243–270, 1999.
7. C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001. ACM Press.
8. A. Gionis, T. Kujala, and H. Mannila. Fragments of order. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 129–136, New York, NY, USA, 2003. ACM Press.
9. A. Gionis, H. Mannila, K. Puolamäki, and A. Ukkonen. Algorithms for discovering bucket orders from data. In *Proceedings 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 561–566. ACM, 2006.
10. R. P. Grimaldi. *Discrete and Combinatorial Mathematics, an Applied Introduction*. Addison Wesley, terza edizione, 1994.
11. Y. Ishitani. Document transformation system from papers to xml data based on pivot xml document method. In *ICDAR '03: 7th International Conference on Document Analysis and Recognition*, page 250. IEEE Computer Society, 2003.
12. S. Kambhampati and J. Chen. Relative utility of EBG based plan reuse in partial ordering vs. total ordering planning. In *AAAI-93*, pages 514–519, Washington, D.C., USA, 1993. AAAI Press/MIT Press.
13. J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
14. N. Lavrač and S. Džeroski. *Inductive Logic Programming: techniques and applications*. Ellis Horwood, Chichester, 1994.
15. G. Levi and F. Sirovich. Generalized and/or graphs. *Artif. Intell.*, 7(3):243–259, 1976.
16. D. Malerba. Learning recursive theories in the normal ilp setting. *Fundamenta Informaticae*, 57(1):39–77, 2003.
17. D. Malerba, F. Esposito, O. Altamura, M. Ceci, and M. Berardi. Correcting the document layout: A machine learning approach. In *ICDAR*, page 97, 2003.
18. J.-L. Meunier. Optimized xy-cut for determining a page reading order. In *ICDAR '05: Proceedings of the Eighth International Conference on Document Analysis and Recognition*, pages 347–351, Washington, DC, USA, 2005. IEEE Computer Society.
19. S. Muggleton. *Inductive Logic Programming*. Academic Press, London, 1992.
20. S.-W. Nienhuys-Cheng and R. de Wolf. *Foundations of inductive logic programming*. Springer, Heidelberg, 1997.
21. S. L. Taylor, D. A. Dahl, M. Lipshutz, C. Weir, L. M. Norton, R. Nilson, and M. Linebarger. Integrated text and image understanding for document understanding. In *HLT '94: Proceedings of the workshop on Human Language Technology*, pages 421–426, 1994.
22. S. Tsujimoto and H. Asada. Understanding multi-articled documents. In *in Proceedings of the 10th International Conference on Pattern Recognition*, pages 551–556, 1990.

ARoGS: Action Rules Discovery based on Grabbing Strategy and LERS

Zbigniew W. Raś^{1,3} and Elżbieta Wyrzykowska²

¹ Univ. of North Carolina, Dept. of Comp. Science, Charlotte, N.C. 28223, USA;
e-mail: ras@uncc.edu

² Univ. of Information Technology and Management, ul. Newelska, Warsaw, Poland;
email: ewrzyko@wit.edu.pl

³ Polish-Japanese Institute of Information Technology, ul. Koszykowa 86, 02-008
Warsaw, Poland; e-mail: ras@pjwstk.edu.pl

Abstract. Action rules can be seen as logical terms describing knowledge about possible actions associated with objects which is hidden in a decision system. Classical strategy for discovering them from a database requires prior extraction of classification rules which next are evaluated pair by pair with a goal to build a strategy of action based on condition features in order to get a desired effect on a decision feature. An actionable strategy is represented as a term $r = [(\omega) \wedge (\alpha \rightarrow \beta)] \Rightarrow [\phi \rightarrow \psi]$, where ω , α , β , ϕ , and ψ are descriptions of objects or events. The term r states that when the fixed condition ω is satisfied and the changeable behavior $(\alpha \rightarrow \beta)$ occurs in objects represented as tuples from a database so does the expectation $(\phi \rightarrow \psi)$. This paper proposes a new strategy, called *ARoGS*, for constructing action rules with the main module which resembles *LERS* [5]. *ARoGS* system is more simple than *DEAR* and its time complexity is also lower.

1 Introduction

Finding useful rules is an important task of a knowledge discovery process. Most researchers focus on techniques for generating patterns from a data set such as classification rules, association rules...etc. They assume that it is user's responsibility to analyze these patterns in order to infer solutions for specific problems within a given domain. The classical knowledge discovery algorithms have the potential to identify enormous number of significant patterns from data. Therefore, people are overwhelmed by a large number of uninteresting patterns and it is very difficult for a human being to analyze them in order to form timely solutions. Therefore, a significant need exists for a new generation of techniques and tools with the ability to assist users in analyzing a large number of rules for a useful knowledge.

There are two aspects of interestingness of rules that have been studied in data mining literature, objective and subjective measures [1], [7]. Objective measures are data-driven and domain-independent. Generally, they evaluate the rules

based on their quality and similarity between them. Subjective measures, including unexpectedness, novelty and actionability, are user-driven and domain-dependent.

For example, classification rules found from a bank's data are very useful to describe who is a good client (whom to offer some additional services) and who is a bad client (whom to watch carefully to minimize the bank loses). However, if bank managers hope to improve their understanding of customers and seek specific actions to improve services, mere classification rules will not be convincing for them. Therefore, we can use the classification rules to build a strategy of action based on condition features in order to get a desired effect on a decision feature [8]. Going back to the bank example, the strategy of action would consist of modifying some condition features in order to improve their understanding of customers and then improve services.

Action rules, introduced in [8] and investigated further in [11], [13], [10], are constructed from certain pairs of classification rules. Interventions, defined in [4], are conceptually very similar to action rules.

The process of constructing action rules from pairs of classification rules is not only unnecessarily expensive but also gives too much freedom in constructing their classification parts. In [10] it was shown that action rules do not have to be built from pairs of classification rules and that single classification rules are sufficient to achieve the same goal. However, the paper only proposed a theoretical lattice-theory type framework without giving any detailed algorithm for action rules construction. In this paper we propose a very simple *LEERS*-type algorithm for constructing action rules from a single classification rule. *LEERS* is a classical example of a bottom-up strategy which constructs rules with a conditional part of the length $k+1$ after all rules with a conditional part of the length k have been constructed. Relations representing rules produced by *LEERS* are marked. System *ARoGS* assumes that *LEERS* is used to extract classification rules. This way *ARoGS* instead of verifying the validity of certain relations only has to check if these relations are marked by *LEERS*. The same, by using *LEERS* as the pre-processing module for *ARoGS*, the overall complexity of the algorithm will decrease.

2 Action Rules

In paper [8], the notion of an action rule was introduced. The main idea was to generate, from a database, special type of rules which basically form a hint to users showing a way to re-classify objects with respect to values of some distinguished attribute (called a decision attribute).

We start with a definition of an information system given in [6].

By an information system we mean a pair $S = (U, A)$, where:

- U is a nonempty, finite set of objects (object identifiers),

- A is a nonempty, finite set of attributes (partial functions) i.e. $a : U \rightarrow V_a$ for $a \in A$, where V_a is called the domain of a .

We often write (a, v) instead of v , assuming that $v \in V_a$. Information systems can be used to model decision tables. In any decision table together with the set of attributes a partition of that set into conditions and decisions is given. Additionally, we assume that the set of conditions is partitioned into stable and flexible conditions [8]. Attribute $a \in A$ is called stable for the set U , if its values assigned to objects from U can not change in time. Otherwise, it is called flexible. "Date of Birth" is an example of a stable attribute. "Interest rate" on any customer account is an example of a flexible attribute. For simplicity reason, we will consider decision tables with only one decision. We adopt the following definition of a decision table:

By a decision table we mean an information system $S = (U, A_1 \cup A_2 \cup \{d\})$, where $d \notin A_1 \cup A_2$ is a distinguished attribute called decision. Additionally, it is assumed that d is a total function. The elements of A_1 are called stable attributes, whereas the elements of $A_2 \cup \{d\}$ are called flexible. Our goal is to suggest changes in values of attributes in A_2 for some objects from U so the values of the attribute d for these objects may change as well. A formal expression describing such a property is called an action rule [8], [11].

<i>Stable</i>	<i>Flexible</i>	<i>Stable</i>	<i>Flexible</i>	<i>Stable</i>	<i>Flexible</i>	<i>Decision</i>
A	B	C	E	G	H	D
a_1	b_1	c_1	e_1			d_1
a_1	b_2			g_2	h_2	d_2

Table 1. Two classification rules extracted from S

To construct an action rule [11], let us assume that two classification rules, each one referring to a different decision class, are considered. We assume here that these two rules have to be equal on their stable attributes, if they are both defined on them. We use Table 1 to clarify the process of action rule construction. Here, "Stable" means stable attribute and "Flexible" means flexible one. In a standard representation, these two classification rules have a form:

$$r_1 = [(a_1 \wedge b_1 \wedge c_1 \wedge e_1) \rightarrow d_1], r_2 = [(a_1 \wedge b_2 \wedge g_2 \wedge h_2) \rightarrow d_2].$$

Assume now that object x supports rule r_1 which means that it is classified as d_1 . In order to reclassify x to a class d_2 , we need to change not only the value of B from b_1 to b_2 but also to assume that $G(x) = g_2$ and that the value H for object x has to be changed to h_2 . This is the meaning of the (r_1, r_2) -action rule defined by the expression below:

$$r = [(a_1 \wedge g_2 \wedge (B, b_1 \rightarrow b_2) \wedge (H, \rightarrow h_2)] \rightarrow (D, d_1 \rightarrow d_2)].$$

The term $[a_1 \wedge g_2]$ is called the header of the action rule. Assume now that by $Sup(t)$ we mean the number of tuples having property t . By the support of (r_1, r_2) -action rule (given above) we mean: $Sup[a_1 \wedge b_1 \wedge g_2 \wedge d_1]$. By the confidence $Conf(r)$ of (r_1, r_2) -action rule r (given above) we mean (see [11], [12]):

$$[Sup[a_1 \wedge b_1 \wedge g_2 \wedge d_1] / Sup[a_1 \wedge b_1 \wedge g_2]] \cdot [Sup[a_1 \wedge b_2 \wedge c_1 \wedge d_2] / Sup[a_1 \wedge b_2 \wedge c_1]].$$

Assume now that $S = (U, A_1 \cup A_2 \cup \{d\})$ is decision system, where $A_1 = \{a, b\}$ are stable attributes, $A_2 = \{c, e, f\}$ are flexible attributes, and d is the decision. For a generality reason, we take an incomplete decision system. It is represented as Table 2. Our goal is to re-classify objects in S from $(d, 2)$ to $(d, 1)$. Additionally, we assume that $Dom(a) = \{2, 3, 10\}$, $Dom(b) = \{2, 3, 4, 5\}$, and the null value is represented as -1 . We will follow optimistic approach in the process of action rules discovery, which means that the Null value is interpreted as the disjunction of all possible attribute values in the corresponding domain.

<i>Stable</i>	<i>Stable</i>	<i>Flexible</i>	<i>Flexible</i>	<i>Flexible</i>	<i>Decision</i>
<i>a</i>	<i>b</i>	<i>c</i>	<i>e</i>	<i>f</i>	<i>d</i>
2	-1	-1	7	8	1
2	5	4	6	8	1
-1	-1	4	9	4	2
10	4	5	8	7	2
2	2	5	-1	9	3
2	2	4	7	6	3
-1	2	4	7	-1	2
2	-1	-1	6	8	3
3	2	4	6	8	2
3	3	5	7	4	2
3	3	5	6	2	3
2	5	4	9	4	1

Table 2. Incomplete Decision System S

Now, we present the preprocessing step for action rules discovery. We start with our incomplete decision system S as the root of the Reduction Tree. The next step is to split S into sub-tables taking an attribute with the minimal number of distinct values as the splitting one. In our example, we chose attribute a . Because the 3rd and the 7th tuple in S contain null values in column a , we move them both to all three newly created sub-tables. This process is recursively continued for all stable attributes. Sub-tables corresponding to outgoing edges from the root node which are labelled by $a = 10$, $a = 3$ are removed because

they do not contain decision value 1. Any remaining node in the resulting tree can be used for discovering action rules. Clearly, if node n is used to construct action rules, then its children are not used for that purpose.

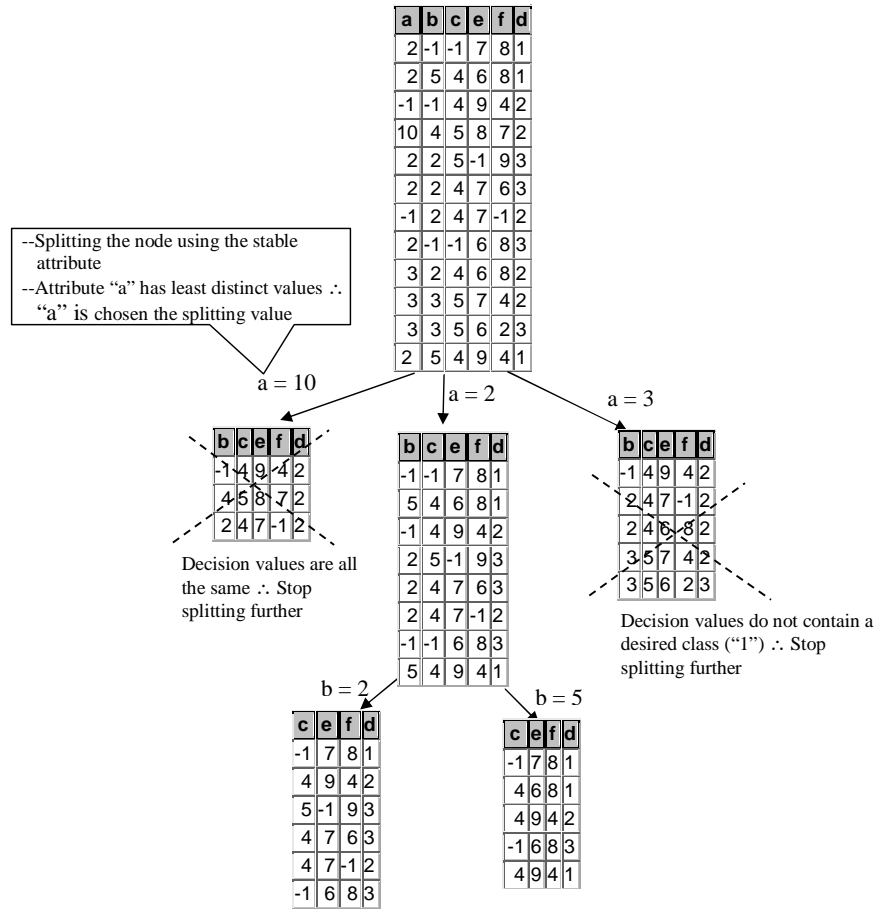


Fig. 1. Table Reduction Process

3 ARoGS: algorithm for discovering action rules

This section covers only complete information systems. For an incomplete information system, we can use *ERID* [2] to discover classification rules. Their syntax is the same as the syntax of rules discovered from a complete system.

Let us assume that $S = (U, A_1 \cup A_2 \cup \{d\})$ is a complete decision system, where $d \notin A_1 \cup A_2$ is a distinguished attribute called the decision. The elements of A_1 are stable conditions, whereas the elements of $A_2 \cup \{d\}$ are flexible. Assume that $d_1 \in V_d$ and $x \in U$. We say that x is a d_1 -object if $d(x) = d_1$. We also assume that $\{a_1, a_2, \dots, a_p\} \subseteq A_1$, $\{a_{p+1}, a_{p+2}, \dots, a_n\} = A_1 - \{a_1, a_2, \dots, a_p\}$, $\{b_1, b_2, \dots, b_q\} \subseteq A_2$, $a_{[i,j]}$ denotes a value of attribute a_i , $b_{[i,j]}$ denotes a value of attribute b_i , for any i, j and that

$$r = [[a_{[1,1]} \wedge a_{[2,1]} \wedge \dots \wedge a_{[p,1]} \wedge b_{[1,1]} \wedge b_{[2,1]} \wedge \dots \wedge b_{[q,1]}] \longrightarrow d_1]$$

is a classification rule extracted from S supporting some d_1 -objects in S . Class d_1 is a preferable class and our goal is to reclassify d_2 -objects into d_1 class, where $d_2 \in V_d$.

By an action rule schema $r_{[d_2 \longrightarrow d_1]}$ associated with r and the above reclassification task $(d, d_2 \longrightarrow d_1)$ we mean the following expression:

$$r_{[d_2 \longrightarrow d_1]} = [[a_{[1,1]} \wedge a_{[2,1]} \wedge \dots \wedge a_{[p,1]} \wedge (b_1, \longrightarrow b_{[1,1]}) \wedge (b_2, \longrightarrow b_{[2,1]}) \wedge \dots \wedge (b_q, \longrightarrow b_{[q,1]})] \longrightarrow (d, d_2 \longrightarrow d_1)]$$

In a similar way, by an action rule schema $r_{[\longrightarrow d_1]}$ associated with r and the reclassification task $(d, \longrightarrow d_1)$ we mean the following expression:

$$r_{[\longrightarrow d_1]} = [[a_{[1,1]} \wedge a_{[2,1]} \wedge \dots \wedge a_{[p,1]} \wedge (b_1, \longrightarrow b_{[1,1]}) \wedge (b_2, \longrightarrow b_{[2,1]}) \wedge \dots \wedge (b_q, \longrightarrow b_{[q,1]})] \longrightarrow (d, \longrightarrow d_1)]$$

The term $[a_{[1,1]} \wedge a_{[2,1]} \wedge \dots \wedge a_{[p,1]}]$, built from values of stable attributes, is called the header of $r_{[d_2 \longrightarrow d_1]}$ and its values can not be changed. It is denoted by $h[r_{[d_2 \longrightarrow d_1]}]$.

The support set of the action rule schema $r_{[d_2 \longrightarrow d_1]}$ is defined as $Sup(r_{[d_2 \longrightarrow d_1]}) = \{x \in U : (a_1(x) = a_{[1,1]}) \wedge (a_2(x) = a_{[2,1]}) \wedge \dots \wedge (a_p(x) = a_{[p,1]}) \wedge (d(x) = d_2)\}$.

Now, we outline *ARoGS* strategy for generating the set **AR** of Action Rules from the action rule schema $r_{[d_2 \longrightarrow d_1]}$.

Assume that:

- $V_{a_{p+1}} = \{a_{[p+1,1]}, a_{[p+1,2]}, \dots, a_{[p+1,J(1)]}\}$
- $V_{a_{p+2}} = \{a_{[p+2,1]}, a_{[p+2,2]}, \dots, a_{[p+2,J(2)]}\}$
- ...
- $V_{a_{p+n}} = \{a_{[p+n,1]}, a_{[p+n,2]}, \dots, a_{[p+n,J(n)]}\}$
- $V_{b_1} = \{b_{[1,1]}, b_{[1,2]}, \dots, b_{[1,J(n+1)]}\}$
- $V_{b_2} = \{b_{[2,1]}, b_{[2,2]}, \dots, b_{[2,J(n+2)]}\}$
-
- $V_{b_q} = \{b_{[q,1]}, b_{[q,2]}, \dots, b_{[q,J(n+q)]}\}$

To simplify the presentation of the algorithm we assume that:

- $c_k = a_{p+k}$ and $c_{[k,i]} = a_{[p+k,i]}$, for $1 \leq i \leq J(k)$, $1 \leq k \leq n$,
- $c_{n+m} = b_m$ and $c_{[n+m,i]} = b_{[m,i]}$, for $1 \leq i \leq J(n+m)$, $1 \leq m \leq q$.

For simplicity reason, we use $U_{[r,d_2]}$ to denote $Sup(r_{[d_2 \rightarrow d_1]})$. We assume that the term $c_{[i_1,j_1]} \wedge c_{[i_2,j_2]} \wedge \dots \wedge c_{[i_r,j_r]}$ is denoted by $[c_{(i_k,j_k)}]_{k \in \{1,2,\dots,r\}}$, where all i_1, i_2, \dots, i_r are distinct integers and $j_p \leq J(i_p)$, $1 \leq p \leq r$. Following **LEERS** notation [5], we also assume that t^* denotes the set of all objects in S having property t .

Algorithm $AR(r, d_2)$

```

i:=1
while  $i \leq n + q$  do
  begin
  j:=2; m:=1
  while  $j < J(i)$  do
    begin
    if  $[h[r_{[d_2 \rightarrow d_1]}] \wedge c_{(i,j)}]^* \subseteq U_{[r,d_2]} \wedge c_i \in A_2$  then
      begin
      mark $[c_{(i,j)}]$ ;
      output Action Rule
         $[[h[r_{[d_2 \rightarrow d_1]}] \wedge (c_i, c_{(i,j)} \rightarrow c_{(i,1)})] \rightarrow [d, d_2 \rightarrow d_1]]$ 
      end
    if  $[h[r_{[d_2 \rightarrow d_1]}] \wedge c_{(i,j)}]^* \subseteq U_{[r,d_2]} \wedge c_i \in A_1$  then
      begin
      mark $[c_{(i,j)}]$ ;
      output Action Rule
         $[[h[r_{[d_2 \rightarrow d_1]}] \wedge (c_i, c_{(i,j)})] \rightarrow [d, d_2 \rightarrow d_1]]$ 
      end
    j:=j+1
    end
  end
   $I_k := \{i_k\}$ ;
  (where  $i_k$  - index randomly chosen from  $\{2, 3, \dots, q + n\}$ ).
  for all  $j_k \leq J(i_k)$  do  $[c_{(i_k,j_k)}]_{i_k \in I_k} := c_{(i_k,j_k)}$ ;
  for all  $i, j$  such that both sets  $[c_{(i_k,j_k)}]_{i_k \in I_k}$ ,  $c_{(i,j)}$  are not marked and
   $i \in I_k$ 
  do
    begin
    if  $[[h[r_{[d_2 \rightarrow d_1]}] \wedge [c_{(i_k,j_k)}]_{i_k \in I_k} \wedge c_{(i,j)}]^* \subseteq U_{[r,d_2]} \wedge c_i \in A_2$  then
      begin
      mark  $[[c_{(i_k,j_k)}]_{i_k \in I_k} \wedge c_{(i,j)}]$ ;
      output Action Rule
         $[[h[r_{[d_2 \rightarrow d_1]}] \wedge [c_{(i_k,j_k)}]_{i_k \in I_k} \wedge (c_i, c_{(i,j)} \rightarrow c_{(i,1)})] \rightarrow [d, d_2 \rightarrow d_1]]$ 
      end
    if  $[[h[r_{[d_2 \rightarrow d_1]}] \wedge [c_{(i_k,j_k)}]_{i_k \in I_k} \wedge c_{(i,j)}]^* \subseteq U_{[r,d_2]} \wedge c_i \in A_1$  then
      begin
      mark  $[[c_{(i_k,j_k)}]_{i_k \in I_k} \wedge c_{(i,j)}]$ ;
      output Action Rule
         $[[h[r_{[d_2 \rightarrow d_1]}] \wedge [c_{(i_k,j_k)}]_{i_k \in I_k} \wedge (c_i, c_{(i,j)})] \rightarrow [d, d_2 \rightarrow d_1]]$ 
      end
    end
  end

```

```

end
else
begin
 $I_k := I_k \cup \{i\}; [c_{(i_k, j_k)}]_{i_k \in I_k} := [c_{(i_k, j_k)}]_{i_k \in I_k} \wedge c_{(i, j)}$ 
end

```

The complexity of *ARoGS* is lower than the complexity of *DEAR* system discovering action rules. The justification here is quite simple. *DEAR* system [11] groups classification rules into clusters of non-conflicting rules and then takes all possible pairs of classification rules within each cluster and tries to build action rules from them. *ARoGS* algorithm is treating each classification rule describing the target decision value as a seed and grabs other classification rules describing non-target decision values to form a cluster and then it builds decision rules automatically from them. Rules grabbed into a seed are only compared with that seed. So, the number of pairs of rules which have to be checked, in comparison to *DEAR* is greatly reduced. Another advantage of the current strategy is that the module generating action rules in *ARoGS* can just check if a mark is assigned by *LERS* to the relation $[h[r_{[d_2 \rightarrow d_1]}] \wedge c_{(i, j)}]^* \subseteq U_{[r, d_2]}$ instead of checking its validity.

The confidence of generated action rules depends on the number of remaining objects supporting them. Also, if $Conf(r) \neq 1$, then some objects in S satisfying the description $[a_{1,1} \wedge a_{2,1} \wedge \dots \wedge a_{p,1} \wedge b_{1,1} \wedge b_{2,1} \wedge \dots \wedge b_{q,1}]$ are classified as d_2 . According to the rule $r_{[d_2 \rightarrow d_1]}$ they should be classified as d_1 which means that the confidence of $r_{[d_2 \rightarrow d_1]}$ will get also decreased.

If $Sup(r_{[d_2 \rightarrow d_1]}) = \emptyset$, then $r_{[d_2 \rightarrow d_1]}$ can not be used for reclassification of objects. Similarly, $r_{[\rightarrow d_1]}$ can not be used for reclassification, if $Sup(r_{[d_2 \rightarrow d_1]}) = \emptyset$, for each d_2 where $d_2 \neq d_1$. From the point of view of actionability, such rules are not interesting.

Let $Sup(r_{[\rightarrow d_1]}) = \bigcup \{Sup(r_{[d_2 \rightarrow d_1]}) : (d_2 \in V_d) \wedge (d_2 \neq d_1)\}$ and $Sup(R_{[\rightarrow d_1]}) = \bigcup \{Sup(r_{[\rightarrow d_1]}) : r \in R(d_1)\}$, where $R(d_1)$ is the set of all classification rules extracted from S which are defining d_1 . So, $Sup(R_S) = \bigcup \{Sup(R_{[\rightarrow d_1]}) : d_1 \in V_d\}$ contains all objects in S which potentially can be reclassified.

Assume now that $U(d_1) = \{x \in U : d(x) \neq d_1\}$. Objects in the set $B(d_1) = [U(d_1) - Sup(R_{[\rightarrow d_1]})]$ can not be reclassified to the class d_1 and they are called d_1 -resistant.

Let $B(\neg d_1) = \bigcap \{B(d_i) : (d_i \in V_d) \wedge (d_i \neq d_1)\}$. Clearly $B(\neg d_1)$ represents the set of d_1 -objects which can not be reclassified. They are called d_1 -stable. Similarly, the set $B_d = \bigcup \{B(\neg d_i) : d_i \in V_d\}$ represents objects in U which can not be reclassified to any decision class. All these objects are called d -stable. In order to show how to find them, the notion of a confidence of an action rule is needed.

Let $r_{[d_2 \rightarrow d_1]}$, $r'_{[d_2 \rightarrow d_3]}$ are two action rules extracted from S . We say that these rules are p-equivalent (\simeq), if the condition given below holds for every $b_i \in A_1 \cup A_2$:

if r/b_i , r'/b_i are both defined, then $r/b_i = r'/b_i$.

Now, we explain how to calculate the confidence of $r_{[d_2 \rightarrow d_1]}$. Let us take d_2 -object $x \in Sup(r_{[d_2 \rightarrow d_1]})$. We say that x positively supports $r_{[d_2 \rightarrow d_1]}$ if there is no classification rule r' extracted from S and describing $d_3 \in V_d$, $d_3 \neq d_1$, which is p-equivalent to r , such that $x \in Sup(r'_{[d_2 \rightarrow d_3]})$. The corresponding subset of $Sup(r_{[d_2 \rightarrow d_1]})$ is denoted by $Sup^+(r_{[d_2 \rightarrow d_1]})$. Otherwise, we say that x negatively supports $r_{[d_2 \rightarrow d_1]}$. The corresponding subset of $Sup(r_{[d_2 \rightarrow d_1]})$ is denoted by $Sup^-(r_{[d_2 \rightarrow d_1]})$.

By the confidence of $r_{[d_2 \rightarrow d_1]}$ in S we mean:

$$Conf(r_{[d_2 \rightarrow d_1]}) = [card[Sup^+(r_{[d_2 \rightarrow d_1]})]/card[Sup(r_{[d_2 \rightarrow d_1]})]] \cdot conf(r).$$

Now, if we assume that $Sup^+(r_{[\rightarrow d_1]}) = \bigcup \{Sup^+(r_{[d_2 \rightarrow d_1]}) : (d_2 \in V_d) \wedge (d_2 \neq d_1)\}$, then by the confidence of $r_{[\rightarrow d_1]}$ in S we mean:

$$Conf(r_{[\rightarrow d_1]}) = [card[Sup^+(r_{[\rightarrow d_1]})]/card[Sup(r_{[\rightarrow d_1]})]] \cdot conf(r).$$

It can be easily shown that the definition of support and confidence of action rules given in Section 3 is equivalent to the definition of support and confidence given in Section 2.

4 An example

Let us assume that the decision system $S = (U, \{A_1 \cup A_2 \cup \{d\}\})$, where $U = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$, is represented by Table 3. A number of different methods can be used to extract rules in which the THEN part consists of the decision attribute d and the IF part consists of attributes belonging to $A_1 \cup A_2$. In our example, the set $A_1 = \{a, b, c\}$ contains stable attributes and $A_2 = \{e, f, g\}$ contains flexible attributes. System *LERS* [5] is used to extract classification rules.

We are interested in reclassifying d_2 -objects either to class d_1 or d_3 . Four certain classification rules describing d_1 , d_3 can be discovered by *LERS* in the decision system S . They are given below:

$$r1 = [b_1 \wedge c_1 \wedge f_2 \wedge g_1] \rightarrow d_1, r2 = [a_2 \wedge b_1 \wedge e_2 \wedge f_2] \rightarrow d_3, r3 = e_1 \rightarrow d_1, r4 = [b_1 \wedge g_2] \rightarrow d_3.$$

It can be shown that $R_{[d, \rightarrow d_1]} = \{r1, r3\}$ and $R_{[d, \rightarrow d_3]} = \{r2, r4\}$. Action rule schemas associated with $r1$, $r2$, $r3$, $r4$ and the reclassification task either $(d, d_2 \rightarrow d_1)$ or $(d, d_2 \rightarrow d_3)$ are:

$$\begin{aligned} r1_{[d_2 \rightarrow d_1]} &= [b_1 \wedge c_1 \wedge (f, \rightarrow f_2) \wedge (g, \rightarrow g_1)] \rightarrow (d, d_2 \rightarrow d_1), \\ r2_{[d_2 \rightarrow d_3]} &= [a_2 \wedge b_1 \wedge (e, \rightarrow e_2) \wedge (f, \rightarrow f_2)] \rightarrow (d, d_2 \rightarrow d_3), \\ r3_{[d_2 \rightarrow d_1]} &= [(e, \rightarrow e_1)] \rightarrow (d, d_2 \rightarrow d_1), \\ r4_{[d_2 \rightarrow d_3]} &= [b_1 \wedge (g, \rightarrow g_2)] \rightarrow (d, d_2 \rightarrow d_3). \end{aligned}$$

Table 3. Decision System

U	a	b	c	e	f	g	d
x_1	a_1	b_1	c_1	e_1	f_2	g_1	d_1
x_2	a_2	b_1	c_2	e_2	f_2	g_2	d_3
x_3	a_3	b_1	c_1	e_2	f_2	g_3	d_2
x_4	a_1	b_1	c_2	e_2	f_2	g_1	d_2
x_5	a_1	b_2	c_1	e_3	f_2	g_1	d_2
x_6	a_2	b_1	c_1	e_2	f_3	g_1	d_2
x_7	a_2	b_3	c_2	e_2	f_2	g_2	d_2
x_8	a_2	b_1	c_1	e_3	f_2	g_3	d_2

We can also show that $U_{[r1,d2]} = Sup(r1_{[d2 \rightarrow d1]}) = \{x_3, x_6, x_8\}$, $U_{[r2,d2]} = Sup(r2_{[d2 \rightarrow d3]}) = \{x_6, x_8\}$, $U_{[r3,d2]} = Sup(r3_{[d2 \rightarrow d1]}) = \{x_3, x_4, x_5, x_6, x_7, x_8\}$, $U_{[r4,d2]} = Sup(r4_{[d2 \rightarrow d3]}) = \{x_3, x_4, x_6, x_8\}$.

Following $AR(r1, d_2)$ algorithm we get: $[b_1 \wedge c_1 \wedge a_1]^* = \{x_1\} \not\subseteq U_{[r1,d2]}$, $[b_1 \wedge c_1 \wedge a_2]^* = \{x_6, x_8\} \subseteq U_{[r1,d2]}$, $[b_1 \wedge c_1 \wedge f_3]^* = \{x_6\} \subseteq U_{[r1,d2]}$, $[b_1 \wedge c_1 \wedge g_2]^* = \{x_2, x_7\} \not\subseteq U_{[r1,d2]}$, $[b_1 \wedge c_1 \wedge g_3]^* = \{x_3, x_8\} \subseteq U_{[r1,d2]}$. It will generate two action rules: $[b_1 \wedge c_1 \wedge (f, f_3 \rightarrow f_2) \wedge (g, \rightarrow g_1)] \rightarrow (d, d_2 \rightarrow d_1)$, $[b_1 \wedge c_1 \wedge (f, \rightarrow f_2) \wedge (g, g_3 \rightarrow g_1)] \rightarrow (d, d_2 \rightarrow d_1)$.

In a similar way we construct action rules from the remaining three action rule schemas.

The action rules discovery process, presented above, is called *ARoGS* and it consists of two main modules. For its further clarification, we use another example which has no connection with Table 3. The first module extracts all classification rules from S following *LERS strategy*. Assuming that d is the decision attribute and user is interested in re-classifying objects from its value d_1 to d_2 , we treat the rules defining d_1 as seeds and build clusters around them. For instance, if $A_1 = \{a, b, g\}$ are stable attributes, $A_2 = \{c, e, h\}$ are flexible in $S = (U, A_1 \cup A_2 \cup \{d\})$, and $r = [[a_1 \wedge b_1 \wedge c_1 \wedge e_1] \rightarrow d_1]$ is a classification rule in S , where $V_a = \{a_1, a_2, a_3\}$, $V_b = \{b_1, b_2, b_3\}$, $V_c = \{c_1, c_2, c_3\}$, $V_e = \{e_1, e_2, e_3\}$, $V_g = \{g_1, g_2, g_3\}$, $V_h = \{h_1, h_2, h_3\}$, then we remove from S all tuples containing values $a_2, a_3, b_2, b_3, c_1, e_1$ and we use again *LERS* to extract rules from this subsystem. Each rule defining d_2 is used jointly with r to construct an action rule. The validation step of each of the set-inclusion relations, in the second module of *ARoGS*, is replaced by checking if the corresponding term was marked by *LERS* in the first module of *ARoGS*.

5 Conclusion

System *ARoGS* differs from the tree-based strategies for action rules discovery (for instance from *DEAR* [11]) because clusters generated by its second module are formed around target classification rules. An action rule can be constructed

in *ARoGS* from two classification rules only if both of them belong to the same cluster and one of them is a target classification rule. So, the complexity of the second module of *ARoGS* is $O(k \cdot n)$, where n is the number of classification rules extracted by *LERS* and k is the number of clusters. The time complexity of the second module of *DEAR* is equal to $O(n \cdot n)$, where n is the same as in *ARoGS*. The first module of *ARoGS* is the same as the first module of *DEAR*, so their complexities are the same.

6 Acknowledgements

This research was partially supported by the National Science Foundation under grant IIS-0414815.

References

1. Adomavicius, G., Tuzhilin, A. (1997) Discovery of actionable patterns in databases: the action hierarchy approach, in **Proceedings of KDD'97 Conference**, Newport Beach, CA, AAAI Press
2. Dardzinska, A., Ras, Z.W., "Extracting Rules from Incomplete Decision Systems: System ERID", in Foundations and Novel Approaches in Data Mining, (Eds. T.Y. Lin, S. Ohsuga, C.J. Liao, X. Hu), Advances in Soft Computing, Vol. 9, Springer, 2006, 143-154
3. Hilderman, R.J., Hamilton, H.J. (2001) **Knowledge Discovery and Measures of Interest**, Kluwer
4. Greco, S., Matarazzo, B., Pappalardo, N., Slowiński, R. (2005) Measuring expected effects of interventions based on decision rules, in **Journal of Experimental and Theoretical Artificial Intelligence**, Taylor Francis, Vol. 17, No. 1-2
5. Grzymala-Busse, J. (1997) A new version of the rule induction system LERS, in **Fundamenta Informaticae**, Vol. 31, No. 1, 27-39
6. Pawlak, Z., (1991) Information systems - theoretical foundations, in **Information Systems Journal**, Vol. 6, 205-218
7. Silberschatz, A., Tuzhilin, A., (1995) On subjective measures of interestingness in knowledge discovery, in **Proceedings of KDD'95 Conference**, AAAI Press
8. Raś, Z., Wiczorkowska, A. (2000) Action Rules: how to increase profit of a company, in **Principles of Data Mining and Knowledge Discovery**, LNAI, No. 1910, Springer, 587-592
9. Raś, Z.W., Tzacheva, A., Tsay, L.-S. (2005) Action rules, in **Encyclopedia of Data Warehousing and Mining**, (Ed. J. Wang), Idea Group Inc., 1-5
10. Raś, Z.W., Dardzińska, A. (2006) Action rules discovery, a new simplified strategy, in **Foundations of Intelligent Systems**, F. Esposito et al. (Eds.), LNAI, No. 4203, Springer, 445-453
11. Tsay, L.-S., Raś, Z.W. (2005) Action rules discovery system DEAR, method and experiments, in **Journal of Experimental and Theoretical Artificial Intelligence**, Taylor & Francis, Vol. 17, No. 1-2, 119-128
12. Tsay, L.-S., Raś, Z.W. (2006) Action rules discovery system DEAR3, in **Foundations of Intelligent Systems**, LNAI, No. 4203, Springer, 483-492
13. Tzacheva, A., Raś, Z.W. (2005) Action rules mining, in **International Journal of Intelligent Systems**, Wiley, Vol. 20, No. 7, 719-736

Discovering Word Meanings Based on Frequent Termsets¹

Henryk Rybinski*, Marzena Kryszkiewicz*, Grzegorz Protaziuk*,
Aleksandra Kontkiewicz*, Katarzyna Marcinkowska, Alexandre Delteil**
*Warsaw University of Technology, **France Telecom R&D
{hrb,mkr, gprotazi@ii.pw.edu.pl}, {akontkie, kmarcink}@elka.pw.edu.pl,
alexandre.delteil@orange-ft.com

Abstract. Word meaning ambiguity has always been an important problem in information retrieval and extraction, as well as, text mining (documents clustering and classification). Knowledge discovery tasks such as automatic ontology building and maintenance would also profit from simple and efficient methods for discovering word meanings. The paper presents a novel text mining approach to discovering word meanings. The offered measures of their context are expressed by means of frequent termsets. The presented methods have been implemented with efficient data mining techniques. The approach is domain- and language-independent, although it requires applying part of speech tagger. The paper includes sample results obtained with the presented methods.

Keywords: association rules, frequent termsets, homonyms, polysemy

1. Introduction

Discovery of word sense is what lexicographers do by profession. Automating this process has a much shorter history. First attempts were made in the 1960s by Sparck Jones [15]. In modern text mining and information retrieval, knowing the exact sense (or meaning) of a word in a document or a query becomes an important issue. It is considered that knowledge of an actual meaning of a polysemous word can considerably improve the quality of the information retrieval process by means of retrieving more relevant documents or extracting relevant information from the documents.

As sufficiently large corpora and efficient computers have become available, several attempts to automate the process have been undertaken. An overview of methods that have used artificial intelligence, machine readable dictionaries (knowledge-based methods) or corpora (knowledge-poor methods) can be found in [7], [12].

Many other text processing and knowledge management tasks, such as automatic translation, information extraction or ontology comparison, also require the ability to detect an actual meaning of a word.

¹ The work has been performed within the project granted by France Telecom

The importance of polysemy detection becomes even clearer when one looks at the statistics for the English language. As stated in [10]:

“It has been estimated that more than 73% of words used in common English texts are polysemous, i.e., more than 73% of words have more than one sense.”

One should also note that many of those 73% words are highly polysemous. As stated in [9], an estimate of the average number of senses per word for all words found in common English texts is about 6.55. Therefore, there is a great need for methods able to distinguish polysemous words, as well as, their meanings.

Unfortunately, manually created dictionaries, which tend to ignore domain specific word senses, are not sufficient for the above mentioned applications. For this reason an amount of research concerning algorithms for automatic discovery of word senses from text corpora was carried out. An excellent survey of the history of ideas used in word sense disambiguation is provided by Ide and Veronis [7].

As for today, no satisfactory method has been described in the literature for discovering word meanings. However, a few methods that address this problem merit attention. Two main strategies for finding homonyms described in the literature are: Word Sense Discrimination (WSDc), i.e. the task of grouping the contexts that represent the senses of a polysemous word, and Word Sense Disambiguation (WSDa), i.e. the task of automatic labeling of polysemous words with a sense tag taken from a predefined set, [12]. There are a couple of different approaches to Word Sense Discrimination, as vector or cluster based, for example. The first one treats words as numeric vectors, whereas the second groups words with similar meaning into clusters. Most methods, however, use word distributional data and statistical analysis.

In the presented paper a novel text mining approach to discovery of homonyms is presented. The method applies the Word Sense Discrimination strategy, carrying out only shallow text analysis, which is restricted to the recognition of parts of speech, and is domain-independent. The method consists in determining atomic contexts of terms of interest by means of maximal frequent termsets, which then are used for determining discriminant contexts.

The rest of the paper is organized as follows: some relevant examples of WSDc systems are described in Section 2. Section 3 presents a deeper analysis of the problem as well as the basic ideas and concepts of the proposed method. The experiments and their results are presented in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

As mentioned above, during the past years, a number of various approaches for the discovery of word senses and meanings have been proposed [7]. In [8] and [11] the authors describe a set of problems that are faced, when applying them to real data. The main ones are: data sparseness, infrequent meanings, overlapping meanings and the use of part-of-speech tags. Some of them are described in more detail below.

The method Clustering by Committee, introduced in [11], is based on grammatical relations between words. It requires that all documents are POS tagged and, what is

also important, grammatically correct. Each word in this method is represented as a feature vector, where each feature corresponds to a context that the word in question appears in. After computing all vectors, words considered as similar are grouped into clusters. Each cluster represents a single meaning. Finally, for each word the most similar clusters, with respect to a given minimal similarity measure value, are determined. These closest clusters constitute all discovered meanings of a word. The authors claim, that this method discovers also less frequent senses of a word and avoids discovering duplicate senses.

Another group of methods used for discovering word senses are Bayesian networks [16]. These methods are based on the statistical analysis and can be used in the situations, when only small or insufficient amount of data is available, e.g. when larger amount of data would prevent the system from being able to handle it. A Bayesian network is built using local dependencies between words. In such a network, each node represents a word, and an edge represents a conditional probability of the connected nodes. Bayesian networks can be used to determine a probability of co-occurrence of a set of words.

Markov clustering method [2] concentrates on analyzing texts written in a natural language. This method allows discovering the meanings of words used by examined group within a particular range. It follows an assumption, that nouns which frequently appear together in a list, are also semantically related. A result of the Markov clustering is a graph, where nouns co-occurring with appropriate frequency are connected with an edge. The ambiguous words are those that are connected with disjunctive parts of such a graph.

The main concern indicated in many papers is the size of the repositories, on which the experiments are carried out. Too little data leads to data sparseness problems; too much data causes the memory lack problems. In addition, current part-of-speech taggers still occasionally fail to produce the correct results, thus leading to errors.

In most cases, the methods for discovering meanings require the use of a predefined set of meanings for a word, which is their main disadvantage. Usually, the methods require checking of found meanings against those contained by a semantic dictionary, thesaurus, ontology etc., which are not easily available. It is therefore desired to develop other methodologies that could also give satisfactory results for the cases of a limited lexical support. Such methodologies are known as "knowledge-poor".

The new method proposed in the paper satisfies this constraint, compromising both simplicity and sufficient efficiency. It also manages to handle the problem of overlapping meanings, as well as data sparseness and infrequent meanings.

3. Homonyms and Maximal Frequent Termset Contexts

Distinct meanings of homonyms are indicated by various distinct contexts in which they appear frequently. This assumption is based on the distributional hypothesis [5], where the underlying idea is that "*a word is characterized by the company it keeps*". The rule is very intuitive and therefore is applied to the proposed approaches. The problem is, however, how the notion of a context is defined. For example, it can be

understood as a set of words surrounding a target word frequently enough in documents, paragraphs, or sentences.

In our approach, a context is evaluated with respect to paragraphs as below.

Let dictionary $D = \{t_1, t_2, \dots, t_m\}$ be a set of distinct words, called terms. In general, any set of terms is called a *termset*. The set \mathcal{P} is a set of *paragraphs*, where each paragraph P is a set of terms such that $P \subseteq \mathcal{P}$.

Statistical significance of a termset X is called *support* and is denoted by $sup(X)$. $sup(X)$ is defined as the number (or percentage) of paragraphs in \mathcal{P} that contain X . Clearly, the supports of termsets that are supersets of termset X are not greater than $sup(X)$.

A termset is called *frequent* if it occurs in more than ε paragraphs in \mathcal{P} , where ε is a user-defined support threshold.

In the sequel, we will be interested in maximal frequent termsets, which we will denote by MF and define as the set of all maximal (in the sense of inclusion) termsets that are frequent.

Let x be a term. By $MF(x)$ we denote all maximal frequent termsets containing x . $MF(x)$ will be used for determining *atomic contexts* for x .

A termset X , $x \notin X$, is defined as an *atomic context* of term x if $\{x\} \cup X$ is an element of $MF(x)$. The set of all atomic contexts of x will be denoted by $AC(x)$:

$$AC(x) = \{X \setminus \{x\} \mid X \in MF(x)\}.$$

Clearly, for each two termsets Y, Z in $AC(x)$, Y differs from Z by at least one term and vice versa. In spite of this, Y and Z may indicate the same meaning of x in reality. Let y be a term in YZ and z be a term in ZY and $\{xyz\}$ be a termset the support of which is significantly less than the supports of Y and Z . This may suggest that Y and Z probably represent different meanings of x . Otherwise, Y and Z are likely to represent the same meaning of x . Please, note that $\{xyz\}$ plays a role of a *potential discriminant* for pairs of atomic contexts. The set of all potential discriminants for Y and Z in $AC(x)$ will be denoted by $\mathcal{D}(x, Y, Z)$:

$$\mathcal{D}(x, Y, Z) = \{\{xyz\} \mid y \in YZ \wedge z \in ZY\}.$$

Among the potential discriminants, those which are relatively infrequent are called *proper discriminants*. Formally, the set of all *proper discriminants* for Y and Z in $AC(x)$ will be denoted by $\mathcal{PD}(x, Y, Z)$, and defined as follows:

$$\mathcal{PD}(x, Y, Z) = \{X \in \mathcal{D}(x, Y, Z) \mid relSup(x, X, Y, Z) \leq \delta\}, \text{ where}$$

$$relSup(x, X, Y, Z) = sup(X) / \min(sup(xY), sup(xZ)), \text{ and}$$

δ is a user-defined threshold.

In the sequel, $relSup(x, X, Y, Z)$ is called a *relative support of discriminant X* for term x with respect to atomic contexts Y and Z .

Our proposal of determining the groups of contexts representing separate meanings of x is based on the introduced notion of proper discriminants for pairs of atomic contexts.

Atomic contexts Y and Z in $AC(x)$ are called *discriminable* if there is at least one proper discriminant in $\mathcal{PD}(x, Y, Z)$. Otherwise, Y and Z are called *indiscriminable*.

A *sense-discriminant context* $SDC(x, X)$ of x for termset X in $AC(x)$ is defined as the family of those termsets in $AC(x)$ that are indiscriminable with X ; that is,

$$SDC(x, X) = \{Y \in AC(x) \mid \mathcal{PD}(x, X, Y) = \emptyset\}.$$

Clearly, $X \in SDC(x, X)$. Please, note that sense-discriminant contexts of x for Y and Z , where $Y \neq Z$, may overlap, and in particular, may be equal.

The family of all distinct sense-discriminant contexts will be denoted by $\mathcal{FSDC}(x)$:

$$\mathcal{FSDC}(x) = \{SDC(x, X) \mid X \in AC(x)\}.$$

Please, note that $|\mathcal{FSDC}(x)| \leq |AC(x)|$.

A given term x is defined as a *homonym candidate* if the cardinality of $\mathcal{FSDC}(x)$ is greater than 1. Final decision on homonymy is given to the user. Let us also note that the more overlapping are distinct sense-discriminant contexts, the more difficult is reusing the contexts for the meaning recognition in the mining procedures.

In order to illustrate the introduced concepts below we consider an example, which is based on an experimentally prepared set of documents.

Example 1. A special repository has been built based on Google search engine and the AMI-SME software [3]. For the term *apple*, the following maximal frequent termsets have been found in the repository (see Table 1):

Table 1. Maximal frequent termsets for term *apple* ($MF(apple)$)

Maximal frequent termset	Support
<i>apple species breeding</i>	1100
<i>apple eat</i>	14000
<i>apple genetics</i>	1980
<i>apple gmo</i>	1480
<i>apple,botanics</i>	2510
<i>apple cake</i>	3600
<i>apple genome</i>	1800
<i>apple motherboard</i>	2500
<i>apple mouse pad</i>	2500
<i>apple iphone</i>	6000

From this table we can evaluate the set of atomic contents, which is:

$$AC(apple) = \{\{species, breeding\}, \{eat\}, \{genetics\}, \{gmo\}, \{botanics\}, \{cake\}, \{genome\}, \{motherboard\}, \{mouse, pad\}, \{iphone\}\}.$$

Given the threshold $\delta = 0.2$, we can evaluate the set of proper discriminants from the set of the potential discriminants $\{apple, z, y\}$. For instance, for the atomic contexts Y and Z such that $Y = \{species, breeding\}$ and $Z = \{motherboard\}$, we have the following potential discriminants $\mathcal{D}(apple, Y, Z) = \{\{apple, species,$

motherboard}, {*apple, breeding, motherboard*}. Given the supports of the potential discriminants (see Table 2): $sup(\{apple, species, motherboard\}) = 209$ and $sup(\{apple, breeding, motherboard\}) = 78$, we can calculate their relative supports with respect to the supports of *Y* and *Z* as follows:

- $relSup(apple, \{apple, species, motherboard\}, Y, Z) = \frac{sup(\{apple, species, motherboard\})}{\min(sup(\{apple, species, breeding\}), sup(apple, motherboard))} = \frac{209}{\min(1100, 2500)} = 0.19$;
- $relSup(apple, \{apple, breeding, motherboard\}, Y, Z) = \frac{sup(\{apple, breeding, motherboard\})}{\min(sup(\{apple, species, breeding\}), sup(apple, motherboard))} = \frac{78}{\min(1100, 2500)} = 0.07$.

Both discriminants {*apple, species, motherboard*} and {*apple, breeding, motherboard*} have been found as proper, since their relative supports are below the threshold δ . The set of all proper discriminants is provided in Table 2.

Table 2. Proper discriminants

<i>PD</i>	<i>Support</i>	<i>Relative support</i>
<i>Apple,species,motherboard</i>	209	209/1100=0.19
<i>Apple,breeding,motherboard</i>	78	78/1100=0.07
<i>Apple,eat motherboard</i>	307	307/2500=0.12
<i>Apple,botanics motherboard</i>	68	68/2500=0.03
<i>Apple,botanics,mouse</i>	482	482/2500=0.19
<i>Apple genome motherboard</i>	74	74/1800=0.04
<i>Apple genome pad</i>	192	192/1800=0.10
<i>Apple gmo motherboard</i>	25	25/1800=0.10
<i>Apple, breeding, iphone</i>	0	0/1100=0.00
<i>Apple, botanics, iphone</i>	209	209/2500=0.08

For this set of proper discriminants we have found two sense-discriminant contexts for the term *apple*:

$\mathcal{FSDC}(apple) = \{\{apple, species, breeding, eat, genetics, gmo, botanics, cake\}, \{apple, motherboard, mouse, pad, iphone\}\}$.

□

4. Homonyms discovering procedure

Text preprocessing phase

The text preprocessing phase has been performed with the text mining platform TOM, which has been implemented at Warsaw University of Technology with the aim to support ontology maintenance and building with text mining techniques [13], [14]. The TOM platform provides options for defining the granularity of the text mining

process. In particular, TOM allows viewing the whole corpus as a set of documents, paragraphs or sentences. For the experiments of discovering homonyms, we set the granularity at the paragraph level. Thus, the first step was to generate a set of paragraphs from all the documents in the repository. It means that the context of particular terms is restricted to the paragraphs.

Then we have used the Hepple tagger [6] for part-of-speech tagging of the words in the sentences. In TOM, the tagger is a wrapped code of the Gate part of speech processing resource [4].

Conversion into “transactional database”

The next step is to convert the text corpora into “transactional database” (in terms of [1]). So, every unit of text (i.e. every paragraph) is converted into a transaction containing a set of terms identifiers. The usage of terms identifiers instead of terms themselves leads to speeding up all the data mining operations. Further on, the identifiers of all terms that do not have required minimum support ϵ are deleted from all the transactions.

Finding maximal termsets

Having reduced transaction representation of the text, we find the maximal frequent termsets $MF(x)$ for all terms x from the list of terms of interest by means of any efficient data mining algorithm discovering maximal frequent itemsets [17].

Identification of the sense-discriminant contexts

Having $MF(x)$ for each term x , we calculate the atomic contexts $AC(x)$ by simple removal of x from each termset in $MF(x)$. For later use, for each atomic context X in $AC(x)$, we store the support of the maximal frequent termset xX , from which X was derived. Now, we create potential discriminants for all the pairs of atomic contexts. Then, from the set of potential discriminants, we search for proper discriminants. This requires the calculation of the relative supports of the potential discriminants based on the supports of termsets xX , where $X \in AC(x)$, and the supports of the potential discriminants themselves. While the supports of termsets xX , where $X \in AC(x)$, are already known, the supports of the potential discriminants must be calculated. This is achieved with one scan over the transaction dataset [1]. Eventually, we calculate all the distinct sense-discriminant contexts for x . If the number of the distinct sense-discriminant contexts for the term x is higher than 1, we classify x as a polysemous term with the meanings determined by the contexts.

4. Experiments

Some experiments were conducted to measure the efficiency of the proposed algorithms. In general the found homonymous words can be classified into three different groups:

- a) Same meaning: berlin
[ontology, european, intelligence, th, proceedings, workshop, eds, conference, learning, germany, artificial, august, ecai, hastings, wiemer-]
[springer-, verlag]

- b) Different meaning: background
[domain, theory]
[web, pages]
- c) Different use: results
[table, taxonomy, phase, similarity, measures]
[information, users, queries]

In our case, we are interested in discovering words which belong to the second and third group. The tests were done using different minimal support values and part-of-speech tags. So far better results were obtained with using smaller values for the support threshold. As the aim is to find homonymous nouns only those were considered. As for the contexts we took into account nouns, adjectives and verbs.

First, the tests have been performed on a repository composed of scientific papers dealing with ontologies, semantic web and text mining. The results have been rather modest, which can be justified by the fact that the scientific texts in a limited domain use homogeneous and well defined terminology. Nevertheless, for the term *background* two sense discriminant contexts have been found {*domain, theory*} and {*web, pages*}

Another group of tests have been performed on the Reuters repository. In this case the polysemous words are provided in Table 3.

Table 3 Results for the Reuters repository (the threshold set to 11 occurrences)

President	chinese jiang zemin
	hussein iraqi saddam
Ltd	pictures seagram unit universal
	corp murdoch news
	percent shares worth
York	consolidated exchange stock trading
	city gate prices
France	budget currency deficits
	brazil cup world

One can see an erroneous candidate *president*, which results from a shallow semantic analysis of the text. With a deeper processing of the text (in this case replacing the proper names by tokens) the candidate would be rejected.

The term *york* has two meanings – one with the stock, and another one with the city. And with the third example, the term *france* has assigned the context referring to the country (economic situation) and the national football team.

The last group of tests referred to a repository composed from a set of queries in the AMI-SME system. These tests were oriented towards an experiment in checking the relevance of the idea and was used to find out meanings for the term *apple* (Example 1) and the term *turkey*. For both terms the method has proven its efficiency. Now, we plan performing more tests in order to determine the relevance and precision of the method.

5. Conclusions and Future Work

As said above, the tests were carried out on three repositories. As the results shown, the repository of scientific papers was not well suited for such experiments. The experiments with the Reuters repository have shown strong dependency on the size of the repository. With the too small repository, the results are rather poor. With the extended repository, we were able to find more polysemous words, on the other hand though, with too large repository we had problems with the data mining algorithms. To this end we have decided to define *a priori* a list of terms for which the experiment is run.

The results obtained so far show that the method presented above is able to distinguish homonymous words correctly. A human expert can easily interpret the contexts generated by the algorithms. Obviously there are among candidate erroneous terms, one can also expect that there are some undiscovered. This results from the following problems: (a) wrong part-of-speech tags assigned to the words by the POSTagger; (b) the values set for the minimal support, which are too high, but on the other hand, if lowered, caused memory problems; (c) the repository is too small and does not cover proper discriminants.

References

1. Agrawal R., Srikant R.: "Fast Algorithms for Mining Association Rules". Proc. of the 20th Int'l Conf. on Very Large Databases, Santiago, 1994. Morgan Kaufmann 487-499
2. Dorow B. and Widdows D., "Discovering Corpus-Specific Word Senses", EACL 2003, pp.79-82 Budapest, Hungary
3. Gawrysiak P., Rybinski H., Skonieczny Ł., Wiech P., "AMI-SME: An Exploratory Approach to Knowledge Retrieval for SME's," 3rd Int'l Conf. on Autonomic and Autonomous Systems (ICAS'07), 2007
4. General Architecture for Text Engineering. <http://gate.ac.uk/projects.html>
5. Harris, Z. (1954). Distributional structure. *Word*, 10(23): 146-162.
6. Hepple, M.: Independence and commitment: Assumptions for rapid training and execution of rule-based POS taggers. In: Proc. of the 38th Annual Meeting of the Association for Computational Linguistics (ACL-2000) (2000)
7. Ide N, Veronis J., Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics* 24(1):1-40. Special Issue on Word Sense Disambiguation.
8. Lin D., "Automatic Retrieval and Clustering of Similar Words", in Proceedings of the 17th international conference on Computational linguistics - Volume 2, Canada, 1998
9. Mihalcea, R., and Moldovan, D., 2001, "Automatic Generation of a Coarse Grained WordNet," in Proceedings of NAACL Workshop on WordNet and Other Lexical Resources, Pittsburgh, PA, June 2001.
10. Miller, G., Chadorow, M., Landes, S., Leacock, C., and Thomas, R. G., 1994, "Using a semantic concordance for sense identification." Proc. of the ARPA Human Language Technology Workshop, pp. 240-243.
11. Pantel, P. and Lin, D. 2002. Discovering word senses from text. In Proceedings of the Eighth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining (Edmonton, Alberta, Canada, July 23 - 26, 2002). KDD '02. ACM Press, New York, NY, 613-619.

12. Portnoy D., Unsupervised Discovery of the Multiple Senses of Words and Their Parts of Speech, The School of Engineering and Applied Science of The George Washington University, September 30, 2006
13. Protaziuk G., Kryszkiewicz M, Rybinski H, Delteil A., “Discovering Compound and Proper Nouns”, Rough Sets and Intelligent Systems Paradigms, Springer, LNAI 4595, pp. 505-515, 2007
14. Rybinski H., Kryszkiewicz M., Protaziuk G., Jakubowski A., Delteil A. Discovering Synonyms based on Frequent Termsets”, Rough Sets and Intelligent Systems Paradigms, Springer, LNAI 4595, pp. 516-525, 2007
15. Sparck Jones K., Synonymy and Semantic Classification, Edinburgh University Press, ISBN 0-85224-517-3, 1986 (originally published in 1964).
16. Young C. Park, Young S. Han and Key-Sun Choi “Automatic Thesaurus Construction using Bayesian Networks”, in the Proceedings of the fourth international conference on Information and knowledge management, United States, 1995
17. Zaki Mohammed J., Gouda Karam, Efficiently Mining Maximal Frequent Itemsets in 1st IEEE International Conference on Data Mining , San Jose, November 2001

Feature Selection: Near Set Approach

James F. Peters¹, Sheela Ramanna^{2*}

¹Department of Electrical and Computer Engineering,
University of Manitoba
Winnipeg, Manitoba R3T 5V6 Canada
jfpeters@ee.umanitoba.ca

² Department of Applied Computer Science,
University of Winnipeg,
Winnipeg, Manitoba R3B 2E9 Canada
s.ramanna@uwinnipeg.ca

Abstract. The problem considered in this paper is the description of objects that are, in some sense, qualitatively near each other and the selection of features useful in classifying near objects. The term *qualitatively near* is used here to mean closeness of descriptions or distinctive characteristics of objects. The solution to this twofold problem is inspired by the work of Zdzisław Pawlak during the early 1980s on the classification of objects. In working toward a solution of the problem of the classification of perceptual objects, this article introduces a near set approach to feature selection. Consideration of the nearness of objects has recently led to the introduction of what are known as near sets, an optimist's view of the approximation of sets of objects that are more or less near each other. Near set theory started with the introduction of collections of partitions (families of neighbourhoods), which provide a basis for a feature selection method based on the information content of the partitions of a set of sample objects. A byproduct of the proposed approach is a feature filtering method that eliminates features that are less useful in the classification of objects. This contribution of this article is the introduction of a near set approach to feature selection.

Keywords: Description, entropy, feature selection, filter, information content, nearness, near set, perception, probe function.

1 Introduction

The problem considered in this paper is the classification of perceptual objects that are qualitatively but not necessarily spatially near each other. The term *qualitatively near* is used here to mean closeness of descriptions or distinctive characteristics of objects. The solution to this problem is inspired by the work

* We gratefully acknowledge the very helpful comments, insights and corrections concerning this paper by Andrzej Skowron and the anonymous reviewers. This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) grants 185986 and 194376.

of Zdzisław Pawlak during the early 1980s on the classification of objects [12], elaborated in [13, 17], and a view of perception that is on the level of classes instead of individual objects [10]. In working toward a solution of the problem of the classification of perceptual objects, this article introduces a nearness description principle. An *object description* is defined by means of a vector of probe function values associated with an object (see, *e.g.*, [11]). Each probe function ϕ_i represents a feature of an object of interest. Sample objects are near each other if, and only if the objects have similar descriptions.

Ultimately, there is interest in selecting the probe functions [11] that lead to descriptions of objects that are *minimally* near each other. This is an essential idea in the near set approach [7, 14, 16, 19] and differs markedly from the minimum description length (MDL) proposed in 1983 by Jorma Rissanen [23]. MDL depends on the identification of possible data models and possible probability models. By contrast, NDP deals with a set X that is the domain of a description used to identify similar objects. The term *similar* is used here to denote the presence of objects that have descriptions that match each other to some degree.

The near set approach leads to partitions of ensembles of sample objects with measurable information content and an approach to feature selection. The proposed feature selection method considers combinations of n probe functions taken r at a time in searching for those combinations of probe functions that lead to partitions of a set of objects that has the highest information content. It is Shannon's measure of the information content [8, 26] of an outcome that provides a basis for the proposed feature selection method. In this work, feature selection results from a filtering method that eliminates those features that have little chance to be useful in the analysis of sample data. The proposed approach does not depend on the joint probability of finding a feature value for an input vectors that belong to the same class as in [6]. In addition, the proposed approach to measuring the information content of families of neighbourhoods differs from the rough set-based form of entropy in [25]. Unlike the dominance-relation rough set approach [5], the near set approach does not depend on preferential ordering of value sets of functions representing object features. The contribution of this article is the introduction of a near set approach to feature selection.

This article has the following organization. A brief introduction to the notation and basic approach to object description is given in Sect. 2. A brief introduction to nearness approximation spaces is given in Sect. 4. A nearness description principle is introduced in Sect. 3. A near set-based feature selection method is introduced in Sect. 5.

2 Object Description

Objects are known by their descriptions. An *object description* is defined by means of a tuple of function values $\phi(x)$ associated with an object $x \in X$ (see (1)). The important thing to notice is the choice of functions $\phi_i \in B$ used to describe an object of interest.

$$\textbf{Object Description : } \phi(x) = (\phi_1(x), \phi_2(x), \dots, \phi_i(x), \dots, \phi_L(x)). \quad (1)$$

Table 1. Description Symbols

Symbol	Interpretation
\mathfrak{R}	Set of real numbers,
\mathcal{O}	Set of perceptual objects,
X	$X \subseteq \mathcal{O}$, set of sample objects,
x	$x \in \mathcal{O}$, sample object,
\mathcal{F}	A set of functions representing object features,
B	$B \subseteq \mathcal{F}$,
ϕ	$\phi : \mathcal{O} \rightarrow \mathfrak{R}^L$, object description,
L	L is a description length,
i	$i \leq L$,
ϕ_i	$\phi_i \in B$, where $\phi_i : \mathcal{O} \rightarrow \mathfrak{R}$, probe function,
$\phi(x)$	$\phi(x) = (\phi_1(x), \phi_2(x), \phi_3(x), \dots, \phi_i(x), \dots, \phi_L(x))$.

The intuition underlying a description $\phi(x)$ is a recording of measurements from sensors, where each sensor is modelled by a function ϕ_i . Assume that $B \subseteq \mathcal{F}$ (see Table 1) is a given set of functions representing features of sample objects $X \subseteq \mathcal{O}$. Let $\phi_i \in B$, where $\phi_i : \mathcal{O} \rightarrow \mathfrak{R}$. The value of $\phi_i(x)$ is a measurement associated with a feature of an object $x \in X$. The function ϕ_i is called a *probe* [11]. In combination, the functions representing object features provide a basis for an *object description* $\phi : \mathcal{O} \rightarrow \mathfrak{R}^L$, a vector containing measurements (returned values) associated with each functional value $\phi_i(x)$ in (1), where the description length $|\phi| = L$.

2.1 Sample Behaviour Description

Table 2. Sample ethogram

x_i	s	a	$p(s, a)$	r	d
x_0	0	1	0.1	0.75	1
x_1	0	2	0.1	0.75	0
x_2	1	2	0.05	0.1	0
x_3	1	3	0.056	0.1	1
x_4	0	1	0.03	0.75	1
x_5	0	2	0.02	0.75	0
x_6	1	2	0.01	0.9	1
x_7	1	3	0.025	0.9	0

By way of illustration, consider the description of the behaviour observable in biological organisms. For example, a behaviour can be represented by a tuple

$$(s, a, p(s, a), r)$$

where $s, a, p(s, a), r$ denote organism functions representing state, action, action preference in a state, and reward for an action, respectively. A reward r is ob-

served in state s and results from an action a performed in the previous state. The preferred action a in state s is calculated using

$$p(s, a) \leftarrow p(s, a) + \beta\delta(s, a),$$

where β is the actor's learning rate and $\delta(r, s)$ is used to evaluate the quality of action a (see [20]). In combination, tuples of behaviour function values form the following description of an object x relative to its observed behaviour:

$$\textbf{Organism Behaviour} : \phi(x) = (s(x), a(x), r(x), V(s(x))).$$

Table 2 exhibits a sample observed behaviours of an organism.

3 Nearness of Objects

Approximate, a [L. *approximat-us* to draw near to.]

A. *adj.*

1. Very near, in position or in character; closely situated; nearly resembling.

–Oxford English Dictionary, 1933.

Table 3. Set, Relation, Probe Function Symbols

Symbol	Interpretation
\sim_B	$\{(x, x') \mid f(x) = f(x') \forall f \in B\}$, indiscernibility relation,
$[x]_B$	$[x]_B = \{x' \in X \mid x' \sim_B x\}$, elementary granule (class),
\mathcal{O} / \sim_B	$\mathcal{O} / \sim_B = \{[x]_B \mid x \in \mathcal{O}\}$, quotient set,
ξ_B	Partition $\xi_B = \mathcal{O} / \sim_B$,
$\Delta\phi_i$	$\Delta\phi_i = \phi_i(x') - \phi_i(x)$, probe function difference,

Sample objects $X \subseteq \mathcal{O}$ are near each other if, and only if the objects have similar descriptions. Recall that each description ϕ^1 defines a description of an object (see Table 1). Then let $\Delta\phi_i$ denote

$$\Delta\phi_i = \phi_i(x') - \phi_i(x),$$

where $x, x' \in \mathcal{O}$ (see Table 3). The difference $\Delta\phi$ leads to a definition of the indiscernibility relation \sim_B introduced by Zdzisław Pawlak [12] (see Def 1).

¹ In a more general setting that includes data mining, ϕ_i would be defined to allow for non-numerical values, *i.e.*, let $\phi_i : X \rightarrow V$, where V is the *value set* for the range of ϕ_i [22]. The more general definition of $\phi_i \in \mathcal{F}$ is also better in setting forth the algebra and logic of near sets after the manner of the algebra and logic of rough sets [1, 3, 4, 22]. Real-valued probe functions are used in object descriptions in this article because we have science and engineering applications of near sets in mind.

Definition 1 Indiscernibility Relation

Let $x, x' \in \mathcal{O}, B \subseteq \mathcal{F}$.

$\sim_B = \{(x, x') \in \mathcal{O} \times \mathcal{O} \mid \forall \phi_i \in B. \Delta\phi_i = 0\}$, where $i \leq$ description length $|\phi|$.

Definition 2 Nearness Description Principle (NDP)

Let $B \subseteq \mathcal{F}$ be a set of functions representing features of objects $x, x' \in \mathcal{O}$. Objects x, x' are minimally near each other if, and only if there exists $\phi_i \in B$ such that $x \sim_{\{\phi_i\}} x'$, i.e., $\Delta\phi_i = 0$.

In effect, objects x, x' are considered *minimally near* each other whenever there is at least one probe function $\phi_i \in B$ so that $\phi_i(x) = \phi_i(x')$. A *probe function* can be thought of as a model for a sensor (see, e.g., [11, 17]). Then ϕ_i constitutes a minimum description of the objects x, x' that makes it possible for us to assert that x, x' are near each other. Ultimately, there is interest in identifying the probe functions that lead to partitions with the highest information content. The nearness description principle (NDP) differs markedly from minimum description length (MDL) proposed by Jorma Rissanen [23]. MDL deals with a set $X = \{x_i \mid i = 1, \dots\}$ of possible data models and a set Θ of possible probability models. By contrast, NDP deals with a set X that is the domain of a description $\phi : X \rightarrow \mathfrak{R}^L$ and the discovery of at least one probe function $\phi_i(x)$ in a particular description $\phi(x)$ used to identify similar objects in X . The term *similar* is used here to denote the presence of objects $x, x' \in X$ and at least one ϕ_i in object description ϕ , where $x \sim_{\phi_i} x'$. In that case, objects x, x' are said to be similar. This leads to a feature selection method, where one considers combinations of n probe functions r in searching for those combinations of probe functions that lead to partitions with the highest information content.

Observation 1 Near Objects in a Class

Let $\xi_B = \mathcal{O} / \sim_B$ denote a partition of \mathcal{O} . Let $[x]_B \in \xi_B$ denote an equivalence class. Assume $x, x' \in [x]_B$. From Table 3 and Def. 1, we know that for each $\phi_i \in B, \Delta\phi_i = 0$. Hence, from Def. 2, x, x' are near objects.

Theorem 1 The objects in a class $[x]_B \in \xi_B$ are near objects.

Proof. The nearness of objects in a class in ξ_B follows from Obs. 1. □

4 Nearness Approximation Spaces

The original generalized approximation space (GAS) model [27] has recently been extended as a result of recent work on nearness of objects (see, e.g., [7, 14, 16, 18, 19, 28, 29]). A nearness approximation space (NAS) is a tuple

$$NAS = (\mathcal{O}, \mathcal{F}, \sim_{B_r}, N_r, \nu_{N_r}),$$

defined using set of perceived objects \mathcal{O} , set of probe functions \mathcal{F} representing object features, indiscernibility relation \sim_{B_r} , defined relative to $B_r \subseteq B \subseteq \mathcal{F}$,

Table 4. Nearness Approximation Space Symbols

Symbol	Interpretation
B	$B \subseteq \mathcal{F}$
B_r	$r \leq B $ probe functions in B ,
\sim_{B_r}	Indiscernibility relation defined using B_r ,
$[x]_{B_r}$	$[x]_{B_r} = \{x' \in \mathcal{O} \mid x \sim_{B_r} x'\}$, equivalence class,
\mathcal{O} / \sim_{B_r}	$\mathcal{O} / \sim_{B_r} = \{[x]_{B_r} \mid x \in \mathcal{O}\}$, quotient set,
ξ_{B_r}	Partition $\xi_{\mathcal{O}, B_r} = \mathcal{O} / \sim_{B_r}$,
ϕ_i	Probe function $\phi_i \in \mathcal{F}$,
r	$\binom{ B }{r}$, <i>i.e.</i> , $ B $ functions $\phi_i \in \mathcal{F}$ taken r at a time,
$N_r(B)$	$N_r(B) = \{\xi_{B_r} \mid B_r \subseteq B\}$, set of partitions,
ν_{N_r}	$\nu_{N_r} : \mathcal{P}(\mathcal{O}) \times \mathcal{P}(\mathcal{O}) \longrightarrow [0, 1]$, overlap function,
$N_r(B)_* X$	$N_r(B)_* X = \bigcup_{x: [x]_{B_r} \subseteq X} [x]_{B_r}$, lower approximation,
$N_r(B)^* X$	$N_r(B)^* X = \bigcup_{x: [x]_{B_r} \cap X \neq \emptyset} [x]_{B_r}$, upper approximation,
$Bnd_{N_r(B)}(X)$	$N_r(B)^* X \setminus N_r(B)_* X = \{x \in N_r(B)^* X \mid x \notin N_r(B)_* X\}$.

family of neighbourhoods N_r , and neighbourhood overlap function ν_{N_r} . The relation \sim_{B_r} is the usual indiscernibility relation from rough set theory restricted to a subset $B_r \subseteq B$. The subscript r denotes the cardinality of the restricted subset B_r , where we consider $\binom{|B|}{r}$, *i.e.*, $|B|$ functions $\phi_i \in \mathcal{F}$ taken r at a time to define the relation \sim_{B_r} . This relation defines a partition of \mathcal{O} into non-empty, pairwise disjoint subsets that are equivalence classes denoted by $[x]_{B_r}$, where

$$[x]_{B_r} = \{x' \in \mathcal{O} \mid x \sim_{B_r} x'\}.$$

These classes form a new set called the quotient set \mathcal{O} / \sim_{B_r} , where

$$\mathcal{O} / \sim_{B_r} = \{[x]_{B_r} \mid x \in \mathcal{O}\}.$$

In effect, each choice of probe functions B_r defines a partition ξ_{B_r} on a set of objects \mathcal{O} , namely,

$$\xi_{B_r} = \mathcal{O} / \sim_{B_r}.$$

Every choice of the set B_r leads to a new partition of \mathcal{O} . The overlap function ν_{N_r} is defined by

$$\nu_{N_r} : \mathcal{P}(\mathcal{O}) \times \mathcal{P}(\mathcal{O}) \longrightarrow [0, 1],$$

where $\mathcal{P}(\mathcal{O})$ is the powerset of \mathcal{O} . The overlap function ν_{N_r} maps a pair of sets to a number in $[0, 1]$ representing the degree of overlap between sets of objects with features defined by probe functions $B_r \subseteq B$. For each subset $B_r \subseteq B$ of probe functions, define the binary relation $\sim_{B_r} = \{(x, x') \in \mathcal{O} \times \mathcal{O} : \forall \phi_i \in B_r, \phi_i(x) = \phi_i(x')\}$. Since each \sim_{B_r} is, in fact, the usual indiscernibility relation [12], let $[x]_{B_r}$ denote the equivalence class containing x , *i.e.*,

$$[x]_{B_r} = \{x' \in \mathcal{O} \mid \forall f \in B_r, f(x') = f(x)\}.$$

If $(x, x') \in \sim_{B_r}$ (also written $x \sim_{B_r} x'$), then x and x' are said to be *B-indiscernible* with respect to all feature probe functions in B_r . Then define a collection of partitions $N_r(B)$ (families of neighbourhoods), where

$$N_r(B) = \{\xi_{B_r} \mid B_r \subseteq B\}.$$

Families of neighborhoods are constructed for each combination of probe functions in B using $\binom{|B|}{r}$, i.e., $|B|$ probe functions taken r at a time. The family of neighbourhoods $N_r(B)$ contains a set of percepts. A *percept* is a byproduct of perception, i.e., something that has been observed [9]. For example, a class in $N_r(B)$ represents *what has been perceived about objects belonging to a neighbourhood*, i.e., observed objects with matching probe function values.

Definition 3 Near Sets *Let $X, X' \subseteq \mathcal{O}, B \subseteq \mathcal{F}$. Set X is near X' if, and only if there exists $x \in X, x' \in X', \phi_i \in B$ such that $x \sim_{\{\phi_i\}} x'$.*

If X is near X' , then X is a near set relative to X' and X' is a near set relative to X . Notice that if we replace X' by X in Def. 3, then a set X containing near objects is a near set.

Theorem 2 Families of Neighbourhoods Theorem *A collection of partitions (families of neighbourhoods) $N_r(B)$ is a near set.*

4.1 Sample Families of Neighbourhoods

Let $X \subseteq \mathcal{O}, B \subseteq \mathcal{F}$ denote a set of sample objects $\{x_0, x_1, \dots, x_7\}$ and set of functions $\{s, a, p, r\}$, respectively. Sample values of the state function $s : X \rightarrow \{0, 1\}$ and action function $a : X \rightarrow \{1, 2, 3\}$ are shown in Table 2. Assume reward function $r : A \rightarrow [0, 1]$ and a preference function $p : S \times A \rightarrow [0, 1]$. From Table 2, we can, for example, extract the collection of partitions $N_1(B)$ for $r = 1$.

$$\begin{aligned} X &= \{x_0, x_1, \dots, x_7\}, \\ N_1(B) &= \{\xi_{\{s\}}, \xi_{\{a\}}, \xi_{\{p\}}, \xi_{\{r\}}\}, \text{ where} \\ \xi_{\{s\}} &= [x_0]_{\{s\}}, [x_2]_{\{s\}}, \\ \xi_{\{a\}} &= [x_0]_{\{a\}}, [x_1]_{\{a\}}, [x_3]_{\{a\}}, \\ \xi_{\{p\}} &= [x_0]_{\{p\}}, [x_2]_{\{p\}}, [x_5]_{\{p\}}, [x_6]_{\{p\}}, \\ \xi_{\{r\}} &= [x_0]_{\{r\}}, [x_2]_{\{r\}}. \end{aligned}$$

4.2 Information Content of a Partition

The Shannon information content of an outcome x is defined by

$$h(x) = \log_2 \frac{1}{Pr(x)},$$

which is measured in bits, where *bit* denotes a variable with value 0 or 1, and $h(v)$ provides a measure of the information content of the event $x = v$ [8], which differs from the rough set-based form of entropy in [25]. The assumption made here is that the event $x = x_i$ recorded in Table 2 is random and that a sample X is symmetric, *i.e.*, each event in the sample has the same probability. In effect, $Pr(x = x_i) = \frac{1}{|X|}$. The occurrence of a class $[x]_{B_r} = [x_i]_{B_r} \in \xi_{B_r}$ is treated as a random event, and all classes in a partition ξ_{B_r} are assumed to be equally likely. Then, for example, there are 2 classes in the partition $\xi_{\{s\}} \in N_1(B)$ and $Pr([x_0] \in \xi_{\{s\}}) = \log_2 \frac{1}{2} = \log_2 2 = 1$. The information content $H(X)$ of an ensemble X is defined to be the average Shannon information content of the events represented by X .

$$H(X) = \sum_{x \in X} Pr(x) \cdot \log_2 \frac{1}{Pr(x)}.$$

Then, for example, the information content of sample partitions in $N_1(B) = \{\xi_{\{s\}}, \xi_{\{a\}}, \xi_{\{p\}}, \xi_{\{r\}}\}$ have the following information content.

$$\begin{aligned} H(\xi_{\{s\}}) &= H(\xi_{\{r\}}) = \frac{1}{2} \cdot \log_2 \frac{1}{2} + \frac{1}{2} \cdot \log_2 \frac{1}{2} = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 = 1, \\ H(\xi_{\{a\}}) &= \frac{1}{3} \cdot (3 \cdot 1.59) = 1.59, \\ H(\xi_{\{p\}}) &= \frac{1}{4} \cdot (4 \cdot 2) = 2. \end{aligned}$$

This suggests an approach to feature selection based on information content, which is computationally rather simple.

Algorithm 1: Partition Selection

Input : $NAS = (\mathcal{O}, \mathcal{F}, \sim_{B_r}, N_r, \nu_{N_r})$, $B \in \mathcal{F}$, choice r .
Output: Partition size list Φ , where $\Phi[i] =$ number of classes in partition $\xi_{B_r} \in N_r(B)$.
Initialize $i = 0$;
while ($i \leq |N_r(B)|$) **do**
 Select i^{th} partition $\xi_{B_r} \in N_r(B)$;
 $\Phi[i] = |\xi_{B_r} \in N_r(B)|$;
 $i = i + 1$;
end

5 Feature Selection

A practical outcome of the near set approach is a feature selection method. Recall that each partition $\xi_{B_r} \in N_r(B)$ contains classes defined by the relation

Algorithm 2: Feature Selection

Input : Array Φ , where $\Phi[i] = \text{number of classes in } |\xi_{B_r} \in N_r(B)|$, threshold th .

Output: Ordered list Γ , where $\Gamma[i]$ is a winning probe function.

Initialize $i = 0$;

Sort Φ in descending order based on the information content of $\xi_{B_r} \in N_r(B)$;

while ($i \geq th$) **do**

$\Gamma[i] = \Phi[i]$;

$i = i + 1$;

end

\sim_{B_r} . We are interested in the classes in each $\xi_{B_r} \in N_r(B)$ with information content greater than or equal to some threshold th . The basic idea here is to identify probe functions that lead to partitions with the highest information content, which occurs in partitions with high numbers of classes. In effect, as the number of classes in a partition increases, there is a corresponding increase in the information content of the partition. A list Φ of partition sizes is constructed using Alg. 1. By sorting Φ based on information content using Alg. 2, we have a means of means of selecting tuples containing probe functions that define partitions having the highest information content.

5.1 Sample Feature Selections

$N_2(B) = \{ \xi_{\{s,a\}}, \xi_{\{s,p\}}, \xi_{\{a,p\}}, \xi_{\{s,r\}}, \xi_{\{a,r\}}, \xi_{\{p,r\}} \}$, where

$$\xi_{\{s,a\}} = \{ [x_0]_{\{s,a\}}, [x_1]_{\{s,a\}}, [x_2]_{\{s,a\}}, [x_3]_{\{s,a\}} \},$$

$$\xi_{\{s,p\}} = \{ [x_0]_{\{s,p\}}, [x_2]_{\{s,p\}}, [x_3]_{\{s,p\}}, [x_4]_{\{s,p\}}, [x_5]_{\{s,p\}}, [x_6]_{\{s,p\}}, [x_7]_{\{s,p\}} \},$$

$$\xi_{\{a,p\}} = \{ [x_0]_{\{a,p\}}, [x_1]_{\{a,p\}}, [x_2]_{\{a,p\}}, [x_3]_{\{a,p\}}, [x_4]_{\{a,p\}}, [x_5]_{\{a,p\}}, [x_6]_{\{a,p\}}, [x_7]_{\{a,p\}} \},$$

$$\xi_{\{s,r\}} = \{ [x_0]_{\{s,r\}}, [x_2]_{\{s,r\}}, [x_6]_{\{s,r\}} \},$$

$$\xi_{\{a,r\}} = \{ [x_0]_{\{a,r\}}, [x_1]_{\{a,r\}}, [x_2]_{\{a,r\}}, [x_3]_{\{a,r\}}, [x_7]_{\{a,r\}} \},$$

$$\xi_{\{p,r\}} = \{ [x_0]_{\{p,r\}}, [x_2]_{\{p,r\}}, [x_3]_{\{p,r\}}, [x_4]_{\{p,r\}}, [x_5]_{\{p,r\}}, [x_6]_{\{p,r\}} \}.$$

This section continues the exploration of the partitions in families of neighbourhoods for each choice of $r \leq |B|$. The observations in Table 2 were produced by an ecosystem simulation reported in [21]. The function values in Table 2 were discretized using ROSE [24]. ROSE was used in this study because it allows the user to use as input a discretized table. This was important for us because Table 2 was discretized using a local Object Recognition System (ORS) toolset. The ORS discretized table was used to define the partition of the sample objects in Table 2. Collections of partitions $N_2(B), N_3(B)$ can be extracted from Table 2 (in this section, $N_2(B)$ is given and the display of $N_3(B)$ is omitted to save space).

5.2 Feature Selection Results

Table 5. Partition Information Content Summary

F_n	H	Fns	H	Fns	H
$\xi_{\{s\}}$	1.0	$\xi_{\{s,a\}}$	2.0	$\xi_{\{s,a,p\}}$	3.0
$\xi_{\{a\}}$	1.59	$\xi_{\{s,p\}}$	2.8	$\xi_{\{a,p,r\}}$	3.0
$\xi_{\{p\}}$	2.0	$\xi_{\{a,p\}}$	3.0	$\xi_{\{s,a,r\}}$	1.63
$\xi_{\{r\}}$	1.0	$\xi_{\{s,r\}}$	1.58	$\xi_{\{s,p,r\}}$	2.8
		$\xi_{\{a,r\}}$	1.46		
		$\xi_{\{p,r\}}$	1.63		

In the single feature case ($r = 1$), functions a and p define partitions with the highest information content for sample X represented by Table 2. Notice that preference p is more important than a in the ecosystem experiments, because p indicates the preferred action in a give state. In the two feature case ($r = 2$), feature combination a, p define a partition with the highest information content, namely, 3 (see Table 5). If we set the threshold $r \geq 3$, then features a, p (action, preference) are selected because the information content of partition $H(\xi_{\{a,p\}}) = 3$ in $N_2(B)$, which is also one of the reducts found by ROSE [24] using Table 2 as input. Notice that a, p defines all of the high scoring partitions in Table 5. Similarly, for $N_3(B)$, the features s, a, p are selected, where the information content of $H(\xi_{\{s,a,p\}}) = 3$ (highest) and $H(\xi_{\{s,a,r\}}) = 1.58$ (lowest) which is at variance with the findings by ROSE, which selects $\{s, a, r\}$ as a reduct. Finally, notice that ROSE selects a as the core feature and the highest scoring partitions defined by combinations of functions containing a .

5.3 Complexity Issue

Notice that the proposed feature selection method is based on collections of partitions, each contain a family of neighborhoods, constructed for each combination of probe functions in B using $\binom{|B|}{r}$, *i.e.*, $|B|$ probe functions taken r at a time. Hence, the proposed method has very high complexity for large B . To achieve feature selection with polynomial time complexity, features are selected by considering only the partitions in $N_1(B)$. This approach can be used to identify all of those partitions with information content $H(\xi_f)$ for partition $X/ \sim_{\{f\}}$ defined relative to a single function f representing a feature of the sample objects. For recognition of sample objects with thousands of features, this approach is useful in an unsupervised learning environment². More work needs to be done on the proposed feature selection method to make it useful

² See, *e.g.* lymphocyte samples with genes representing 4026 features [2]. Assuming that a maximum of m operations are required in each partition, then $4026 \cdot m$ operations are necessary for feature selection for single feature partitions. By choosing a

in discovering reducts suitable for supervised learning for sample objects with many features.

6 Conclusion

The proposed approach to the nearness of objects and sets and feature selection based on object description is a direct result of the original approach to the classification of objects introduced by Zdzisław Pawlak during the early 1980s. Feature selection is based on measurement of the information content of the partitions in each family of neighbourhoods. Future work will include consideration of the relation between the proposed feature selection method for large numbers of features for objects in various sample spaces.

References

1. Banerjee, M, Chakraborty, M.K.: Algebras from rough sets. In: Pal, S.K., Polkowski, L., Skowron, A. (Eds.): Rough-neuro Computing: Techniques for Computing with Words. Springer, Berlin (2004) 157-184.
2. Banerjee, M., Mitra, S., Banka, H.: Evolutionary rough feature selection in gene expression data, IEEE Transactions on Systems, Man, and Cybernetics–Part C: Applications and Reviews, 37(4) July (2007) 1-12.
3. Cattaneo, G., Ciucci, D.: Algebraic structures for rough sets, Transactions on Rough Sets, vol. II, LNCS 3135 (2004) 63-101.
4. Düntsch, I.: A logic for rough sets, Theoretical Computer Science 179 (1997) 427-436.
5. Greco, S., Matarazzo, B., Slowinski, R.: Dominance-based rough set approach to knowledge discovery. In: N. Zhong, J. Liu (Eds.): Intelligent Technologies for Information Analysis, Springer, Berlin (2004) 513-552.
6. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A. (Eds.): Feature Extraction. Foundations and Applications. Springer, Berlin (2006).
7. Henry, C., Peters, J.F.: Image Pattern Recognition Using Approximation Spaces and Near Sets, In: Proceedings of Eleventh International Conference on Rough Sets, Fuzzy Sets, Data Mining and Granular Computing (RSFDGrC 2007), Joint Rough Set Symposium (JRS 2007), Lecture Notes in Artificial Intelligence, vol. 4482 (2007) 475-482.
8. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms, Cambridge University Press, UK (2003).
9. Murray, J.A., Bradley, H., Craigie, W., Onions, C.: The Oxford English Dictionary, Oxford University Press, London (1933).
10. Orłowska, E.: *Semantics of Vague Concepts*, Applications of Rough Sets, Institute for Computer Science, Polish Academy of Sciences, Report 469 (1982).
11. Pavel, M.: Fundamentals of Pattern Recognition, 2nd Ed., Marcel Dekker, Inc., NY (1993).
12. Pawlak, Z.: *Classification of Objects by Means of Attributes*, Institute for Computer Science, Polish Academy of Sciences, Report 429 (1981).

threshold th , it is then possible to select the most discriminating features useful in classifying the lymphocyte samples.

13. Pawlak, Z., Skowron, A.: Rudiments of rough sets, *Information Sciences*, vol. 177 (2007) 3-27.
14. Peters, J.F.: *Near sets. Special theory about nearness of objects*, *Fundamenta Informaticae*, vol. 76, (2007) 1-28.
15. Peters, J.F.: *Near sets. General theory about nearness of objects*, *Applied Mathematical Sciences Journal* (2007), *in press*.
16. Peters, J.F.: *Near Sets. Toward Approximation Space-Based Object Recognition*, In: Yao, Y., Lingras, P., Wu, W.-Z, Szczuka, M., Cercone, N., Ślęzak, D., Eds., *Proc. of the Second Int. Conf. on Rough Sets and Knowledge Technology (RSKT07), Joint Rough Set Symposium (JRS07), Lecture Notes in Artificial Intelligence 4481, Springer, Berlin, (2007) 22-33.*
17. Peters, J.F.: *Classification of objects by means of features*, In: *Proc. IEEE Symposium Series on Foundations of Computational Intelligence (IEEE SSCI 2007), Honolulu, Hawaii (2007) 1-8.*
18. Peters, J.F., Skowron, A., Stepaniuk, J.: *Nearness in approximation spaces*, G. Lindemann, H. Schilngloff et al. (Eds.), *Proc. Concurrency, Specification & Programming (CS&P'2006). Informatik-Berichte Nr. 206, Humboldt-Universität zu Berlin (2006) 434-445.*
19. Peters, J.F., Skowron, A., Stepaniuk, J.: *Nearness of Objects: Extension of Approximation Space Model*, *Fundamenta Informaticae*, vol. 79 (2007) 1-24.
20. Peters J.F., Henry C., Gunderson D.S.: *a Biologically-inspired approximate adaptive learning control strategies: A rough set approach*, *International Journal of Hybrid Intelligent Systems*, (2007) *to appear*.
21. Peters J.F., Henry C., Ramanna, S.: *Rough Ethograms: Study of Intelligent System behaviour*. In: M.A. Kłopotek, S. Wierchoń, K. Trojanowski (Eds.), *New Trends in Intelligent Information Processing and Web Mining (IIS05)*, Gdańsk, Poland (2005) 117-126.
22. Polkowski, L.: *Rough Sets. Mathematical Foundations*. Springer-Verlag, Heidelberg (2002).
23. J.J. Rissanen, *A universal prior for integers and estimation by Minimum Description Length*. *Annals of Statistics* 11(2) (1983) 416-431.
24. *Rough Sets Data Explorer, Version 2.2, ©1999-2002 IDSS*, <http://idss.cs.put.poznan.pl/site/software.html>
25. Shankar, B.U.: *Novel classification and segmentation techniques with application to remotely sensed images*, *Transactions on Rough Sets*, vol. VII, LNCS 4400 (2007) 295-380.
26. Shannon, C.E.: *A mathematical theory of communication*, *Bell Systems Technical Journal* 27 (1948) 279-423, 623-656.
27. Skowron, A., Stepaniuk, J.: *Generalized approximation spaces*. In: Lin, T.Y., Wildberger, A.M. (Eds.), *Soft Computing, Simulation Councils, San Diego (1995) 18-21.*
28. Skowron, A., Swiniarski, R., Synak, P.: *Approximation spaces and information granulation*, *Transactions on Rough Sets*, vol. III (2005) 175-189.
29. Skowron, A., Stepaniuk, J., Peters, J.F., Swiniarski, R.: *Calculi of approximation spaces*, *Fundamenta Informaticae*, 72(1-3) (2006) 363-378.

Contextual Adaptive Clustering with Personalization

Krzysztof Ciesielski, Mieczysław A. Kłopotek, Sławomir Wierzchoń

Institute of Computer Science, Polish Academy of Sciences,
ul. Ordona 21, 01-237 Warszawa, Poland
kciesiel,kłopotek,stw@ipipan.waw.pl

Abstract. We present a new method of modeling of cluster structure of a document collection and outline an approach to integrate additional knowledge we have about the document collection like prior categorization of some documents or user defined / deduced preferences in the process of personalized document map creation.

1 Introduction

Clustering of Web documents, especially in large and heterogeneous collections, is a challenging task, both in terms of processing time complexity and clustering quality.

But the most challenging part is the way how the clustering information is conveyed to the end user and how it meets his expectations (personalization).

Note that a text document is usually a very complex information structure from the point of view of human beings dealing with this document. But actually a computer system fully understanding the document contents is beyond technological possibilities. Therefore some kinds of approximation to the content are done. Documents are frequently treated as a bag of words in computer models, and the documents are viewed as points in term-document vector space, but this approach turns out to be insufficient, hence more complex representations, both of the content of a single document and their collections are investigated.

In the recent years, various projects (WebSOM, [11–13, 7], Themescape etc.) were aimed at developing a new form of cluster description – a visual document map representation. In a two-dimensional space, consisting of quadratic or hexagonal cells, the split into clusters is represented as an assignment of documents to cells in such a way, that documents assigned to cells are as homogenous as possible and cells (clusters) containing similar documents are placed close to one another on the map, and map regions are labeled with best-fitting terms from the documents. An inversion of the clustering (that is clustering of terms instead of documents) is also possible.

It is generally believed that individual information needs of users may differ and there is a general feeling that therefore also the data processing results should accommodate to the profile of the user. Countless methods and ways of user profile representation and acquisition have been designed so far.

The problem with map-like representation of document collections, however, lies in the pretty expensive processing / high complexity (in terms of time and space) so that a personalized ad-hoc representation is virtually impossible.

In the current paper we contest this view and claim that personalization of map representation of large scale document collections is possible by a careful separation of the concept of individual needs from the concept of common knowledge. This leads to the possibility of separation of computationally intense tasks of identification of the structure of clustering space from the relatively less resource consuming pure presentation part.

The paper is concerned with outlining the general concept, while the Reader may refer to our earlier papers for experimental results supporting validity of partial claims from which the current exposition is derived.

The paper is organized as follows: In section 2 the concept of cluster space is presented. Section 3 describes practical ways of cluster space approximation from data. Section 4 outlines possible ways of personalizing the maps presentation. Last section, 5, summarizes the paper and gives some future directions for research.

2 Clustering Space

It is usually assumed that personalization is needed because of cultural, ethnical etc. differences that influence the world of values, the attitudes and the views. So, in the particular case of clustering, the distances (or more strictly speaking dissimilarities) between the objects may change from person to person. So a personalization under this assumption would be reduced to a total re-clustering of the objects.

We disagree with this assumption. While avoiding here a deeper philosophical discussion of the issue, let us point at some issues significant for the text processing task. Human beings possess to a large extent an objective world which they describe with a common set of concepts (a vocabulary) that is intended for them to communicate to other human beings. So the vast majority of concepts is shared and their meaning not determined by values, attitudes etc. What differs the human beings is the current task they are interested in. So if discussing e.g. an issue in biology, one does not care about concepts important for chemical engineering. Hence it is not the personal attitude, but rather the context in which an issue is discussed that impacts the feeling of dissimilarity of opinions (in this case documents).

Therefore, in our opinion, the proper approach to personalization has to be well founded on the proper representation of document information.

It has been generally agreed that the processing of textual information, especially at large scale, requires a simplification of the document representation. So a document is treated as a bag of words (without bothering about sentence structure). It is represented in the space of documents (with dimensions being spread by words and phrases, that is terms) as a point with coordinates being

a function of the frequency of each term in the document. The similarity between two documents is measured as a cosine of the angle between the vectors drawn from the origin of the coordinate system to these points. Furthermore, dimensionality reduction may be of high importance [6]

So the weight of a term in the document is calculated as the so-called *tfidf* (term frequency times the inverse document frequency) computed according to the formula:

$$w_{t,d} = f_{t,d} \times \log \frac{|D|}{f_D^{(t)}} \quad (1)$$

where $f_{t,d}$ is the number of occurrences of term t in document d , $|D|$ is the cardinality of the set of documents, and $f_D^{(t)}$ is the number of documents in collection D containing at least one occurrence of term t .

A document d is described by a vector $d = (w_{t_1}, \dots, w_{t_{|T|}})$, where T is the set of terms. Usually, the document is turned into a normalized form $d = (w_{t_1}, \dots, w_{t_{|T|}})$ where d has a unit length.

We notice immediately that the weights of terms in a document are influenced by their distribution in entire document collection. If we look, however, at any process of clustering, we immediately notice that weights calculated by the very same method, but within any reasonable (homogenous) cluster, not being a random subsample of the whole collection, would exhibit considerable differences to the global weighing scheme. However, this fact is actually ignored by most researchers.

So our first methodological step is to resign from the rigid term weighing scheme for the sake of local differentiation of term weighing taking into account the local context (of the identified cluster). Though this vision of document similarity works quite well in practice, agreeing with human view of text similarity, it has been early recognized, that some terms should be weighed more than other. One should reject the common words (stop-words) appearing frequently in all documents, as well as those which appear quite seldom.

But we do not want to replace global term weighing scheme with a global weighing scheme. Rather than this, we reconsider the impact of the documents that are far away from cluster core, on the term weighing. Another important point is that terms specific for a given cluster should weigh more than terms not specific for any cluster. Last not least, let us relax the notion of a cluster to the concept of clustering space. Let us speak of a point p in the document space (a point of an unit hyper-sphere) as a vector $p = (w_{t_1}, \dots, w_{t_{|T|}})$ where $\|p\| = 1$ (unit length). Each such a point can be treated as cluster center in a (continuous) clustering space. We can then define, for each document d , a membership function $m_{d,C(p)}$ in the style of fuzzy set membership function [2], for example as

$$m_{d,C(p)} = \sum_{t \in T} w_t(d) \cdot w_t(p) \quad (2)$$

that is the dot product of vectors d and p .

Given this concept, we can define the specificity $s_{t,C}$ of a term t in a cluster $C(p)$ as

$$s_{t,C(p)} = |C(p)| \cdot \frac{\sum_{d \in D} (f_{t,d} \cdot m_{d,C(p)})}{f_{t,D} \cdot \sum_{d \in D} m_{d,C(p)}} \quad (3)$$

where $f_{t,d}$ is (as earlier) the number of occurrences of term t in document d , $f_{t,D}$ is the number of occurrences of term t in document collection D , and $|C(p)|$ is the fuzzy cardinality of documents at point p , defined as

$$|C(p)| = \sum_{d \in D} \mu_{d,C(p)} \quad (4)$$

where $\mu_{d,C}$ is the normalized membership:

$$\mu_{d,C(p)} = \frac{m_{d,C(p)}}{\int_{p \in HS} m_{d,C(p)}}$$

and HS is the unit hyper-sphere.

In this way we arrive at a new (contextual [5]) term weighing formula for term t in the document d from the point of view of the

$$w_{t,d,C(p)} = s_{t,C(p)} \times f_{t,d} \times \log \frac{|C(p)|}{f_{C(p)}^{(t)}} \quad (5)$$

where $f_{C(p)}^{(t)}$ is the fuzzy count of documents in collection $C(p)$ containing at least one occurrence of term t ,

$$f_{C(p)}^{(t)} = \sum_{\{d: f_{t,d} > 0\}} m_{d,C(p)} \quad (6)$$

For consistency, if $f_{C(p)}^{(t)} = 0$ we define $w_{t,d,C(p)} = 0$.

The universal weight *tfidf* given by equation (1) will be replaced by the concept of an averaged local weight

$$w_{t,d} = \frac{\int_{p \in HS} m_{d,C(p)} \cdot w_{t,d,C(p)}}{\int_{p \in HS} m_{d,C(p)}} \quad (7)$$

where HS is the unit hyper-sphere.

Note that the definition of term weights $w_{t,d}$ becomes recursive in this way ($m_{d,C(p)}$ is used here, which is computed in equation (2) based on $w_{t,d}$ itself) and the fixpoint of this recursion is the intended meaning of term weight.

Our further concern is the way how typical hierarchical (or other multi-stage) algorithms handle lower level clusters. The cluster is viewed as a kind of averaged document, eventually annotated with standard deviation of term frequencies and/or term weights. In our opinion, the distribution (approximated in our approach by a discrete histogram) of the term weight (treated as a random variable) reflects much better the linguistic nature of data. The clusters should

be formed not as hyperspheres around some center, but rather as collections of documents with terms used in a similar way. This was confirmed by our reclassification experiments [4], showing higher stability of histogram-based cluster description versus centroid-based representation¹.

So for any point p in the clustering space and any term t we define a term-weight distribution as one approximated by the histogram in the following manner: Let $\Delta(w, t)$ be a discretization of the normalized weights for the term t , assigning a weight for a term the integer identifier of the interval it belongs to (higher interval identifiers denote higher weights). Let $\chi(d, t, q, p)$ be the characteristic function of the term t in the document d and the discretization interval identifier q at point p , equal to $m_{d, C(p)}$ if $q = \Delta(w_{t, d, C(p)}, t)$, and equal zero otherwise. Then the histogram $h(t, p, q)$ is defined as

$$h(t, p, q) = \sum_{d \in D} \chi(d, t, q, p) \quad (8)$$

With h we denote a histogram normalized in such a way that the sum over all intervals q for a given t and p is equal 1:

$$h'(t, p, q) = \frac{h(t, p, q)}{\sum_q h(t, p, q)} \quad (9)$$

We can easily come to the conclusion, when looking at typical term histograms that terms significant for a cluster would be ones that do not occur too frequently nor too rarely, have diversified range of values and have many non-zero intervals, especially with high indices.

Hence the significance of a term t for the clustering point p may be defined as

$$m_{t, C} = \frac{\sum_q [q \cdot \log(h(t, p, q))]}{Q_t} \quad (10)$$

where Q_t is the number of intervals for the term t under discretization.

Let us denote with $H(t, p, q)$ the right cumulative histograms, that is $H(t, p, q) = \sum_{k \geq q} h(t, p, k)$. The right cumulative histograms are deemed to reflect the idea, that terms with more weight should be more visible. For technical reasons H is a histogram normalized in the same way as h .

Let us measure the divergence between clustering points p_i, p_j with respect to term t as (Hellinger divergence, called also Hellinger-Matsushita-Bhattacharya divergence, [1])

$$Hell_k(p_i, p_j, t) = \sqrt{\sum_q (H(t, p_i, q)^{(1/k)} H(t, p_j, q)^{(1/k)})^k} \quad (11)$$

¹ reclassification measure evaluates consistency of the model-derived clustering with the histogram-based clustering space description (cf. [4])

Finally let us measure the divergence between clustering points p_i, p_j as such as

$$dst(p_i, p_j) = \frac{\sum_{t \in T} m_{t, C(p_i), C(p_j)} \cdot Hell_k(p_i, p_j, t)}{\sum_{t \in T} m_{t, C(p_i), C(p_j)}} \quad (12)$$

where

$$m_{t, C(p_i), C(p_j)} = \sqrt{(m_{t, C(p_i)} + 1) \cdot (m_{t, C(p_j)} + 1)} - 1$$

With this definition, we can speak of a general notion of a cluster as islands in the clustering space such that the divergence within them differs not significantly, and there exist at least n documents belonging predominantly to such an island. Thus, it can be treated as a dissimilarity measure.

It may be easily deduced that equation (10) gives also interesting possibilities of labeling of cluster space with meaningful sets of terms (concepts).

2.1 User-related sources of information

Let us now turn to the user related information. Some documents may be pre-labeled by the user (with category, liking, etc.), there may be past queries available etc.

Note that the contextual document space, as described in the previous section, may be viewed as a pure space with some material objects causing a kind of curvature of this space.

The user-related sources can be viewed as consisting of two types of documents: material objects (all the positively perceived, relevant information) and the anti-material objects (all the negatively perceived information).

The user-related documents may be also represented in a clustering space, in at least two different ways:

- in separate user-material, user-anti-material and proper document clustering spaces - in this case a superposition of these spaces would serve as an additional labeling of the proper document space, beside the original labels derived from document collection content.
- in a joint space in this case user-related information will transform the document space of the document collection.

While the second approach may be considered as a stronger personalization, it will be more resource consuming and raises the issue of pondering the impact of user related documents against the entire collection, and also that of the relation between positive and negative user information. The first approach will be for sure much less resource consuming, because the processing of the big entire document collection has to be done only once, and the user related information is usually of marginal size and can be processed in a speedy way.

3 Clustering Space Approximation

If we look at the clustering space as a continuum, it is obvious, that we cannot consider clusters in isolation, but we want to take relationships between them into account. It is also obvious that need to provide a finite, discrete approximation of this continuum. To achieve it, the usually wide areas of clustering space of next to zero proximity to the documents will be ignored, as well as those terms that within the given subspace are of marginal significance. This is caused by the fact that each document modifies the space close to it having marginal impact of the rest. So the space may be greedy subdivided into non-empty subspaces that are deemed to be linked if they adhere to one another, and not, if they are of next to zero similarity.

The process, that we apply to approximate the clustering space [10], which we call Adaptive Clustering Algorithm, starts with splitting of the document collection into a set of roughly equally sized sub-collections using the expression (1) as an approximation of term weights for document similarity computation in a traditional clustering algorithm. We work in a hierarchical divisive mode, using the algorithm to split the collection in a small number of subcollections and apply further splitting to sub-collections of too big size. At the end too small clusters are merged with most similar ones. As a next iteration for each sub-collection, being now treated as a context (as it is now feasible), an iterative recomputation of term weights according to equation (7) with respect to cluster center, making the simplifying assumption that documents from other contexts have no impact. Within each context, the dictionaries of terms are reduced removing insignificant terms in a given context (different terms may be zeroed in different contexts). Subsequently the inter-document structure is formed. For this purpose one of the known networking clustering algorithms is used, either the growing neural gas [8] or idiotypic (artificial immune) network [14, 3]. Finally we turn back to the global set of contexts and apply a networking clustering algorithm to representatives of each context. This time, the histograms of contexts are applied to compute a measure of similarity between contexts see equation (12). While applying the networking clustering, we additionally compute so-called major topics, that is a split of the (sub)collection into up to 6 sub-clusters, the representatives of which are deemed to be major topics of the collection.

In this way, an approximation of the clustering space is obtained. In case of visualization, the WebSOM algorithm is applied to context representatives, in case one wants to view the global map, and to neural gas cells, or immune network cells in case of detailed view of a context. The computation of the map given the clustering space model is drastically simplified because e.g. with a collection of 12,000,000 documents we need to cluster only 400 representatives. So given such a cluster network, its projection onto a flat rigid document map structure, with treating each whole cluster as a single document, is a dramatically simpler task than the map creation process for individual documents.

Our implementation of WebSOM differs from the original one in a number of ways, accelerating the processing significantly. One of the features is the topic-sensitive initialization. While WebSOM assigns random initial cluster centers

for map cells, we distribute evenly the vectors of major topics over the map and initialize the remaining cells with in-between values (with slight noise). In this way the maps are learned usually quicker and are more stable (no drastic changes from projection to projection).

We have demonstrated in our earlier work [3–5, 10] that such an approach to document space modeling is stable, scalable and can be run in an incremental manner.

3.1 Exploiting user-related sources

With this background we can explain our approach to personalization. We treat the document collection as a piece of knowledge that is esteemed by any user in the same way. So the identified clusters and the identified interrelationships between them are objective, independent of the user. The user at a given moment may be, however, interested to view the collection from a different direction. So the personalization may be reduced to the act of projection of the cluster network onto the flat map, that is, contrary to projection of document collection, a speedy process, to be managed within seconds. In this process, we can proceed in two distinct ways:

- instead of using the topical vectors of a context / global collection, the user profile topical vector is applied, or
- the user related documents are attached to the collection clusters prior to projection (and may or may not influence the projection process) and serve as a source of additional labeling.

3.2 Another view of the Adaptive Clustering Algorithm

Our incremental textual data clustering algorithm relies on merging two known paradigms of clustering: the fuzzy clustering and the subspace clustering. The method differs essentially from Fuzzy C-Means in that it is designed solely for text data and is based on contextual vector representation and histogram-based description of vector subspaces.

Like Fuzzy-C-Means, we start with an initial split into subgroups, represented by a matrix $U(\tau_0)$, rows of which represent documents, and columns representing groups, they are assigned to. Iteratively, we adapt (a) the document representation, (b) the histogram description of contextual groups, (c) membership degree of documents and term significance in the individual groups.

These modifications can be viewed as a recursive relationship leading to a precise description of a contextual subspace in terms of the membership degree of documents and significance of terms in a context and on the other hand improving the understanding of document similarity.

So we can start without any knowledge of document similarity, via a random assignment of documents to a number of groups and global term weighing. But through the iterative process some terms specific for a group would be strengthened, so that class membership of documents would be modified, hence also their vector representation and indirectly similarity definition.

So we can view the algorithm as a kind of reinforcement learning. The usage of histogram approach makes this method incremental.

4 Personalization

The outlined approach to document map oriented clustering enables personalization among others along the following lines:

- personalized topic -oriented initialization of map like visualization of the selected document space model (also rebuilding of a component model is possible, treating the user profile as a modifier of term weights of all documents)
- personalized identification of key words, document space / map cell labeling, query expansion
- document recommendation based on document membership degree in client profile context
- recommendation of map cells
- recommendation of other users (measuring the histogram distances between user profiles)
- clustering of users as well as users and contexts

Present day search engines are characterized by a static information model. This means that textual data bases are updated in a heavily discontinuous way which results in abrupt changes of query results (after each cycle of indexing new documents). On the other hand the data organization and search model does not take into account the user profile information for the given document base and the given user query. Hence the reply is frequently identical, independent of the user.

The experimental search engine BEATCA [10] exhibits several capabilities that can become a starting point for a radical change of this situation.

- reduced processing time, scalability of the adaptive contextual approach, reduced memory requirements of the implemented clustering algorithms (contextual reduction of the vector space) and search (inverted lists compression)
- possibility of construction and maintenance of multiple models/maps representing diverse views of the same document collection (and fitting the map to the query)
- possibility of inclusion of system-user interaction history into the algorithm of map initialization (e.g. by strengthening / weakening of terms from documents evaluated by the user as more or less interesting)
- possibility of inclusion of user preference profiles into the modeling process itself by taking into account the automatically collected information on user walk through the collection or provided externally

5 Conclusions

We presented a new concept of document cluster characterization via term (importance) distribution histograms. This idea allows the clustering process to have a deeper insight into the role played by the term in formation of a particular cluster. So a full profit can be taken from our earlier idea of "contextual clustering", that is of representing different document clusters in different subspaces of a global vector space. Such an approach to mining high dimensional datasets proved to be an effective solution to the problem of massive data clustering. The contextual approach appears to be fast, of good quality and scalable (with the data size and dimension). Additionally, the histogram-based characterization of document clusters proved to be a stabilizing factor in creating the clustering structure, and well suited for document classification. As a side effect, a new internal cluster quality measure, based on histograms, has been developed.

We believe that the idea of histogram-based subspace identification and evaluation can be efficiently applied not only to textual, but also other challenging high dimensional datasets (especially those characterized by attributes from heterogeneous or correlated distributions).

Contextual approach leads to many interesting research issues, such as context-dependent dictionary reduction and keywords identification, topic-sensitive document summarization, subjective model visualization based on particular user's information requirements, dynamic adaptation of the document representation and local similarity measure computation. Especially, the user-oriented, contextual data visualization can be a major step on the way to information retrieval personalization in search engines.

Our further research on personalization in document maps will be directed towards:

- chronology (sequence) of visiting the documents by the user versus the incremental growth of document collection
- capturing the relationship between contexts and users and identification of topical trends versus random walk
- exploitation of link structure (between documents and contexts) as modifiers of clustering space

Acknowledgement

This research has been partly funded by the European Commission and by the Swiss Federal Office for Education and Science with the 6th Framework Programme project REVERSE no. 506779 (cf. <http://reverse.net>). Also co-financed by Polish state budget funds for scientific research under grants No. N516 00531/0646 and No. N516 01532/1906.

References

1. A. Basu, I. R. Harris, and S. Basu. Minimum distance estimation: The approach using density-based distances. In G. S. Maddala and C. R. Rao, editors, *Handbook of Statistics*, volume 15, pages 21-48. North-Holland, 1997.
2. J.C. Bezdek, S.K. Pal, *Fuzzy Models for Pattern Recognition: Methods that Search for Structures in Data*, IEEE, New York, 1992
3. K. Ciesielski, S. Wierzchon, M. Kłopotek, An Immune Network for Contextual Text Data Clustering, in: H.Bersini, J.Carneiro (Eds.), 5th International Conference on Artificial Immune Systems (ICARIS-2006), Oeiras, LNCS 4163, Springer-Verlag, 2006, pp.432-445
4. K. Ciesielski, M. Kłopotek, Towards Adaptive Web Mining: Histograms and Contexts in Text Data Clustering, to appear in: M.R.Berthold, J.Shawe-Taylor (Eds.), *Intelligent Data Analysis – Proceedings of IDA-2007*, Ljubljana, September 2007, Springer-Verlag, LNCS 4723, pp.284-295
5. K. Ciesielski, M. Kłopotek, Text Data Clustering by Contextual Graphs, in: L.Todorovski, N.Lavrac, K.P.Jantke, 9th International Conf. on Discovery Science (ALT/DS 2006), Barcelona, LNAI 4265, Springer-Verlag, 2006, pp.65-76
6. S.C. Deerwester, S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harshman, Indexing by Latent Semantic Analysis, *Journal of the American Society of Information Science*, 41(1990)6, ppp. 391-407, citeseer.nj.nec.com/deerwester90indexing.html
7. M. Dittenbach, A. Rauber, D. Merkl, Uncovering hierarchical structure in data using the Growing Hierarchical Self-Organizing Map. *Neurocomputing* 48 (1-4)2002, pp. 199-216.
8. B. Fritzke, A growing neural gas network learns topologies, in: G. Tesauero, D.S. Touretzky, and T.K. Leen (Eds.) *Advances in Neural Information Processing Systems* 7, MIT Press Cambridge, MA, 1995, pp. 625-632.
9. C. Hung, S. Wermter, A constructive and hierarchical self-organising model in a non-stationary environment, *Int.Joint Conference in Neural Networks*, 2005
10. M. Kłopotek, S. Wierzchon, K. Ciesielski, M. Draminski, D. Czerski, Techniques and Technologies Behind Maps of Internet and Intranet Document Collections, in: Lu, Jie; Ruan, Da; Zhang, Guangquan (Eds.): *E-Service Intelligence – Methodologies, Technologies and Applications*. Springer-Verlag Series: Studies in Computational Intelligence, Vol. 37, 2007, X, 711 p., 190 illus., Hardcover, ISBN-10: 3-540-37015-3, ISBN-13: 978-3-540-37015-4
11. T. Kohonen, *Self-Organizing Maps*, Springer Series in Information Sciences, vol. 30, Springer-Verlag, 2001
12. T. Kohonen, S. Kaski, P. Somervuo, K. Lagus, M. Oja, V. Paatero, Self-organization of very large document collections, Helsinki University of Technology technical report, 2003, <http://www.cis.hut.fi/research/reports/biennial02-03>
13. A. Rauber, *Cluster Visualization in Unsupervised Neural Networks*, Diplomarbeit, Technische Universitt Wien, Austria, 1996
14. J. Timmis, aiVIS: Artificial Immune Network Visualization, in: *Proceedings of EuroGraphics UK 2001 Conference*, Univeristy College London 2001, pp.61-69

Unsupervised Grouping of Trajectory Data on Laboratory Examinations for Finding Exacerbating Cases in Chronic Diseases

Shoji Hirano and Shusaku Tsumoto

Department of Medical Informatics, Shimane University, School of Medicine
89-1 Enya-cho, Izumo, Shimane 693-8501, Japan
E-mails: hirano@ieee.org, tsumoto@computer.org

Abstract. In this paper we present a method for finding exacerbating cases in chronic diseases based on the cluster analysis technique. Cluster analysis of time series hospital examination data is still a challenging task as it requires comparison of data involving temporal irregularity and multidimensionality. Our method first maps a set of time series containing different types of laboratory tests into a directed trajectory representing the time course of patient status. Then the trajectories for individual patients are compared in multiscale and grouped into similar cases. Experimental results on synthetic digit-stroke data showed that our method could yield low error rates (0.016 ± 0.014 for classification and 0.118 ± 0.057 for cluster rebuild). Results on the chronic hepatitis dataset demonstrated that the method could discover the groups of exacerbating cases based on the similarity of ALB-PLT trajectories.

1 Introduction

Steady operations of hospital information systems over the past two decades gave them a new role of archiving temporal data about long-term condition of patients, in addition to providing information necessary for daily clinical services. Such archives of longitudinal, time-series data can be used as a new source for retrospective study on chronic diseases, which may lead to the discovery of novel knowledge useful for diagnosis or treatment. However, large-scale, cross-patient analysis of time-series medical data is still a challenging task because of the multidimensionality and temporal irregularity of data caused by the variety of laboratory tests and change of patient conditions over time, as well as the difficulty in determining observation scales appropriate for capturing short-term and long-term events.

In this paper, we present a novel cluster analysis method for time-series medical data and its application to finding groups of exacerbating cases in chronic hepatitis. Our method represents time series of test results as trajectories in multidimensional space, and compares their structural similarity by using the multiscale comparison technique [1]. It enables us to find the part-to-part correspondences between two trajectories, taking into account the relationships between different tests. The resultant dissimilarity can be further used as input for clustering algorithms for finding the groups of similar cases.

In the experiments we demonstrate the usefulness of our approach through the grouping tasks of artificially generated digit stroke trajectories and medical test trajectories on chronic hepatitis patients.

The main contributions of our method are twofold. First, it proposes a new approach to comparing trajectories of medical data by shape comparison techniques, not by standard time-series analysis techniques. Second, it introduces a two-step method for deriving dissimilarity between trajectories; multiscale matching is firstly applied in order to find structurally similar parts, and after that value-based dissimilarity is derived for each of the matched pairs and accumulated as the final dissimilarity between trajectories. This scheme makes the dissimilarity more informative as it takes not only value-based features but also structural features into account.

2 Comparison of Trajectories

2.1 Trajectory Representation of Time Series

Let us assume that each patient received M types of hospital laboratory examinations. Each of the M examinations constitutes time series; thus we represent it as $ex_m(t)$, where $m \in M$. Then the M -dimensional trajectory of examination results, $c(t)$, is represented by

$$c(t) = \{ex_1(t), ex_2(t), \dots, ex_M(t)\}$$

Next, let us denote an observation scale by σ . Then the time-series of the m -th examination at scale σ , $EX_m(t, \sigma)$ is derived by convoluting $ex_m(t)$ with a smoothing kernel $g(t, \sigma)$ as follows.

$$EX_m(t, \sigma) = ex_m(t) \otimes g(t, \sigma)$$

The sampled Gaussian kernel is a general choice for the smoothing kernel; however, according to Lindeberg [2], the use of modified Bessel function is more appropriate for discrete time series because the sampled Gaussian may lose some of the desirable properties that a continuous Gaussian has, for example, non-creation of local extrema with the increase of scale. Using the modified Bessel function, $EX_m(t, \sigma)$ can be derived as follows.

$$EX_m(t, \sigma) = \sum_{n=-\infty}^{\infty} e^{-\sigma} I_n(\sigma) ex_m(t - n)$$

where $I_n(\sigma)$ denotes the modified Bessel function of order n . The first- and second-order derivatives of $EX_m(t, \sigma)$ are derived as follows.

$$EX'_m(t, \sigma) = \sum_{n=-\infty}^{\infty} -\frac{n}{\sigma} e^{-\sigma} I_n(\sigma) ex_m(t - n)$$

$$EX''_m(t, \sigma) = \sum_{n=-\infty}^{\infty} \frac{1}{\sigma} \left(\frac{n^2}{\sigma} - 1 \right) e^{-\sigma} I_n(\sigma) ex_m(t - n)$$

By applying the above convolution independently to each time series $ex_m(t), \forall m \in M$, we obtain the trajectory of examination results at scale σ , $C(t, \sigma)$, as follows.

$$C(t, \sigma) = \{EX_1(t, \sigma), EX_2(t, \sigma), \dots, EX_M(t, \sigma)\}$$

By changing the scale factor σ , we can represent the shape of trajectory at various observation scales. Figure 1 illustrates an example of multiscale representation of trajectories where $M = 2$. Increase of σ induces the decrease of convolution weights for neighbors. Therefore, more flat trajectories with less inflection points are observed at higher scales.

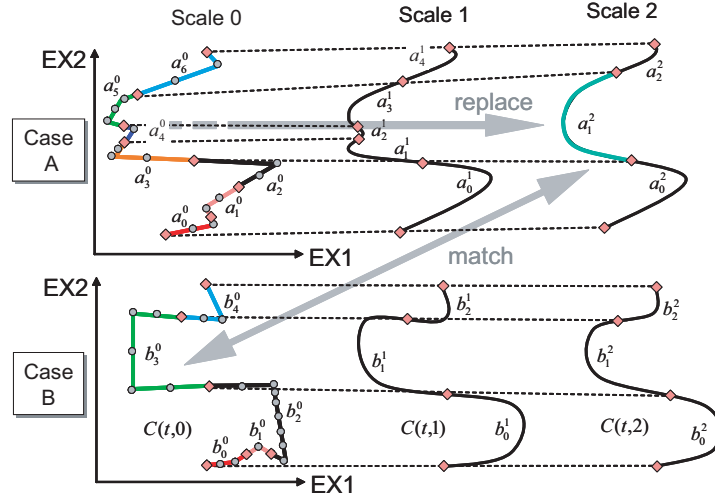


Fig. 1. Multiscale representation and matching.

2.2 Hierarchy Trace and Matching

For each trajectory $C(t, \sigma)$ represented at various scales, we identify the places of inflection points according to the sign of curvature. Curvature of the trajectory at time point t is defined by, for example of $M = 2$,

$$K(t, \sigma) = \frac{EX_1' EX_2'' + EX_1'' EX_2'}{(EX_1'^2 + EX_2'^2)^{3/2}}$$

where EX_m' and EX_m'' denotes the first- and second-order derivatives of $EX_m(t, \sigma)$ respectively. Then we divide each trajectory into a set of convex/concave segments, where both ends of a segment correspond to adjacent inflection points. Let A be a trajectory at scale k composed of $N_A^{(k)}$ segments. Then A is represented by $\mathbf{A}^{(k)} = \{a_i^{(k)} \mid i = 1, 2, \dots, N_A^{(k)}\}$, where $a_i^{(k)}$ denotes i -th segment at scale k . Similarly, another trajectory B at scale h is represented by $\mathbf{B}^{(h)} = \{b_j^{(h)} \mid j = 1, 2, \dots, N_B^{(h)}\}$.

Next, we chase the cross-scale correspondence of inflection points from top scales to bottom scale. It defines the hierarchy of segments and guarantees the connectivity of segments at different scales. Details of the algorithm for checking segment hierarchy is available in ref. [1]. In order to apply the algorithm for closed curve to open trajectory, we modified it to allow replacement of odd number of segments at sequence ends, since cyclic property of a set of inflection points can be lost.

The main procedure of multiscale matching is to search the best set of segment pairs that satisfies both of the following conditions:

1. Complete Match: By concatenating all segments, the original trajectory must be completely formed without any gaps or overlaps.
2. Minimal Difference: The sum of segment dissimilarities over all segment pairs should be minimized.

The search is performed throughout all scales. For example, in Figure 1, three contiguous segments $a_3^{(0)} - a_5^{(0)}$ at the lowest scale of case *A* can be integrated into one segment $a_1^{(2)}$ at upper scale 2, and the replaced segment well matches to one segment $b_3^{(0)}$ of case *B* at the lowest scale. Thus the set of the three segments $a_3^{(0)} - a_5^{(0)}$ and one segment $b_3^{(0)}$ will be considered as a candidate for corresponding segments. On the other hand, segments such as $a_6^{(0)}$ and $b_4^{(0)}$ are similar even at the bottom scale without any replacement. Therefore they will be also a candidate for corresponding segments. In this way, if segments exhibit short-term similarity, they are matched at a lower scale, and if they present long-term similarity, they are matched at a higher scale.

2.3 Local Segment Difference

In order to evaluate the structural (dis-)similarity of segments, we first describe the structural feature of a segment by using shape parameters defined below.

1. Gradient at starting point: $g(a_m^{(k)})$
2. Rotation angle: $\theta(a_m^{(k)})$
3. Velocity: $v(a_m^{(k)})$

Figure 2 illustrates these parameters. Gradient represents the direction of the trajectory at the beginning of the segment. Rotation angle represents the amount of change of direction along the segment. Velocity represents the speed of change in the segment, which is calculated by dividing segment length by the number of points in the segment.

Next, we define the local dissimilarity of two segments, $a_m^{(k)}$ and $b_n^{(h)}$, as

$$d(a_m^{(k)}, b_n^{(h)}) = \sqrt{\left(g(a_m^{(k)}) - g(b_n^{(h)})\right)^2 + \left(\theta(a_m^{(k)}) - \theta(b_n^{(h)})\right)^2} + \left|v(a_m^{(k)}) - v(b_n^{(h)})\right| + \gamma \left\{cost(a_m^{(k)}) + cost(b_n^{(h)})\right\}$$

where $cost()$ denotes a cost function used for suppressing excessive replacement of segments, and γ is the weight of costs. We define the cost function using local segment

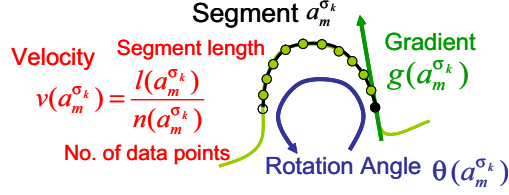


Fig. 2. Segment Parameters.

dissimilarity as follows. For segment $a_m^{(k)}$ that replaces p segments $a_r^{(0)} - a_{r+p-1}^{(0)}$ at the bottom scale,

$$cost(a_m^{(k)}) = \sum_{q=r}^{r+p-1} d(a_q^{(0)}, a_{q+1}^{(0)})$$

2.4 Sequence Dissimilarity

After determining the best set of segment pairs, we newly calculate value-based dissimilarity for each pair of matched segments. The local segment dissimilarity defined in the previous section reflects the structural difference of segments, but does not reflect the difference of original sequence values; therefore, we calculate the value-based dissimilarity that can be further used as a metric for proximity in clustering.

Suppose we obtained L pairs of matched segments after multiscale matching of trajectories A and B . The value-based dissimilarity between A and B , $D_{val}(A, B)$, is defined as follows.

$$D_{val}(A, B) = \sum_{l=1}^L d_{val}(\alpha_l, \beta_l)$$

where α_l denotes a set of contiguous segments of A at the lowest scale that constitutes the l -th matched segment pair ($l \in L$), and β_l denotes that of B . For example, suppose that segments $a_3^{(0)} \sim a_5^{(0)}$ of A and segment $b_3^{(0)}$ of B in Figure 1 constitute the l -th matched pair. Then, $\alpha_l = a_3^{(0)} \sim a_5^{(0)}$ and $\beta_l = b_3^{(0)}$, respectively. $d_{val}(\alpha_l, \beta_l)$ is the difference between α_l and β_l in terms of data values at the peak and both ends of the segments. For the m -th examination ($m \in M$), $d_{val_m}(\alpha_l, \beta_l)$ is defined as

$$d_{val_m}(\alpha_l, \beta_l) = peak_m(\alpha_l) - peak_m(\beta_l) + \frac{1}{2} \{left_m(\alpha_l) - left_m(\beta_l)\} + \frac{1}{2} \{right_m(\alpha_l) - right_m(\beta_l)\}$$

where $peak_m(\alpha_l)$, $left_m(\alpha_l)$, and $right_m(\alpha_l)$ denote data values of the i -th examination at the peak, left end and right end of segment α_l , respectively. If α_l or β_l is composed of plural segments, the centroid of the peak points of those segments is used as the peak of α_l . Finally, d_{val} is integrated over all examinations as follows.

$$d_{val}(\alpha_l, \beta_l) = \frac{1}{M} \sqrt{\sum_m d_{val_m}(\alpha_l, \beta_l)}$$

3 Experimental Results

The usefulness of our method was evaluated through the clustering experiments using two types of datasets. The first dataset contained artificially generated trajectories representing the strokes of nine digits. The second dataset contained real medical data; time series laboratory examination data acquired from chronic hepatitis patients. With artificial data we quantitatively evaluate the property of dissimilarities produced by the proposed method. With medical data we evaluate the practical usefulness of our method for finding groups of interesting (in this case exacerbating) cases.

3.1 Clustering of Synthetic Trajectories

Dataset We have created a total of 90 noisy trajectories representing strokes of nine digits according to the following procedure.

1. Create a base trajectory for each of the nine digits (1,2,...,9) by manually tracing the center line of the display font (we used Arial font for simplicity). Each trajectory is represented as a pair of time series $c(t) = (x(t), y(t))$, where $x(t)$ and $y(t)$ denote horizontal and vertical positions of a point at time t respectively. Time proceeds according to the standard stroke order in Japan; e.g., 1 (one) is written from top to bottom. We modified stroke order of two digits, 4 (four) and 5 (five), so that they can be written by one stroke. Figure 3 provides an example of trajectory for digit 3 (three).

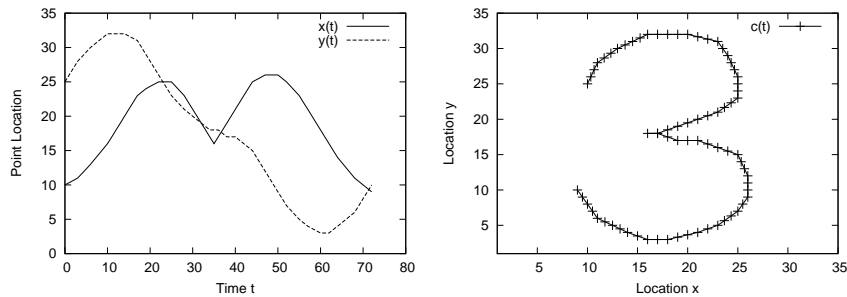


Fig. 3. An example of trajectory for digit 3. Left: time series $x(t)$ and $y(t)$. Right: trajectory $c(t) = (x(t), y(t))$.

2. Add a Gaussian noise $\sim N(0, 1)$ to the position of each point. The noise is added independently to $x(t)$ and $y(t)$; therefore the local shapes of trajectories are disturbed quite largely. Figure 4 provides an example of the trajectory after adding noise. By repeating this process, we generated 10 noisy trajectories for each of the nine base trajectories. Consequently, we obtained a dataset containing a total of 90 noisy trajectories.

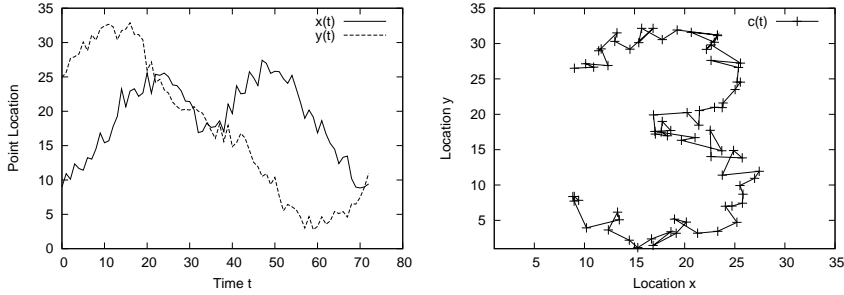


Fig. 4. An example of trajectory after adding noise to Figure 3.

Classification First, we tested whether the proposed method could produce the dissimilarities that can be used for classification of trajectories. A pseudo 1-NN classification experiment was conducted and classification error was evaluated according to the following procedure.

1. Perform pair-wise comparison of 90 trajectories using the proposed method and construct a 90×90 dissimilarity matrix.
2. Select one case, and classify it into the class of the nearest object.
3. Repeat the select/classify step for each of 90 cases and calculate the classification error

We have repeated the above test 100 times using different (disjoint) sets of trajectories. The error ratio was 0.016 ± 0.014 (mean \pm SD), which means $> 98\%$ of trajectories were classified correctly using the dissimilarities produced by the proposed method.

Cluster Rebuild Next, we tested whether the dissimilarities can be used to correctly form the original structure of the data. The procedure was as follows: (1) using the 90×90 dissimilarity matrix, rebuild clusters fixing the cluster number=9. and (2) evaluate the grouping error ratio with regard to digit labels determined by voting. We used group average hierarchical clustering [3] as the grouping method. Similarly to classification experiment, this procedure was repeated 100 times using different sets of trajectories.

The error ratio was 0.118 ± 0.057 , which means about 88% of trajectories were grouped correctly. Figure 5 shows an example of the dendrogram that resulted in incorrect grouping. The horizontal line shows cutting level for nine clusters. The notions c_1, c_2, \dots, c_9 represent the cluster number respectively. The notions 6, 8, \dots , 1 at the bottom of the dendrogram represent correct groups, namely, labels of digits that the trajectories should represent.

Each of the clusters except for c_3 , c_7 and c_8 contained 10 trajectories that correctly represented the same digit. Cluster c_3 contained a total of 20 trajectories, half representing digit 5 (five) and the remaining half representing 9 (nine). Figure 5 provides the first 16 of trajectories in c_3 . The stroke of 9 (nine) starts from the junction at center right. The stroke of 5 (five) should start from upper left, however, we modified its

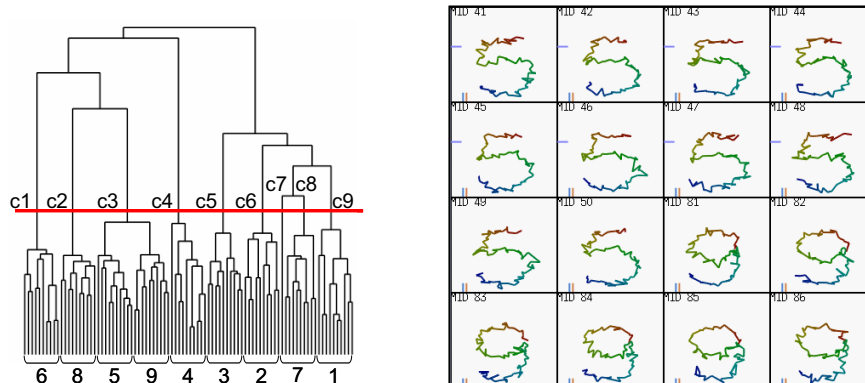


Fig. 5. Left: An example of dendrogram resulted in incorrect grouping. Right: Trajectories in cluster 3.

stroke order to start from upper right so that it could be written in one stroke. As a result, the base trajectories for these two digits look very similar. They were grouped into the same cluster under this nine clusters solution, however, the dendrogram shows they could be separated correctly at relatively low values of dissimilarity. Trajectories representing the digit 7 (seven) were separated into two clusters, c_7 with 1 case, and c_8 with remaining 9 cases, though they could be merged at the next step. This seemed to occur because the method failed to find the good match for the one case in c_7 under the given parameters.

3.2 Clustering of Trajectories of Medical Data

Data and Procedure of Experiments We applied our method to the chronic hepatitis dataset which was a common dataset in ECML/PKDD discovery challenge 2002-2004. The dataset contained time series laboratory examinations data collected from 771 patients of chronic hepatitis B and C. According to the previous study conducted by Hirano et al. [4], we focused on the analysis about temporal relationships between albumin (ALB) and platelet count (PLT), and their relation to fibrotic stages. The subjects were 99 cases of Type C viral hepatitis who did not receive the interferon (IFN) treatment.

We performed cluster analysis on ALB-PLT trajectories according to the following procedure. (1) Select a pair of cases (patients) and calculate the dissimilarity by using the proposed method. (2) Apply this procedure for all pairs of cases and construct a dissimilarity matrix. (3) Create a dendrogram by using conventional hierarchical clustering [3] and the dissimilarity matrix. Then perform cluster analysis.

3.3 Results

Figure 6 left shows the dendrogram generated from ALB-PLT trajectories. The overall shape of the dendrogram implied the existence of a few major clusters; however, in order to carefully examine the data structure, we here avoided excessive merge of clusters

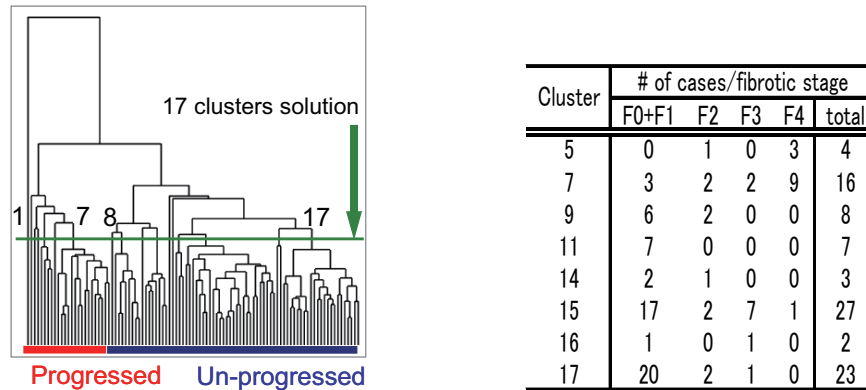


Fig. 6. Left: Dendrogram. Right: Cluster constitutions w.r.t. fibrotic stages (Small clusters of $N < 2$ omitted.)

and determined to split it into 17 clusters where dissimilarity increased relatively large at early stage. For each of the 8 clusters that contained ≥ 2 cases, we classified cases according to the fibrotic stage. Figure 6 right shows the summary. The leftmost column shows cluster number. The next column shows the number of cases whose fibrotic stages were F0 or F1. The subsequent three columns show the number of F2, F3, and F4 cases respectively. The rightmost column shows the total number of cases in each cluster. Globally, it could be recognized that clusters can be classified into one of the two categories: one containing progressed cases of liver fibrosis (clusters 5 and 7) and another containing un-progressed cases (clusters 9, 11, 14, 15, 16 and 17). This implied that the difference about ALB and PLT might be related to the fibrotic stages.

Figures 7 and 8 show the examples of grouped ALB-PLT trajectories. Each quadrature region contains a trajectory of ALB-PLT values for a patient. If the number of cases in a cluster was larger than 16, the first 16 cases w.r.t. ID number were selected for visualization. The bottom part of Figure 7 provides the legend. The horizontal axis represents ALB value, and the vertical axis represents PLT value. Lower end of the normal range (ALB:3.9g/dl, PLT:120 $\times 10^3$ /ul) and Upper end of the normal range (ALB:5.0g/dl, PLT:350 $\times 10^3$ /ul) were marked with blue and red short lines on each axis respectively. Time phase on each trajectory was represented by color phase: red represents the start of examination, and it changes toward blue as time proceeds.

Figure 7 shows cases grouped into cluster 5 which contained remarkably many F4 cases (3/4). The skewed trajectory of ALT and PLT clearly demonstrated that both values decreased from the normal range to the lower range as time proceeded, due to the dysfunction of the liver, as observed in the exacerbating cases. Cluster 7, shown in Figure 8 left, also contained similarly large number of progressed cases (F4:9/16, F3:2/16) and exhibited the similar characteristics, though it was relatively weaker than in cluster 5. On the contrary, clusters that contained many un-progressed cases exhibited different characteristics. Figure 8 right shows the trajectories grouped into cluster 17, where the

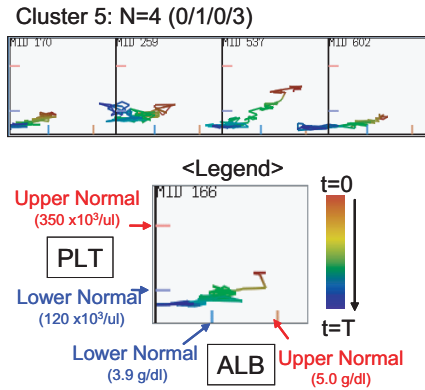


Fig. 7. Trajectories in Cluster 5.

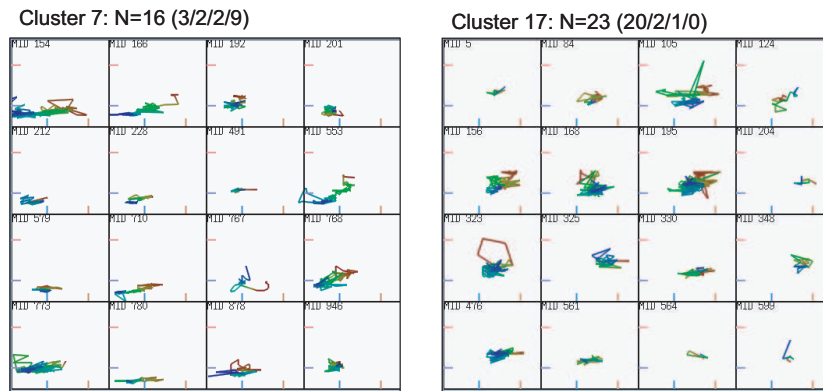


Fig. 8. Left: Trajectories in Cluster 7. Right Trajectories in Cluster 17.

number of F0/F1 cases was large (20/23). Most of the trajectories moved within the normal range, and no clear feature about time-direction dependency was observed.

The above observation suggested the existence of two major classes on the dendrogram: progressed cases and un-progressed cases. In order to confirm this hypothesis, we have investigated the number of cases that reached abnormally low PLT level. As PLT level is considered to correlate with fibrotic stage [5], its decrease to the abnormally low level would imply that liver fibrosis progressed to F4. The judgment of reaching F4 was based on the following assumption [4] : “If the PLT level of a patient is continuously lower than the normal range for at least 6 months, and after that never keeps normal range more than 6 months, then the patient is F4.”

Table 1 provides the results. The left subtable summarizes the results for clusters 1-7, which were grouped into the left branch on the dendrogram in Figure 6. The right subtable are those for cluster 8-17, which were grouped into the right branch of the

Table 1. Comparison of clusters w.r.t. the number of cases that reached abnormally low PLT level

Cluster	Number of Cases		Cluster	Number of Cases	
	Reached	Total		Reached	Total
1	0	1	8	0	1
2	1	1	9	0	8
3	1	1	10	0	1
4	0	1	11	0	7
5	4	4	12	0	1
6	1	1	13	1	1
7	12	16	14	2	3
Total	19	25	15	4	27
			16	0	2
			17	0	23
			Total	7	74

dendrogram. It can be clearly observed that the ratio of cases that reached abnormally low PLT level was remarkably higher in the left cluster (19/25) than in the right cluster (7/74). This demonstrated that the proposed method could discover the interesting feature of ALB-PLT trajectories; there might exist two major categories representing progressed cases and un-progressed cases, and the trajectories of progressed cases might take the skewed shapes representing their exacerbating patterns.

4 Conclusions

In this paper, we have presented a novel cluster analysis method for time-series medical data and its application to finding groups of exacerbating cases in chronic hepatitis. Our method employed a two-stage approach. Firstly, it compared two trajectories based on their structural similarity, and determines the best correspondence of partial trajectories. After that, it calculated the value-based dissimilarity for the all pairs of matched segments, and outputs the total sum as dissimilarity of the two trajectories.

Experimental results on digit stroke data showed that the method could produce dissimilarities that can be used for classification/clustering of trajectories. Experiments on chronic hepatitis data showed that the method could discover the groups of exacerbating cases based on the trajectories of ALB-PLT. Although further examination is needed, the results may open a new way to knowledge discovery in medical data, that is, the use of data-driven, time-series clustering results as a new decision class for rule discovery.

Our future work include the clinical validation of the results, treatment of the high dimensional data, and refinement of segment parameters.

Acknowledgments

This work was supported in part by the Grant-in-Aid for Scientific Research on Priority Area “Knowledge Discovery from Multidimensional Trajectory Data and Its Application to Medicine” (#19024060) by the Ministry of Education, Culture, Science and Technology of Japan.

References

1. N. Ueda and S. Suzuki (1990): A Matching Algorithm of Deformed Planar Curves Using Multiscale Convex/Concave Structures. *IEICE Transactions on Information and Systems*, J73-D-II(7): 992–1000.
2. T. Lindeberg (1990): Scale-Space for Discrete Signals. *IEEE Trans. PAMI*, 12(3):234–254.
3. B. S. Everitt, S. Landau, and M. Leese (2001): *Cluster Analysis Fourth Edition*. Arnold Publishers.
4. S. Tsumoto, S. Hirano, and K. Takabayashi (2005): Development of the Active Mining System in Medicine Based on Rough Sets. *Journal of Japan Society of Artificial Intelligence* 20(3):203-210.
5. H. Matsumura and M. Moriyama and I. Goto and N. Tanaka, and H. Okubo and Y. Arakawa (2000): Natural course of progression of liver fibrosis in patients with chronic liver disease type C in Japan - a study of 527 patients at one establishment in Japan. *J. Viral Hepat.* 7:375–381.

Improving Boosting by exploiting former assumptions

Emna Bahri, Nicolas Nicoloyannis, and
Mondher Maddouri

Laboratoire Eric, University Lyon 2.
5 avenue Pierre Mendès France, 69676 Bron Cedex
{e.bahri,nicolas.nicoloyannis}@univ-lyon2.fr
mondher.maddouri@fst.rnu.tn
<http://eric.univ-lyon2.fr>

Abstract. The error reduction in generalization is one of the principal motivations of research in machine learning. Thus, a great number of work is carried out on the classifiers aggregation methods in order to improve generally, by voting techniques, the performance of a single classifier. Among these methods of aggregation, we find the Boosting which is most practical thanks to the adaptive update of the distribution of the examples aiming at increasing in an exponential way the weight of the badly classified examples. However, this method is blamed because of overfitting, and the convergence speed especially with noise. In this study, we propose a new approach and modifications carried out on the algorithm of AdaBoost. We will demonstrate that it is possible to improve the performance of the Boosting, by exploiting assumptions generated with the former iterations to correct the weights of the examples. An experimental study shows the interest of this new approach, called hybrid approach.

Key words: Machine learning, Data mining, Classification, Boosting, Recall, convergence

1 Introduction

The great emergence of the modern databases and their evolution in an exponential way as well as the evolution of transmission systems result in a huge mass of data which exceeds the human processing and understanding capabilities. Certainly, these data are sources of relevant information and require means of synthesis and interpretation. As a result, researches were based on powerful systems of artificial intelligence allowing the extraction of useful information helping us in decisions making. Responding to this need, data mining was born. It drew its tools from the statistics and databases. The methodology of data mining gives the possibility to build a model of prediction. This model is a phenomenon starting from other phenomena more easily accessible, based on the process of the knowledge discovery from data which is a process of intelligent

data classification. However, the built model can sometimes generate errors of classification that even a random classification does not make. To reduce these errors, a great amount of research in data mining and specifically in machine learning has been carried out on classifiers aggregation methods having as goal to improve by voting techniques the performance of a single classifier. These aggregation methods are good for compromised Skew-variance, thanks to the three fundamental reasons explained in [6]. These methods of aggregation are divided into two categories. The first category refers to those which merge preset classifiers, such as simple voting [2], the weighted voting [2], and the weighted majority voting [13]. The second category consists of those which merge classifiers according to data during the training, such as adaptive strategies (Boosting) and the basic algorithm AdaBoost [22] or random strategies (Bagging) [3]. We are interested in the method of Boosting, because of the comparative study [7] that shows, in little noise, AdaBoost is seemed to be working against the overfitting. In fact, AdaBoost tries to optimize directly the weighted votes. This observation has been proved not only by the fact that the empirical error on the training set decreases exponentially with iterations, but also by the fact that the error in generalization also decreases, even when the empirical error reached its minimum. However, this method is blamed because of overfitting, and the speed of convergence especially with noise. In the last decade, many studies focused on the weaknesses of AdaBoost and proposed its improvement. The important improvements were carried on the modification of the weight of examples [20], [19], [1], [21], [15], [9], the modification of the margin [10], [21], [18], the modification of the classifiers' weight [16], the choice of weak learning [5], [25] and the speed of convergence [23], [14], [19]. In this paper, we propose a new improvement to the basic Boosting algorithm AdaBoost. This approach aims exploiting assumptions generated with the former iterations of AdaBoost to act both on the modification of the weight of examples and the modification of the classifiers' weight. By exploiting these former assumptions, we think that we will avoid the re-generation of a same classifier within different iterations of AdaBoost. Thus, consequently, we expect a positive effect on the improvement of the speed of convergence. The paper is organized in three sections. In the following section, we describe the studies whose purpose is to improve the Boosting against its weaknesses. In the third section, we describe our improvement of boosting by exploiting former assumptions. In the fourth section, we present an experimental study of the proposed improvement by comparing its error in generalization, its recall and its speed of convergence with AdaBoost, on many real databases. We study also the behavior of the proposed improvement on noisy data. We present also comparative experiments of our proposed method with BrownBoost (a new method known that it improves AdaBoost M1 with noisy data). Lastly, we give our conclusions and perspectives.

2 State of art

Due to the finding of some weaknesses, such as the overfitting and the speed of convergence, met by the basic algorithm of boosting AdaBoost, several researchers have tried to improve it. Therefore, we make a study of main methods having as purpose to improve boosting relatively to these weaknesses. With this intention, the researchers try to use the strong points of Boosting such as the update of the badly classified examples, the maximization of the margin, the significance of the weights that AdaBoost associates the hypothesis and finally the choice of weak learning.

2.1 Modification of the examples' weight:

The distributional adaptive update of the examples, aiming at increasing the weight of those badly learned by the preceding classifier, makes it possible to improve the performance of any training algorithm. Indeed, with each iteration, the current distribution supports the examples having been badly classified by the preceding hypothesis, which characterizes the adaptivity of AdaBoost. As a result, several researchers proposed strategies related to a modification of weight update of the examples, to avoid the overfitting.

Indeed, we can quote for example MadaBoost [9] whose aim is to limit the weight of each example by its initial probability. It acts thus on the uncontrolled growth of the weight of certain examples (noise) which is the problem of overfitting.

Another approach which make the algorithm of boosting resistant to the noise is Brownboost [15], an algorithm based on Boost-by-Majority by incorporating a time parameter. Thus for an appropriate value of this parameter, BrownBoost is able to avoid the overfitting. Another approach, which adapts to AdaBoost a logistic regression model, is Logitboost [19].

An approach, which produces less errors of generalization compared with the traditional approach but with the cost of an error of training slightly more raised, is the Modest boost [1]. In fact, its update is based on the reduction in the contribution of classifier, if that functions "too well" on the data correctly classified. This is why the method is called Modest AdaBoost - it forces the classifiers to be "modest" and it works only in the field defined by a distribution.

An approach, which tries to reduce the effect of overfitting by imposing limitations on the distribution produced during the process of boosting is used in SmoothBoost [21]. In particular, a limited weight is assigned to each example individually during each iteration. Thus, the noisy data can be excessively underlined during the iterations since they are assigned to the extremely large weights.

A last approach, Iadaboost [20], is based on the idea of building around each example a local information measurement, making it possible to evaluate the overfitting risks, by using neighboring graph to measure information around each example. Thanks to these measurements, we have a function which translates

the need for updating the example. This function makes it possible to manage the outliers and the centers of clusters at the same time.

2.2 Modification of the margin:

Certain studies, analyzing the behavior of Boosting, showed that the error in generalization still decreases even when the errors in training are stable. The explanation is that even if all the examples of training are already well classified, Boosting tends to maximize the margins [21].

Following this, some studies try to modify the margin either by maximizing it or by minimizing it with the objective of improving the performance of Boosting against overfitting.

Several approaches followed such as AdaBoostReg [18] which tries to identify and remove badly labeled examples, or to apply the constraint of the maximum margin to examples supposed to be badly labeled, by using the Soft Margin.

In the algorithm, proposed by [10], the authors use a weighting diagram which exploits a margin function that grows less quickly than the exponential function.

2.3 Modification of the classifiers' weight:

During the performance evaluation of Boosting, researchers wondered about the significance of the weights $\alpha(t)$ that AdaBoost associates with the produced hypotheses.

However, they noted at the time of experiments on very simple data that the error in generalization decreased further whereas the weak learning had already provided all the possible hypotheses. In other words, when a hypothesis appears several times, it votes finally with a weight, office sum of all $\alpha(t)$, which is perhaps absolute. So several researchers hoped to approach these values by a nonadaptive process , such as locboost [16] an alternative to the construction of the whole representations of experts which allows the coefficients $\alpha(t)$ to depend on the data.

2.4 Choice of weak learner :

A question that several researchers posed against the problems of boosting is that of weak learner and how to make a good choice of this classifier?

A lot of research moves towards the study of choosing the basic classifier of boosting, such as GloBoost [25]. This approach use a weak learner which produces only correct hypotheses. RankBoost [5] is also an approach which is based on weak learner which accepts as data attributes functions of rank.

2.5 The speed of convergence

In addition to the problem of overfitting met by boosting in the modern databases mentioned above, we find another problem : the speed of convergence of Boosting especially AdaBoost.

Indeed, in the presence of noisy data, the optimal error of the training algorithm used is reached after a long time. In other words, AdaBoost "loses" iterations, and thus time, with reweighing examples which do not deserve in theory any attention, since it is a noise.

Thus research was made to detect these examples and improve the performance of Boosting in terms of convergence such as: iBoost [23] which aims at specializing weak hypotheses on the examples supposed to be correctly classified.

The IAdaBoost approach also contributes to improve AdaBoost against its speed of convergence. In fact, the basic idea of the improvement is the modification of the theorem [19]. This modification is carried out in order to integrate the risk of Bayes. The effects of this modification are a faster convergence towards the optimal risk and a reduction of the number of weak hypotheses to build. Finally, RegionBoost [14] is a new weighting strategy of each classifier. This weighting is evaluated at the voting time by a technique based on K Nearest Neighbors of the example to label. This approach makes it possible to specialize each classifier on areas of the training data.

3 Boosting by exploiting former assumptions

To improve the performance of AdaBoost and to avoid forcing it to learn either from the examples that contain noise, or from the examples which would become too difficult to learn during the process of Boosting, we propose a new approach. This approach is based on the fact that for each iteration, Adaboost, builds hypotheses on a defined sample, it makes its updates and it calculates the error of training according to the results given only by these hypotheses. In addition, it does not exploit the results provided by the hypotheses already built on other samples to the former iterations. This approach is called AdaBoostHyb

Program Code Input X_0 to classify , $S = (x_1, y_1), \dots, (x_n, y_n)$ Sample

- For $i=1, n$ Do
- $p_0(x_i) = 1/n$;
- End FOR
- $t \leftarrow 0$
- While $t \leq T$ Do
- Learning sample S_t from S with probabilities p_t .
- Build a hypotheses h_t on S_t with weak learning A.
- ϵ_t apparent error of h_t on S with $\epsilon_t = \sum \text{weight of examples}$
such that $\text{argmax}(\sum_{i=1}^t \alpha_i h_i(x_i) \neq y_i)$. $\alpha_t = 1/2 \ln((1 - \epsilon_t)/\epsilon_t)$.
- For $i=1, m$ Do
- $P_{t+1}(x_i) \leftarrow (p_t(x_i)/Z_t)e^{-\alpha_t}$ **if** $\text{argmax}(\sum_{i=1}^t \alpha_i h_i(x_i)) = y_i$ (**correctly classified**)
- $P_{t+1}(x_i) \leftarrow (p_t(x_i)/Z_t)e^{+\alpha_t}$ **if** $\text{argmax}(\sum_{i=1}^t \alpha_i h_i(x_i)) \neq y_i$ (**badly classified**)
- (Z_t normalized to $\sum_{i=1}^n p_t(x_i) = 1$)

- End For
- $t \leftarrow t + 1$
- End While
- Final hypotheses :
- $H(x) = \operatorname{argmax}_{y \in Y} \sum_{t=1}^T \alpha_t$

The modification within the algorithm is made

During the modification of the weights of the examples: Indeed, this strategy, with each iteration, is based on the opinion of the experts already used (hypotheses of the former iterations) for the update of the weight of the examples.

In fact, we do not compare only the class predicted by the hypothesis of the current iteration with the real class but also the sum of the hypotheses balanced from the first iteration to the current iteration. If this sum votes for a class different from the real class, an exponential update such as in the case of AdaBoost is applied to the badly classified example. Thus, this modification lets the algorithm be interested only in the examples which are either badly classified or not classified yet. So, results related to the improvement the speed of convergence are awaited, similarly for the reduction of the error of generalization , because of the richness of the space of hypotheses to each iteration.

During the error analysis $\epsilon(t)$ of the hypothesis to the iteration T: Indeed, this other strategy is rather interested in the classifiers' coefficient (hypothesis) to each iteration $\alpha(t)$.

In fact, this coefficient depends on the apparent error analysis $\epsilon(t)$. This method, with each iteration, takes into account hypotheses preceding the current iteration during the calculation of $\epsilon(t)$. So the apparent error with each iteration is the weight of the examples voted badly classified by the hypotheses weighted of the former iterations by comparison to the real class .

Results in improving the error of generalization are expected since the vote of each hypothesis (coefficient $\alpha(t)$) is calculated from the other hypotheses.

4 Experiments

The objective of this part is to compare our new approach and especially its contribution with the original approach of Adaboost and to look further into this comparison by the choice of a version improved of Adaboost (BrownBoost [15]).

Our Choice of BrownBoost was based on its robustness against the problems of noisy data. In fact, BrownBoost is an adaptive algorithm which incorporates a time parameter that corresponds to the proportion of noise in the training data. So by a good estimation of this parameter BrownBoost is capable of avoiding overfitting. The comparison criterions chosen in this article are the error rate, the recall, the speed of convergence and the sensitivity to noise.

To do this experimental comparison, we used the C4.5 algorithm as a weak learner (according to the study of Dietterich [6]). To estimate without skew

the theoretical success rate, we used a procedure of cross-validation in 10 folds (according to the study [12]). In order to choose the databases for our experiments, we considered the principle of diversity . We have considered 15 databases of the UCI [8]. Some databases are characterized by their missing values (NHL, Vote, Hepatitis, Hypothyroid). Some others concern the problem of multi-class prediction (Iris: 3 classes, Diabetes: 4 classes, Zoo: 7 classes, IDS: 12 classes). We choose the IDS database [24] especially because it has 35 attributes. Table 1 describes the 15 databases used in the experimental comparison.

Databases	Nb. Inst	Attrib	Cl. Pred	Miss.VaL
IRIS	150	4 numeric	3	no
NHL	137	8 numeric and symbolic	2	yes
VOTE	435	16 boolean valued	2	yes
WEATHER	14	4 numeric and symbolic	2	no
CREDIT-A	690	16numeric and symbolic	2	yes
TITANIC	750	3 symbolic	2	no
DIABETES	768	8 numeric	2	no
HYPOTHYROID	3772	30 numeric and symbolic	4	yes
HEPATITIS	155	19 numeric and symbolic	2	yes
CONTACT-LENSES	24	4 nominal	3	no
ZOO	101	18 numeric and boolean	7	no
STRAIGHT	320	2 numeric	2	no
IDS	4950	35 numeric and symbolic	12	no
LYMPH	148	18 numeric	4	no
BREAST-CANCER	286	9 numeric and symbolic	2	yes

Table 1. Databases Description

4.1 Comparison of generalization error

Table 2 indicated the error rates in 10-fold cross-validation corresponding to the algorithm AdaBoost M1, BrownBoost and the proposed one. We used the same samples for the tree algorithms in cross-validation for comparison purposes. The results are obtained while having chosen for each algorithm to carry out 20 iterations. The study of the effect of the number of iterations on the error rates of the tree algorithms will be presented in the section 4.3, where we will consider about 100 iterations.

The results in table 2 show already that the proposed modifications improve the error rates of AdaBoost. Indeed, for 14 databases out of 15, the proposed algorithm shows an error rate lower or equal to AdaBoost M1. We remark, also, a significant improvement of the error rates corresponding to the three databases NHL, CONTACT-LENS and BREAST-CANCER. For example, the error rate corresponding to the BREAST-CANCER database goes from 45.81% to 30.41%.

Even, if we compare the proposed algorithm with BrownBoost, we remark that for 11 databases out of 15 the proposed algorithm shows an error rate lower or equal to BrownBoost.

This gain shows that by exploiting hypotheses generated with the former iterations to correct the weights of the examples, it is possible to improve the

performance of the Boosting. This can be explained by the calculation of the precision of the error analysis $\epsilon(t)$ and consequently the calculation of the coefficient of the classifier $\alpha(t)$ as well as the richness of the space of the hypotheses to each iteration since it acts on the whole of the hypotheses generated by the preceding iterations and the current iteration.

Databases	AdaBoost M1	BrownBoost	AdaBoostHyb
IRIS	6.00%	3.89	3.00%
NHL	35.00%	30.01	28.00%
VOTE	4.36%	4.35	4.13%
WEATHER	21.42%	21.00	21.00%
CREDIT-A	15.79%	13.00	13.91%
TITANIC	21.00 %	24.00	21.00%
DIABETES	27.61%	25.05	25.56%
HYPOTHYROID	0.53%	0.6	0.42%
HEPATITIS	15.62%	14.10	14.00%
CONTACT-LENSES	25.21%	15.86	16.00%
ZOO	7.00%	7.23	7.00%
STRAIGHT	2.40%	2.00	2.00%
IDS	1.90%	0.67	0,37%
LYMPH	19.51%	18.54	20.97%
BREAST-CANCER	45.81%	31.06	30.41%

Table 2. Rate of error of generalization

4.2 Comparison of recall

The encouraging results, found previously, enable us to proceed further within the study of this new approach. Indeed, in this part we try to find out the impact of the approach on the recall, since our approach does not really improve Boosting if it acts negatively on the recall.

Table 3 indicates the recall for the algorithms AdaBoost M1, Brownboost and the proposed one. We remark that the proposed algorithm has the best recall overall the 14 for 15 studied databases. This result confirms the preceding ones. We remark also that it increases the recall of the databases having less important error rates.

Considering Brownboost, we remark that it improves the recall of AdaBoostM1, overall the data sets (except the TITANIC one). However, the recall rates given by our proposed algorithm are better than those of BrownBoost. Except, with the zoo dataset.

It is also noted that our approach improves the recall in the case of the Lymph base where the error was more important. It is noted though that the new approach does not act negatively on the recall but it improves it even when it can not improve the error rates.

4.3 Comparison with noisy data

In this part, we are based on the study already made by Dietterich [6] by adding random noise to the data. This addition of noise of 20% is carried out, for each

Databases	AdaBoost M1	BrownBoost	AdaBoostHyb
IRIS	0,93	0,94	0,96
NHL	0,65	0,68	0,71
VOTE	0,94	0,94	0,95
WEATHER	0,63	0,64	0,64
CREDIT-A	0,84	0,85	0,86
TITANIC	0,68	0,54	0,68
DIABETES	0,65	0,66	0,68
HYPOTHYROID	0,72	0,73	0,74
HEPATITIS	0,69	0,70	0,73
CONTACT-LENSES	0,67	0,75	0,85
ZOO	0,82	0,9	0,82
STRAIGHT	0,95	0,95	0,97
IDS	0,97	0,97	0,98
LYMPH	0,54	0,62	0,76
BREAST-CANCER	0,53	0,55	0,6

Table 3. Rate of recall

one of these databases, by changing randomly the value of the predicted class by another possible value of this class.

Table 3 shows us the behavior of the algorithms with noise. We notice that the hybrid approach is also sensitive to the noise since the error rate in generalization is increased for all the databases.

However this increase remains always inferior with that of the traditional approach except for the databases such as Credit-A, Hepatitis and Hypothyroid.

So, we studied these databases and we observed that all these databases have missing values. In fact, Credited, Hepatitis and Hypothyroid have respectively 5%, 6% and 5,4% of missing values. It seems that our improvement loses its effect with accumulation of two types of noise: missing values and artificial noise, although the algorithm *AdaBoostHyb* improves the performance of *AdaBoost* against the noise. Considering *Brownboost*, we remark that it gives better error rates than *AdaboostM1* on all the noisy data sets. However, It gives better error rates than our proposed method, only with 6 data sets. Our proposed method gives better error rates with the other 9 data sets. This encourages us to study in details the behavior of our proposed method on noisy data.

4.4 Comparison of convergence speed

In this part, we are interested in the number of iterations that allow the algorithms to converge, i.e. where the error rate is stabilized. Tables 4, 5 and 6 shows us that the hybrid approach allows *AdaBoost* to converge more quickly. Indeed, the error rate of *AdaBoost M1* is not stabilized even after 100 iterations, whereas *Adaboost Hyb* converges after 20 iterations or even before.

For this reason we choose for the first part 20 iterations to carry out the comparison in terms of error and recall. These results are also valid for the database Hepatitis. In fact, This database has a lot of missing values (Rate 6%). These missing values always present a problem of convergence. Moreover, the same results appear on databases of various types (several attributes, the class to be predicted with K modalities, important sizes).

Databases	AdaBoost M1	BrownBoost	AdaBoostHyb
IRIS	33.00%	26.00	28.00%
NHL	45.00%	40.00	32.00%
VOTE	12.58%	7.00	7.76%
WEATHER	25.00%	22	21%
CREDIT-A	22.56%	20.99	24.00%
TITANIC	34.67%	28.08	26.98%
DIABETES	36.43%	32.12	31.20%
HYPOTHYROID	0.92%	0.86	2.12%
HEPATITIS	31.00%	27.38	41.00%
CONTACT-LENSES	33%	30.60	25%
ZOO	18.84%	14.56	11.20%
STRAIGHT	3.45%	2.79	2.81%
IDS	2.40%	1.02	0.50%
LYMPH	28.73%	24.57	24.05%
BREAST-CANCER	68.00%	50.98	48.52%

Table 4. Rate of error on Noisy data

This makes us think that due to the way of calculating the apparent error, the algorithm reaches stability more quickly. Finally, we remark that BrownBoost does'nt converge even after 1000 iterations. This remark prove the fact that the BrownBoost problem is the speed of convergence.

-	AdaBoost M1				BrownBoost				AdaBoost hyb			
	10	20	100	1000	10	20	100	1000	10	20	100	1000
Nb. iterations	10	20	100	1000	10	20	100	1000	10	20	100	1000
Iris	7,00	6,00	5,90	5,85	3,96	3,89	3,80	3,77	3,50	3,00	3,00	3,00
Nhl	37,00	35,00	34,87	34,55	30,67	30,01	29,89	29,76	31,00	28,00	28,00	28,00
Weather	21,50	21,42	21,40	14,40	21,10	21,00	20,98	21,95	21,03	21,00	21,00	21,00
Credit-A	15,85	15,79	15,75	14,71	13,06	13,00	12,99	12,97	14,00	13,91	13,91	13,91
Titanic	21,00	21,00	21,00	21,00	24,08	24,00	23,89	23,79	21,00	21,00	21,00	21,00
Diabetes	27,70	27,61	27,55	27,54	25,09	25,05	25,03	25,00	25,56	25,56	25,56	25,56
Hypothyroid	0,60	0,51	0,51	0,50	0,62	0,60	0,59	0,55	0,43	0,42	0,42	0,42
Hepatitis	16,12	15,60	14,83	14,19	14,15	14,10	14,08	14,04	14,03	14,00	14,00	14,00
Contact-Lenses	26,30	24,80	24,50	16,33	15,90	15,86	15,83	15,80	16,00	16,00	16,00	16,00
Zoo	7,06	7,00	7,00	7,00	7,25	7,23	7,19	7,15	7,00	6,98	7,00	7,00
Straight	2,50	2,46	2,45	2,42	2,12	2,00	1,98	1,96	0,42	0,42	0,42	0,42
IDS	2,00	1,90	1,88	1,85	0,7	0,67	0,65	0,63	0,7	0,67	0,65	0,63
Lymph	19,53	19,51	19,51	19,50	18,76	18,54	18,50	18,45	18,76	18,54	18,50	18,45
Breast-Cancer	45,89	45,81	45,81	45,79	31,10	31,06	31,04	31,00	31,10	31,06	31,04	31,00

Table 5. comparison of speed convergence

Conclusion

In this paper, we proposed an improvement of AdaBoost which is based on the exploitation of the hypotheses already built with the preceding iterations. The experiments carried out and the results show that this approach improves the performance of AdaBoost in error rate, in recall, in speed of convergence and in sensibility to the noise. However, it proved that this same approach remains sensitive to the noise.

We did an experimental comparison of the proposed method with BrownBoost (a new method known that it improves AdaBoost M1 with noisy data). The results show that our proposed method improves the recall rates and the speed of convergence of BrownBoost overall the 15 data sets. The results show also

that BrownBoost gives better error rates with some datasets, and our method gives better error rates with other data sets. The same conclusion is reached with noisy data.

To confirm the experimental results obtained, more experimentations are planned. We are working on further databases that were considered by other researchers in their studies of the boosting algorithms. We plan to choose weak learning methods other than C4.5, in order to see whether the obtained results are specific to C4.5 or general. We plan to compare the proposed algorithm to new variants of boosting, other than AdaBoost M1. We can consider especially those that improve the speed of convergence like IAdaBoost and RegionBoost. In the case of encouraging comparisons, a theoretical study on convergence will be done to confirm the results of the experiments.

Another objective which seems important to us consists in improving this approach against the noisy data. In fact, the emergence and the evolution of the modern databases force the researchers to study and improve the boosting's capacities of tolerance to the noise. Indeed, these modern databases contain a lot of noise, due to new technologies of data acquisition such as the Web. In parallel, studies such as [5], [17] and [19], show that AdaBoost tends to overfit the data and especially the noise. So, a certain number of recent work tried to limit these risks of overfitting. These improvements are based primarily on the concept that AdaBoost tends to increase the weight of the noise in an exponential way. Thus two solutions were proposed to reduce the sensibility to noise. One is by detecting these data and removing them based on the heuristic and selection of prototypes such as research presented in [4] and [26]. The other solution is by detecting these data through the process of boosting, in which case we speak about a good management of noise. According to the latest approach, we plan to improve the proposed algorithm against the noisy data, by using neighboring graphs or using update parameters.

Finally, a third perspective work aims at studying the Boosting with a weak learner that generates several rules (Rule learning [11]). Indeed, the problem of this type of learners is the production of conflicting rules within the same iteration of boosting. These conflicting rules will have the same weights (attributed by the boosting algorithm). In the voting procedure, we are thinking about a combination of the global weights (those attributed by the boosting algorithm) and the local weights (those attributed by the learning algorithm).

References

1. Vladimir Vezhnevets Alexander Vezhnevets. Modest adaboost: Teaching adaboost to generalize better. *Moscow State University*, 2002.
2. E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 24:173–202, 1999.
3. L. Breiman. Bagging predictors. *Machine Learning*, 26:123–140, 1996.
4. C. E. Brodley and M. A. Friedl. Identifying and eliminating mislabeled training instances. In *AAAI/IAAI, Vol. 1*, pages 799–805, 1996.
5. R. Dharmarajan. An efficient boosting algorithm for combining preferences. Technical report, MIT, September 1999.

6. T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, pages 1–22, 1999.
7. T. G. Dietterich. Ensemble methods in machine learning. *First International Workshop on Multiple Classifier Systems*, pages 1–15, 2000.
8. C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. Uci repository of machine learning databases, 1998.
9. C. Domingo and O. Watanabe. Madaboost: A modification of adaboost. In *Proc. 13th Annu. Conference on Comput. Learning Theory*, pages 180–189. Morgan Kaufmann, San Francisco, 2000.
10. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Dept. of Statistics, Stanford University Technical Report.*, 1998.
11. Friedman J. H and Popescu B. E. Predictive learning via rule ensembles (technical report). *Stanford University*, (7), 2005.
12. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *International Joint Conference on Artificial Intelligence (IJCAI)*, 1995.
13. N. Littlestone and M. K. Warmuth. The weighted majority algorithm. In *Information and computation*, volume 24, pages 212–261, 1994.
14. R. Maclin. Boosting classifiers regionally. In *AAAI/IAAI*, pages 700–705, 1998.
15. R. McDonald, D. Hand, and I. Eckley. An empirical comparison of three boosting algorithms on real data sets with artificial class noise, 2003.
16. Ron Meir, Ran El-Yaniv, and Shai Ben-David. Localized boosting. In *Proc. 13th Annu. Conference on Comput. Learning Theory*, pages 190–199. Morgan Kaufmann, San Francisco, 2000.
17. G. Ratsch. Ensemble learning methods for classification. *Master's thesis, Dep of computer science, University of Potsdam*, April 1998.
18. G. Rtsch, T. Onoda, and K.-R. Miller. Soft margins for adaboost. *Mach. Learn.*, 42(3):287–320, 2001.
19. R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence rated predictions. *Machine Learning*, 37(3):297–336, 1999.
20. Marc Sebban and Henri-Maxime Suchier. tude sur amlioration du boosting : rduction de l'erreur et acclration de la convergence. *Journal lectronique d'intelligence artificielle*, 2003. submitted.
21. Rocco A. Servedio. Smooth boosting and learning with malicious noise. In *14th Annual Conference on Computational Learning Theory, COLT 2001 and 5th European Conference on Computational Learning Theory, EuroCOLT 2001, Amsterdam, The Netherlands, July 2001, Proceedings*, volume 2111, pages 473–489. Springer, Berlin, 2001.
22. R. Shapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.
23. S.Kwek and C.Nguyen. iboost: Boosting using an instance-based exponential weighting scheme. *hirteenth European Conference on Machine Learning*, pages 245–257, 2002.
24. Salvatore J. Stolfo, Wei Fan, Wenke Lee, Andreas Prodromidis, , and Philip K. Chan. Cost-based modeling and evaluation for data mining with application to fraud and intrusion detection, 1999.
25. F. Torre. Globoost: Boosting de moindres gnraliss. Technical report, GRAppA - Universit Charles de Gaulle - Lille 3, September 2004.
26. D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

Ordinal Classification with Decision Rules

Krzysztof Dembczyński¹, Wojciech Kotłowski¹, and Roman Słowiński^{1,2}

¹ Institute of Computing Science, Poznań University of Technology,
60-965 Poznań, Poland

{kdembczynski, wkotlowski, rslowinski}@cs.put.poznan.pl

² Institute for Systems Research, Polish Academy of Sciences, 01-447 Warsaw, Poland

Abstract. We consider the problem of ordinal classification, in which a value set of the decision attribute (output, dependent variable) is finite and ordered. This problem shares some characteristics of multi-class classification and regression, however, in contrast to the former, the order between class labels cannot be neglected, and, in the contrast to the latter, the scale of the decision attribute is not cardinal. In the paper, following the theoretical framework for ordinal classification, we introduce two algorithms based on gradient descent approach for learning ensemble of base classifiers being decision rules. The learning is performed by greedy minimization of so-called threshold loss, using a forward stage-wise additive modeling. Experimental results are given that demonstrate the usefulness of the approach.

1 Introduction

In the prediction problem, the aim is to predict the unknown value of an attribute y (called *decision attribute*, *output* or *dependent variable*) of an object using known joint values of other attributes (called *condition attributes*, *predictors*, or *independent variables*) $\mathbf{x} = (x_1, x_2, \dots, x_n)$. In the *ordinal classification*, it is assumed that $y = \{r_1, \dots, r_K\}$, with r_k , $k \in \mathcal{K} = \{1, \dots, K\}$, being K distinct and ordered class labels $r_K \succ r_{K-1} \succ \dots \succ r_1$, where \succ denotes the ordering relation between labels. Let us assume in the following, without loss of generality, that $r_k = k$. This problem shares some characteristics of multi-class classification and regression. A value set of y is finite, but in contrast to the multi-class classification, the order between class labels can not be neglected. The values of y are ordered, but in contrast to regression, the scale of y is not cardinal. Such a setting of the prediction problem is very common in real applications. For example, in recommender systems, users are often asked to evaluate items on five value scale (see Netflix Prize problem [16]). Another example is the problem of email classification to ordered groups, like: “very important”, “important”, “normal”, and “later”.

The problem of ordinal classification is often solved by multi-class classification or regression methods. In recent years, however, some new approaches tailored for ordinal classification were introduced [13, 6, 7, 18, 17, 3, 14, 15]. In this paper, we take first a closer look at the nature of ordinal classification. Later

on, we introduce two novel algorithms based on gradient descent approach for learning ensemble of base classifiers. The learning is performed by greedy minimization of so-called threshold loss [17] using a forward stagewise additive modeling [12]. As a base classifier, we have chosen single decision rule which is a logical expression having the form: *if [conditions], then [decision]*. This choice is motivated by simplicity and ease in interpretation of decision rule models. Recently, one can observe a growing interest in decision rule models for classification purposes (e.g. such algorithms like SLIPPER [5], LRI [19], RuleFit [11], ensemble of decision rules [1, 2]).

Finally, we report experimental results that demonstrate the usefulness of the proposed approach for ordinal classification. In particular our approach is competitive to traditional regression and multi-class classification methods. It also shows some advantages over existing ordinal classification methods.

2 Statistical Framework for Ordinal Classification

Similarly to classification and regression, the task is to find a function $F(\mathbf{x})$ that predicts accurately an ordered label of y . The optimal prediction function is given by:

$$F^*(\mathbf{x}) = \arg \min_{F(\mathbf{x})} E_{y\mathbf{x}} L(y, F(\mathbf{x})) \quad (1)$$

where the expected value $E_{y\mathbf{x}}$ is over joint distribution of all variables $P(y, \mathbf{x})$ for the data to be predicted. $L(y, F(\mathbf{x}))$ is a loss or cost for predicting $F(\mathbf{x})$ when the actual value is y . $E_{y\mathbf{x}} L(y, F(\mathbf{x}))$ is called *prediction risk* or *expected loss*. Since $P(y, \mathbf{x})$ is generally unknown, the learning procedure uses only a set of training examples $\{y_i, \mathbf{x}_i\}_1^N$ to construct $F(\mathbf{x})$ to be the best possible approximation of $F^*(\mathbf{x})$. Usually, it is performed by minimization of *empirical risk* $R_e = \frac{1}{N} \sum_{i=1}^N L(y_i, F(\mathbf{x}_i))$.

Let us remind that the typical loss function in binary classification (for which $y \in \{-1, 1\}$) is 0-1 loss:

$$L_{0-1}(y, F(\mathbf{x})) = \begin{cases} 0 & \text{if } y = F(\mathbf{x}), \\ 1 & \text{if } y \neq F(\mathbf{x}), \end{cases} \quad (2)$$

and in regression (for which $y \in \mathbb{R}$), it is squared-error loss:

$$L_{se}(y, F(\mathbf{x})) = (y - F(\mathbf{x}))^2. \quad (3)$$

One of the important properties of the loss function is a form of prediction function minimizing the expected risk $F^*(\mathbf{x})$, so called *population minimizer*. In other words, it is an answer to a question: what does a minimization of expected loss estimate on a population level? Let us remind that the population minimizers for 0-1 loss and squared-error loss are, respectively:

$$F^*(\mathbf{x}) = \text{sgn}(\Pr(y = 1|\mathbf{x}) - 0.5), \quad F^*(\mathbf{x}) = E(y|\mathbf{x}).$$

Table 1. Commonly used loss functions and their population minimizers

Loss function	Notation	$L(y, F(\mathbf{x}))$	$F^*(\mathbf{x})$
Binary classification, $y \in \{-1, 1\}$:			
Exponential loss	L_{exp}	$\exp(-y \cdot F(\mathbf{x}))$	$\frac{1}{2} \log \frac{\Pr(y=1 \mathbf{x})}{\Pr(y=-1 \mathbf{x})}$
Deviance	L_{dev}	$\log(1 + \exp(-2 \cdot y \cdot F(\mathbf{x})))$	$\frac{1}{2} \log \frac{\Pr(y=1 \mathbf{x})}{\Pr(y=-1 \mathbf{x})}$
Regression, $y \in \mathbb{R}$:			
Least absolute deviance	L_{lad}	$ y - F(\mathbf{x}) $	$\text{median}(y \mathbf{x})$

Apart from 0-1 and squared error loss, some other important loss functions are considered. Their definitions and population minimizers are given in Table 1.

In ordinal classification, one minimizes prediction risk based on the $K \times K$ loss matrix:

$$L_{K \times K}(y, F(\mathbf{x})) = [l_{ij}]_{K \times K} \quad (4)$$

where $y, F(\mathbf{x}) \in \mathcal{K}$, and $i = y, j = F(\mathbf{x})$. The only constraints that (4) must satisfy in ordinal classification problem are the following, $l_{ii} = 0, \forall i, l_{ik} \geq l_{ij}, \forall k > j > i$, and $l_{ik} \leq l_{ij}, \forall k < j < i$. Observe that for

$$l_{ij} = 1, \quad \text{if } i \neq j, \quad (5)$$

loss matrix (4) boils down to the 0-1 loss for ordinary multi-class classification problem. One can also simulate typical regression loss functions, such as least absolute deviance and squared-error, by taking:

$$l_{ij} = |i - j|, \quad (6)$$

$$l_{ij} = (i - j)^2, \quad (7)$$

respectively. It is interesting to see, what are the population minimizers of the loss matrices (5)-(7). Let us observe that we deal here with the multinomial distribution of y , and let us denote $\Pr(y = k|\mathbf{x})$ by $p_k(\mathbf{x})$. The population minimizer is then defined as:

$$F^*(\mathbf{x}) = \arg \min_{F(\mathbf{x})} \sum_{k=1}^K p_k(\mathbf{x}) \cdot L_{K \times K}(k, F(\mathbf{x})). \quad (8)$$

For loss matrices (5)-(7) we obtain, respectively:

$$F^*(\mathbf{x}) = \arg \max_{k \in \mathcal{K}} p_k(\mathbf{x}), \quad (9)$$

$$F^*(\mathbf{x}) = \text{median}_{p_k(\mathbf{x})}(y) = \text{median}(y|\mathbf{x}), \quad (10)$$

$$F^*(\mathbf{x}) = \sum_{k=1}^K k \cdot p_k(\mathbf{x}) = E(y|\mathbf{x}). \quad (11)$$

In (11) it is assumed that the range of $F(\mathbf{x})$ is a set of real values.

The interesting corollary from the above is that in order to solve ordinal classification problem one can use any multi-class classification method that estimates $p_k(\mathbf{x})$, $k \in \mathcal{K}$. This can be, for example, logistic regression or gradient boosting machine [9]. A final decision is then computed according to (8) with respect to chosen loss matrix. For (5)-(7) this can be done by computing mode, median or average over y with respect to estimated $p_k(\mathbf{x})$, respectively. For loss matrix entries defined by (7) one can use any regression method that aims at estimating $E(y|\mathbf{x})$. We refer to such an approach as *simple ordinal classifier*.

Let us notice that multi-class classification problem is often solved as K (one class against $K - 1$ classes) or $K \times (K - 1)$ (one class against one class) binary problems. However, taking into account the order on y , we can solve the ordinal classification by solving $K - 1$ binary classification problems. In the k -th ($k = 1, \dots, K - 1$) binary problem, objects for which $y \leq k$ are labeled as $y' = -1$ and objects for which $y > k$ are labeled as $y' = 1$. Such an approach has been used in [6].

The ordinal classification problem can also be formulated from a value function perspective. Let us assume that there exists a latent value function that maps objects to scalar values. The ordered classes correspond to contiguous intervals on a range of this function. In order to define K intervals, one needs $K + 1$ thresholds: $\theta_0 = -\infty < \theta_1 < \dots < \theta_{K-1} < \theta_K = \infty$. Thus k -th class is determined by $(\theta_{k-1}, \theta_k]$. The aim is to find a function $F(\mathbf{x})$ that is possibly close to any monotone transformation of the latent value function and to estimate thresholds $\{\theta_k\}_1^{K-1}$. Then, instead of the loss matrix (4) one can use a continuous and convex loss function, so-called immediate-threshold or all-threshold loss [17] defined respectively as:

$$L^{imm}(y, F(\mathbf{x})) = L(1, F(\mathbf{x}) - \theta_{y-1}) + L(-1, F(\mathbf{x}) - \theta_y), \quad (12)$$

$$L^{all}(y, F(\mathbf{x})) = \sum_{k=1}^{y-1} L(1, F(\mathbf{x}) - \theta_k) + \sum_{k=y}^{K-1} L(-1, F(\mathbf{x}) - \theta_k). \quad (13)$$

In the above, $L(y, f)$ is one of the standard binary classification loss functions. There is, however, a problem with interpretation what does minimization of expected threshold loss estimate. Only in the case when 0-1 loss is chosen as the basis of (12) and (13), the population minimizer has a nice interpretable form. For (12), we have:

$$F^*(\mathbf{x}) = \arg \min_{F(\mathbf{x})} \sum_{k=1}^K p_k(\mathbf{x}) \cdot L_{0-1}^{imm}(k, F(\mathbf{x})) = \arg \max_{k \in \mathcal{K}} p_k(\mathbf{x}), \quad (14)$$

and for (13), we have:

$$F^*(\mathbf{x}) = \arg \min_{F(\mathbf{x})} \sum_{k=1}^K p_k(\mathbf{x}) \cdot L_{0-1}^{all}(k, F(\mathbf{x})) = \text{median}(y|\mathbf{x}). \quad (15)$$

An interesting theoretical result is obtained in [15], where (12) and (13) are used in derivation of the upper bound of generalization error for any loss matrix (4).

Threshold loss functions were already considered in building classifiers. In [17] the classifier was learned by conjugate gradient descent. Among different base loss functions, also deviance was used. In [18, 3, 15], a generalization of SVM (support vector machines) was derived. The algorithm based on AdaBoost [8] was proposed in [15]. In the next section, we present two algorithms based on forward stagewise additive modeling. The first one is an alternative boosting formulation for threshold loss functions. The second one is an extension of the gradient boosting machine [9].

Let us remark at the end of our theoretical considerations that (13) can also be formulated as a specific case of so-called rank loss [13, 7, 4]:

$$L_{rank}(y_1, y_2, F(\mathbf{x}_1), F(\mathbf{x}_2)) = L(\text{sgn}(y_1 - y_2), F(\mathbf{x}_1) - F(\mathbf{x}_2)). \quad (16)$$

This loss function requires that all objects are compared pairwise. Assuming that thresholds $\{\theta_k\}_1^{K-1}$ are values of $F(\mathbf{x})$ for some virtual objects/profiles and all other objects are compared only with these virtual profiles, one obtains (13). Rank loss was used in [13] to introduce a generalization of SVM for ordinal classification problems, and in [7], an extension of AdaBoost for ranking problems was presented. The drawback of this approach is the complexity of empirical risk minimization defined by rank loss that grows quadratically with the problem size (number of training examples). For this reason we do not use this approach in our study.

3 Ensemble of Decision Rules for Ordinal Classification

The introduced algorithms generating an ensemble of ordinal decision rules are based on forward stagewise additive modeling [12]. The decision rule being the base classifier is a logical expression having the form: *if [conditions], then [decision]*. If an object satisfies conditions of the rule, then the suggested decision is taken. Otherwise no action is performed. By conditions we mean a conjunction of expressions of the form $x_j \in S$, where S is a value subset of j -th attribute, $j \in \{1, \dots, n\}$. Denoting set of conditions by Φ and decision by α , the decision rule can be equivalently defined as:

$$r(\mathbf{x}, \mathbf{c}) = \begin{cases} \alpha & \text{if } \mathbf{x} \in \text{cov}(\Phi), \\ 0 & \text{if } \mathbf{x} \notin \text{cov}(\Phi), \end{cases} \quad (17)$$

where $\mathbf{c} = (\Phi, \alpha)$ is a set of parameters. Objects that satisfy Φ are denoted by $\text{cov}(\Phi)$ and referred to as cover of conditions Φ .

The general scheme of the algorithm is presented as Algorithm 1. In this procedure, $F_m(\mathbf{x})$ is a real function being a linear combination of decision rules $r(\mathbf{x}, \mathbf{c})$, $\{\theta_k\}_1^{K-1}$ are thresholds and M is a number of rules to be generated. $L^{all}(y_i, F(\mathbf{x}))$ is an all-threshold loss function. The algorithm starts with $F_0(\mathbf{x}) = 0$ and $\{\theta_k\}_1^{K-1} = 0$. In each iteration of the algorithm, function $F_{m-1}(\mathbf{x})$ is

Algorithm 1: Ensemble of ordinal decision rules

input : set of training examples $\{y_i, \mathbf{x}_i\}_1^N$,
 M – number of decision rules to be generated.
output: ensemble of decision rules $\{r_m(\mathbf{x})\}_1^M$,
 thresholds $\{\theta_k\}_1^{K-1}$.
 $F_0(\mathbf{x}) := 0$; $\{\theta_{k0}\}_1^{K-1} := 0$;
for $m = 1$ *to* M **do**
 $(\mathbf{c}, \{\theta_k\}_1^{K-1}) := \arg \min_{(\mathbf{c}, \{\theta_k\}_1^{K-1})} \sum_{i=1}^N L^{all}(y_i, F_{m-1}(\mathbf{x}_i) + r(\mathbf{x}_i, \mathbf{c}))$;
 $r_m(\mathbf{x}, \mathbf{c}) := r(\mathbf{x}, \mathbf{c})$;
 $\{\theta_{km}\}_1^{K-1} := \{\theta_k\}_1^{K-1}$;
 $F_m(\mathbf{x}) := F_{m-1}(\mathbf{x}) + r_m(\mathbf{x}, \mathbf{c})$;
end
 $ensemble = \{r_m(\mathbf{x}, \mathbf{c})\}_1^M$; $thresholds = \{\theta_{kM}\}_1^{K-1}$;

augmented by one additional rule $r_m(\mathbf{x}, \mathbf{c})$. A single rule is built by sequential addition of new conditions to Φ and computation of α . This is done in view of minimizing

$$\begin{aligned}
 L_m &= \sum_{i=1}^N L^{all}(y_i, F_{m-1}(\mathbf{x}_i) + r(\mathbf{x}_i, \mathbf{c})) = \\
 &= \sum_{\mathbf{x}_i \in cov(\Phi)} \left(\sum_{k=1}^{y_i-1} L(1, F_{m-1}(\mathbf{x}_i) + \alpha - \theta_k) + \sum_{k=y_i}^{K-1} L(-1, F(\mathbf{x}_i)_{m-1} + \alpha - \theta_k) \right) \\
 &\quad + \sum_{\mathbf{x}_i \notin cov(\Phi)} \left(\sum_{k=1}^{y_i-1} L(1, F_{m-1}(\mathbf{x}_i) - \theta_k) + \sum_{k=y_i}^{K-1} L(-1, F(\mathbf{x}_i)_{m-1} - \theta_k) \right) \quad (18)
 \end{aligned}$$

with respect to Φ , α and $\{\theta_k\}_1^{K-1}$. A single rule is built until L_m cannot be decreased.

Ordinal classification decision is computed according to:

$$F(\mathbf{x}) = \sum_{k=1}^K k \cdot I \left(\sum_{m=1}^M r_m(\mathbf{x}, \mathbf{c}) \in [\theta_{k-1}, \theta_k) \right), \quad (19)$$

where $I(a)$ is an indicator function, i.e. if a is true then $I(a) = 1$, otherwise $I(a) = 0$. Some other schemes of classification are also possible. For example, in experiments we have used a procedure that assigns intermediate values between class labels in order to minimize squared error.

In the following two subsections, we give details of two introduced algorithms.

3.1 Ordinal Decision Rules based on Exponential Boosting (ORDER-E)

The algorithm described in this subsection can be treated as generalization of AdaBoost [8] with decision rules as base classifiers. In each iteration of the

algorithm, a strictly convex function (18) defined using the exponential loss L_{exp} is minimized with respect to parameters Φ , α and $\{\theta_k\}_1^{K-1}$. In iteration m , it is easy to compute the following auxiliary values that depend only on $F_{m-1}(\mathbf{x})$ and Φ :

$$\begin{aligned} A_{km} &= \sum_{\mathbf{x}_i \in cov(\Phi)} I(y_i > k) e^{-F_{m-1}(\mathbf{x}_i)} & B_{km} &= \sum_{\mathbf{x}_i \in cov(\Phi)} I(y_i \leq k) e^{F_{m-1}(\mathbf{x}_i)} \\ C_{km} &= \sum_{\mathbf{x}_i \notin cov(\Phi)} I(y_i > k) e^{-F_{m-1}(\mathbf{x}_i)} & D_{km} &= \sum_{\mathbf{x}_i \notin cov(\Phi)} I(y_i \leq k) e^{F_{m-1}(\mathbf{x}_i)} \end{aligned}$$

These values are then used in computation of the parameters. The optimal values for thresholds $\{\theta_k\}_1^{K-1}$ are obtained by setting the derivative to zero:

$$\frac{\partial L_m}{\partial \theta_k} = 0 \Leftrightarrow \theta_k = \frac{1}{2} \log \frac{B_k \cdot \exp(\alpha) + D_k}{A_k \exp(-\alpha) + C_k}, \quad (20)$$

where parameter α is still to be determined. Putting (20) into (18), we obtain the formula for L_m :

$$L_m = 2 \sum_{k=1}^{K-1} \sqrt{B_k \cdot \exp(\alpha) + D_k} (A_k \cdot \exp(-\alpha) + C_k). \quad (21)$$

which now depends only on single parameter α . The optimal value of α can be obtained by solving

$$\frac{\partial L_m}{\partial \alpha} = 0 \Leftrightarrow \sum_{k=1}^{K-1} \frac{B_k \cdot C_k \cdot \exp(\alpha) - A_k \cdot D_k \cdot \exp(-\alpha)}{\sqrt{(B_k \cdot \exp(\alpha) + D_k)(A_k \cdot \exp(-\alpha) + C_k)}} = 0 \quad (22)$$

There is, however, no simple and fast exact solution to (22). That is why we approximate α by a single Newton-Raphson step:

$$\alpha := \alpha_0 - \nu \cdot \frac{\partial L_m}{\partial \alpha} \cdot \left(\frac{\partial^2 L_m}{\partial^2 \alpha} \right)^{-1} \Bigg|_{\alpha=\alpha_0} \quad (23)$$

computed around zero, i.e. $\alpha_0 = 0$. Summarizing, a set of conditions Φ is chosen which minimizes (21) with α given by (23). One can notice the absence of thresholds in the formula for total loss (21). Indeed, thresholds are necessary only for further classification and can be determined once, at the end of induction procedure. However, the total loss (21) is not additive anymore, i.e. it is not the sum of losses of objects due to implicit dependence between objects through the (hidden) thresholds values.

Another boosting scheme for ordinal classification has been proposed in [14]. Similar loss function has been used, although expressed in terms of margins (therefore called “left-right margins” and “all-margins” instead of “immediate-thresholds” and “all-thresholds”). However, in [14] optimization over parameters is performed sequentially. First, a base learner is fitted with $\alpha = 1$. Then, the

optimal value of α is obtained, using thresholds values from previous iterations. Finally, the thresholds are updated. In section 4, we compared this boosting strategy with our methods, showing that such a sequential optimization does not work well with decision rule as a base learner.

3.2 Ordinal Decision Rules based on Gradient Boosting (ORDER-G)

The second algorithm is an extension of the gradient boosting machine [9]. Here, the goal is to minimize the all-threshold loss function (18) defined by deviance loss L_{dev} (thus, denoted as L_{dev}^{all}). Φ is determined by searching for regression rule that fits pseudoresponses \tilde{y}_i being negative gradients:

$$\tilde{y}_i = - \left. \frac{\partial L_{dev}^{all}(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)} \right|_{F(\mathbf{x}_i)=F_{m-1}(\mathbf{x}_i)} \quad (24)$$

with $\{\theta_{km-1}\}_1^{K-1}$ determined in iteration $m - 1$. The regression rule is fit by minimization of the squared-error loss:

$$\sum_{\mathbf{x}_i \in cov(\Phi)} (\tilde{y}_i - F_{m-1}(\mathbf{x}_i) - \tilde{\alpha})^2 + \sum_{\mathbf{x}_i \notin cov(\Phi)} (\tilde{y}_i - F_{m-1}(\mathbf{x}_i))^2. \quad (25)$$

The minimum of (25) is reached for

$$\tilde{\alpha} = \frac{\sum_{\mathbf{x}_i \in cov(\Phi)} (\tilde{y}_i - F_{m-1}(\mathbf{x}_i))}{\sum_{\mathbf{x}_i \in cov(\Phi)} 1}. \quad (26)$$

The optimal value for α is obtained by setting $\frac{\partial L_m}{\partial \alpha} = 0$ with Φ already determined in previous step. However, since this equation has no closed-form solution, the value of α is then approximated by a single Newton-Raphson step, as in (23). Finally, $\{\theta_{km}\}_1^{K-1}$ are determined by $\frac{\partial L_m}{\partial \theta_{km}} = 0$. Once again, since there is no closed-form solution, θ_{km} is approximated by a single Newton-Raphson step, $\theta_{km} = \theta_{km-1} - \frac{\partial L_m}{\partial \theta_{km}} \cdot (\frac{\partial^2 L_m}{\partial^2 \theta_{km}})^{-1} \Big|_{\theta_{km}=\theta_{km-1}}$, with Φ and α previously determined.

Notice that the scheme presented here is valid not only for L_{dev} , but for any other convex, differentiable loss function used as a base loss function in (18).

4 Experimental Results

We performed two experiments. Our aim was to compare simple ordinal classifiers, ordinal decision rules and approaches introduced in [3, 14]. We also wanted to check, how the introduced approaches works on Netflix Prize dataset [16]. As a comparison criteria we chose zero-one error (ZOE), mean absolute error (MEA) and root mean squared error (RMSE). The former two were used in referred papers. RMSE was chosen because of Netflix Prize rules.

The simple ordinal classifiers were based on logistic regression, LogitBoost [10, 9] with decision stumps, linear regression and additive regression [9]. Implementations of these methods were taken from Weka package [20]. In the case of

logistic regression and LogitBoost, decisions were computed according to the analysis given in section 2. In order to minimize, ZOE, MAE and RMSE a final decision was computed as a mode, median or average over the distribution given by these methods, respectively. We used three ordinal rule ensembles. The first one is based on ORBoost-All scheme introduced in [14]. The other two are ORDER-E and ORDER-G introduced in this paper. In this case, a final decision was computed according to (19) in order to minimize ZOE and MAE. For minimization of RMSE, we have assumed that the ensemble constructs $F_M(\mathbf{x})$ which is monotone transformation of a value function defined on an interval $[1, 5] \subseteq \mathbb{R}$. In classification procedure, values of $F_M(\mathbf{x})$ are mapped to $[1, 5] \subseteq \mathbb{R}$ by:

$$F(\mathbf{x}) = \sum_{k=1}^K \left(k + \frac{F_M(\mathbf{x}) - (\theta_k + \theta_{k-1})/2}{\theta_k - \theta_{k-1}} \right) \cdot I(F_M(\mathbf{x}) \in [\theta_{k-1}, \theta_k]),$$

where $\theta_0 = \theta_1 - 2 \cdot (\theta_2 - \theta_1)$ and $\theta_K = \theta_{K-1} + 2 \cdot (\theta_{K-1} - \theta_{K-2})$. These methods were compared with SVM with explicit constraints and SVM with implicit constraints introduced in [3] and with ORBoost-LR and ORBoost-All with perceptron and sigmoid base classifiers introduced in [14].

In the first experiment we used the same datasets and settings as in [3, 14] in order to compare the algorithms. These datasets were discretized by equal-frequency bins from some metric regression datasets. We used the same $K = 10$, the same “training/test” partition ratio, and also averaged the results over 20 trials. We report in Table 2 the mean and standard errors of all test results for zero-one error (ZOE) and mean absolute error (MEA) as it was done in the referred papers. In the last column of the table we put the best result found in [3, 14] for a given dataset. The optimal parameters for simple ordinal classifiers and ordinal rule ensembles were obtained in 5 trials without changing all other settings.

Second experiment was performed on Netflix Prize dataset [16]. We chose 10 first movies from the list of Netflix movies, which have been evaluated by at least 10 000 and at most 30 000 users. Three types of error (ZOE, MEA and RMSE) were calculated. We compared here only simple ordinal classifiers with ordinal rule ensembles. Classifiers were learned on Netflix-training dataset and tested on Netflix-probe dataset (all evaluations from probe dataset were removed from training dataset). Ratings on 100 movies, selected in the same way for each movie, were used as condition attributes. For each method, we tuned its parameters to optimize its performance, using 10% of training set as a validation set for the parameters; to avoid favouring methods with more parameters, for each algorithm we performed the same number of tuning trials. The results are shown in Table 3.

The results from both experiments indicate that ensembles of ordinal decision rules are competitive to other methods used in the experiment:

- From the first experiment, one can conclude that ORBoost strategy does not work well with decision rule as a base learner, and that simple ordinal classifiers and ordinal decision rules perform comparably to approaches introduced in [3, 14].

Table 2. Experimental results on datasets used in [3, 14]. The same data preprocessing is used that enables comparison of the results. In the last column, the best results obtained by ¹⁾SVM with explicit constraints [3], ²⁾SVM with implicit constraints [3], ³⁾ORBoost-LR [14], and ⁴⁾ORBoost-All [14] are reported. Two types of error are considered (zero-one and mean-absolute). Best results are marked in bold among all compared methods and among methods introduced in this paper.

Zero-one error (ZOE)						
Dataset	Logistic Regression	LogitBoost with DS	ORBoost-All with Rules	ORDER-E	ORDER-G	Best result from [3, 14]
Pyrim.	0.754±0.017	0.773±0.018	0.852±0.011	0.754±0.019	0.779±0.018	0.719±0.066 ²
CPU	0.648±0.009	0.587±0.012	0.722±0.011	0.594±0.014	0.562±0.009	0.605±0.010 ⁴
Boston	0.615±0.007	0.581±0.007	0.653±0.008	0.560±0.006	0.581±0.007	0.549±0.007 ³
Abal.	0.678±0.002	0.694±0.002	0.761±0.003	0.710±0.002	0.712±0.002	0.716±0.002 ³
Bank	0.679±0.001	0.693±0.001	0.852±0.002	0.754±0.001	0.759±0.001	0.744±0.005 ¹
Comp.	0.489±0.001	0.494±0.001	0.593±0.002	0.476±0.002	0.479±0.001	0.462±0.001 ¹
Calif.	0.665±0.001	0.606±0.001	0.773±0.002	0.631±0.001	0.609±0.001	0.605±0.001 ³
Census	0.707±0.001	0.665±0.001	0.793±0.001	0.691±0.001	0.687±0.001	0.694±0.001 ³
Mean absolute error (MAE)						
Dataset	Logistic Regression	LogitBoost with DS	ORBoost-All with Rules	ORDER-E	ORDER-G	Best result from [3, 14]
Pyrim.	1.665±0.056	1.754±0.050	1.858±0.074	1.306±0.041	1.356±0.063	1.294±0.046 ²
CPU	0.934±0.021	0.905±0.025	1.164±0.026	0.878±0.027	0.843±0.022	0.889±0.019 ¹
Boston	0.903±0.013	0.908±0.017	1.068±0.017	0.813±0.010	0.828±0.014	0.747±0.011 ²
Abal.	1.202±0.003	1.272±0.003	1.520±0.008	1.257±0.002	1.281±0.004	1.361±0.003 ²
Bank	1.445±0.003	1.568±0.003	2.183±0.005	1.605±0.005	1.611±0.004	1.393±0.002 ²
Comp.	0.628±0.002	0.619±0.002	0.930±0.005	0.583±0.002	0.588±0.002	0.596±0.002 ²
Calif.	1.130±0.004	0.957±0.001	1.646±0.007	0.955±0.003	0.897±0.002	0.942±0.002 ⁴
Census	1.432±0.003	1.172±0.002	1.669±0.006	1.152±0.002	1.166±0.002	1.198±0.002 ⁴

- The second experiment shows that especially ORDER-E outperforms other methods in RMSE for most of the movies and in MAE for half of the movies. However, this method was the slowest between all tested algorithms. ORDER-G is much more faster than ORDER-E, but it obtained moderate results.
- In both experiments logistic regression and LogitBoost perform well. It is clear that these algorithms achieved the best results with respect to ZOE. The reason is that they can be tailored to multi-classification problem with zero-one loss, while ordinal decision rules can not.
- It is worth noticing, that regression algorithms resulted in poor accuracy in many cases.
- We have observed during the experiment that ORDER-E and ORDER-G are sensitive to parameters setting. We plan to work on some simple method for parameters selection.

5 Conclusions

From the theoretical analysis, it follows that different formulations are possible for the ordinal classification problem. In our opinion, there is still a lot to do in order to establish a theoretic framework for ordinal classification. In this

Table 3. Experimental results on 10 movies from Netflix Prize data set. Three types of error are considered (zero-one, mean-absolute and root mean squared). For each movie, best results are marked in bold.

Zero-one error (ZOE)						
Movie #	Linear Regression	Additive Regression	Logistic Regression	LogitBoost with DS	ORDER-E	ORDER-G
8	0.761	0.753	0.753	0.714	0.740	0.752
18	0.547	0.540	0.517	0.493	0.557	0.577
58	0.519	0.496	0.490	0.487	0.513	0.496
77	0.596	0.602	0.583	0.580	0.599	0.605
83	0.486	0.486	0.483	0.398	0.462	0.450
97	0.607	0.607	0.591	0.389	0.436	0.544
108	0.610	0.602	0.599	0.593	0.613	0.596
111	0.563	0.561	0.567	0.555	0.572	0.563
118	0.594	0.596	0.532	0.524	0.511	0.551
148	0.602	0.610	0.593	0.536	0.522	0.573
Mean absolute error (MAE)						
Movie #	Linear Regression	Additive Regression	Logistic Regression	LogitBoost with DS	ORDER-E	ORDER-G
8	1.133	1.135	1.115	1.087	1.013	1.018
18	0.645	0.651	0.583	0.587	0.603	0.613
58	0.679	0.663	0.566	0.543	0.558	0.560
77	0.831	0.839	0.803	0.781	0.737	0.755
83	0.608	0.614	0.519	0.448	0.500	0.502
97	0.754	0.752	0.701	0.530	0.537	0.654
108	0.777	0.776	0.739	0.739	0.768	0.739
111	0.749	0.766	0.720	0.715	0.693	0.705
118	0.720	0.734	0.626	0.630	0.596	0.658
148	0.747	0.735	0.688	0.626	0.604	0.659
Root mean squared error (RMSE)						
Movie #	Linear Regression	Additive Regression	Logistic Regression	LogitBoost with DS	ORDER-E	ORDER-G
8	1.332	1.328	1.317	1.314	1.268	1.299
18	0.828	0.836	0.809	0.856	0.832	0.826
58	0.852	0.847	0.839	0.805	0.808	0.817
77	1.067	1.056	1.056	1.015	0.999	1.043
83	0.775	0.772	0.737	0.740	0.729	0.735
97	0.968	0.970	0.874	0.865	0.835	0.857
108	0.984	0.993	0.969	0.979	0.970	0.989
111	0.985	0.992	0.970	0.971	0.967	0.986
118	0.895	0.928	0.862	0.860	0.836	0.873
148	0.924	0.910	0.900	0.863	0.838	0.893

paper, we introduced a decision rule induction algorithm based on forward stage-wise additive modeling that utilizes the notion of threshold loss function. The experiment indicates that ordinal decision rules are quite promising. They are competitive to traditional regression and multi-class classification methods and also show some advantages over existing ordinal classification methods. Let us remark that the algorithm can also be used for other base classifiers like decision trees instead of decision rules. In this paper, we remained with rules because of their simplicity in interpretation. It is also interesting that such a simple classifier works so well as a part of the ensemble.

References

1. Błaszczyński, J., Dembczyński, K., Kotłowski, W., Słowiński, R., and Szeląg, M.: Ensembles of Decision Rules. *Foundations of Computing and Decision Sciences*, **31** (2006) 21–232
2. Błaszczyński, J., Dembczyński, K., Kotłowski, W., Słowiński, R., and Szeląg, M.: Ensembles of Decision Rules for Solving Binary Classification Problems in the Presence of Missing Values. *Lecture Notes in Artificial Intelligence*, **4259** (2006) 224–234
3. Chu, W., Keerthi, S.S.: New approaches to support vector ordinal regression. In: Proceedings of International Conference on Machine Learning (2005) 321–328
4. Cléménçon, S., Lugosi, G., and Vayatis, N.: Ranking and empirical minimization of U-statistics. (to appear)
5. Cohen, W., Singer, Y.: A simple, fast, and effective rule learner. In *Proc. of 16th National Conference on Artificial Intelligence* (1999) 335–342
6. Frank, E., Hall, M.: A simple approach to ordinal classification. *Lecture Notes in Computer Science*, **2167** (2001) 145–157
7. Freund, Y., Iyer, R., Schapire, R., and Singer, Y.: An efficient boosting algorithm for combining preferences. *J. of Machine Learning Research*, **4** (2003) 933–969.
8. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. *J. of Computer and System Sciences*, **55** **1** (1997) 119–139
9. Friedman, J.: Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, **29** **5** (2001) 1189–1232
10. Friedman, J., Hastie, T., and Tibshirani, R.: Additive logistic regression: a statistical view of boosting. *Annals of Statistics* (2000) 337–407
11. Friedman, J., Popescu, B.: Predictive learning via rule ensembles. Research report, Dept. of Statistics, Stanford University (2005)
12. Hastie, T., Tibshirani, R., and Friedman, J. H.: *Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer (2003)
13. Herbrich, R., Graepel, T., and Obermayer, K.: Regression models for ordinal data: A machine learning approach. Technical report TR-99/03, TU Berlin (1999)
14. Lin, H.-T., Li, L.: Large-margin thresholded ensembles for ordinal regression: Theory and practice. *Lecture Notes in Artificial Intelligence* **4264** (2006) 319–333
15. Lin, H.-T., Li, L.: Ordinal regression by extended binary classifications. *Advances in Neural Information Processing Systems* **19** (2007) 865–872
16. Netflix prize, <http://www.netflixprize.com>.
17. Rennie, J., Srebro, N.: Loss functions for preference levels: Regression with discrete ordered labels. In *Proc. of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling* (2005)
18. Shashua, A., Levin, A.: Ranking with large margin principle: Two approaches. *Advances in Neural Information Processing Systems*, **15** (2003)
19. Weiss, S., Indurkha, N.: Lightweight rule induction. In *Proc. of 17th International Conference on Machine Learning* (2000) 1135–1142
20. Witten, I., Frank, E.: *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann (2005)

Quality of Musical Instrument Sound Identification for Various Levels of Accompanying Sounds

Alicja Wieczorkowska¹ and Elżbieta Kolczyńska²

¹ Polish-Japanese Institute of Information Technology,
Koszykowa 86, 02-008 Warsaw, Poland
alicja@pjwstk.edu.pl,

² Agricultural University in Lublin
Akademicka 13, 20-950 Lublin, Poland
elzbieta.kolczynska@ar.lublin.pl

Abstract. Research on automatic identification of musical instrument sounds has already been performed through last years, but mainly for monophonic singular sounds. In this paper we work on identification of musical instrument in polyphonic environment, with added accompanying orchestral sounds for the training purposes, and using mixes of 2 instrument sounds for testing. Four instruments of definite pitch has been used. For training purposes, these sounds were mixed with orchestral recordings of various levels, diminished with respect to the original recording level. The experiments have been performed using WEKA classification software.

1 Introduction

Recognition of musical instrument sound from audio files is not a new topic and research in this area has already been performed worldwide [2], [3]. This research was mainly performed on singular monophonic sounds. However, the recognition of instrument, or instruments, in polyphonic recording is much more difficult, especially when no spatial clues are used. In our paper we aim at training classifiers for the purpose of recognition of predominant musical instrument sound, using various sets of training data. We believe that using for training not only clean singular monophonic musical instrument sound samples, but also the sounds with added other accompanying sounds, may improve classification quality. We are interested in checking how various levels of accompanying sounds (distorting the original sound waves) influence correctness of classification of predominant (louder) instrument in mixes containing 2 instrumental sounds.

2 Sound Parameterization

For classification purposes the sound data are parameterized, using various features describing temporal, spectral, and spectral-temporal properties of sounds.

Features implemented in the worldwide research on musical instrument sound recognition so far include parameters based on DFT, wavelet analysis, MFCC (Mel-Frequency Cepstral Coefficients), MSA (Multidimensional Analysis Scaling) trajectories, and so on [1], [3], [5], [6], [8]. Also, MPEG-7 sound descriptors can be applied [4], although these parameters are not dedicated to recognition of particular instruments in recordings. In our research we applied the following 219 parameters, based mainly on MPEG-7 audio descriptors, and also other parameters used for musical instrument sound identification purposes [9], [10]:

- MPEG-7 audio descriptors [4]:
 - *AudioSpectrumSpread* - a RMS value of the deviation of the Log frequency power spectrum with respect to the gravity center in a parameterized frame;
 - *AudioSpectrumFlatness*, $flat_1, \dots, flat_{25}$ - describes the flatness property of the power spectrum within a frequency bin; 25 out of 32 frequency bands were used to calculate these parameters;
 - *AudioSpectrumCentroid* - computed as power weighted average of the frequency bins in the power spectrum of all the frames in a sound segment with a Welch method (time-frequency feature);
 - *AudioSpectrumBasis*: $basis_1, \dots, basis_{165}$ - spectrum basis function is used to reduce the dimensionality by projecting the spectrum from high dimensional space to low dimensional space with compact salient statistical information. Spectral basis parameters are calculated for the spectrum basis functions, where total number of sub-spaces in basis function is 33, and for each sub-space, minimum/maximum/mean/distance/standard deviation are extracted to flat the vector data. Distance is calculated as the summation of dissimilarity (absolute difference of values) of every pair of coordinates in the vector;
 - *HarmonicSpectralCentroid* - the average over the sound segment duration of the instantaneous Harmonic Centroid within a frame. The instantaneous Harmonic Spectral Centroid is computed as the amplitude in linear scale weighted mean of the harmonic peak of the spectrum;
 - *HarmonicSpectralSpread* - the average over the sound segment duration of the instantaneous harmonic spectral spread of frame, i.e. the amplitude weighted standard deviation of the harmonic peaks of the spectrum with respect to the instantaneous harmonic spectral centroid;
 - *HarmonicSpectralVariation* - mean value over the sound segment duration of the instantaneous harmonic spectral variation, i.e. the normalized correlation between the amplitude of the harmonic peaks of two adjacent frames;
 - *HarmonicSpectralDeviation* - the average over the sound segment duration of the instantaneous harmonic spectral deviation in each frame, i.e. the spectral deviation of the log amplitude components from a global spectral envelope;
 - *LogAttackTime*;
 - *TemporalCentroid* - time average over the energy envelope;

- other audio descriptors:
 - *Energy* - average energy of spectrum in the entire sound;
 - MFCC - min, max, mean, distance, and standard deviation of the MFCC vector;
 - *ZeroCrossingDensity*;
 - *RollOff* - measure of spectral shape, used in the speech recognition, where it is used to distinguish between voiced and unvoiced speech. The roll-off is defined as the frequency below which an experimentally chosen percentage of the accumulated magnitudes of the spectrum is concentrated;
 - *Flux* - the difference between the magnitude of the FFT points in a given frame and its successive frame (value multiplied by 10^7 to comply with WEKA requirements)
 - *AverageFundamentalFrequency*;
 - *TristimulusParameters*: $tris_1, \dots, tris_{11}$ - describe the ratio of the amplitude of a harmonic partial to the total harmonic partials.

The calculations of parameters changing in time were performed using 120 ms analyzing frame with Hamming window and hop size 40 ms. Data from the left channel only were taken for parameterization.

3 Experiments

The goal of our research was to check how modification (i.e. sound mixing) of the initial audio data, representing musical instrument sounds, influences the quality of classifiers trained to recognize these instruments. The initial data were taken from McGill University CDs, used worldwide in research on music instrument sounds [7]. The sounds were recorded stereo with 44.1 kHz sampling rate, and 16 bit resolution. We have chosen 188 sounds of the following instruments (i.e. representing 4 classes):

1. B-flat clarinet - 37 sound objects,
2. C-trumpet (also trumpet muted, mute Harmon with stem out) - 65 objects,
3. violin vibrato - 42 objects
4. cello vibrato - 43 objects.

The sounds were parameterized as described in the previous section, thus yielding the clean data for further work. Next, the clean data were distorted in such a way that an excerpt from orchestral recording was added. Adagio from Symphony No. 6 in B minor, Op. 74, Pathétique by P. Tchaikovsky was used for this purpose. Four short excerpts from this symphony were diminished to 10%, 20%, 30%, 40% and 50% of original amplitude, and added to the initial sound data, thus yielding 5 versions of distorted data, used for training of classifiers. The disturbing data were changing in time, but since the parameterization was performed applying short analysis window, we did not decide to search for excerpts with stable spectra (i.e. long lasting chords), but the harmonic contents was relatively stable in the chosen excerpts.

For testing purposes, all clean singular sound objects were mixed with the following 4 sounds

1. C4 sound of c-trumpet,
2. A4 sound of clarinet,
3. D5 sound of violin vibrato
4. G3 sound of cello vibrato,

where A4 = 440 Hz (i.e. MIDI notation is used for pitch). The amplitude of these added 4 sounds was diminished to 10% of the original level, to make sure that the recognized instrument is actually the main, dominating sound in the mixed recording.

For classification purposes, we decided to use WEKA software, with the following classifiers: Bayesian Network, decision trees (Tree J48), Logistic Regression Model (LRM), and Locally Weighted Learning (LWL). The training of each classifier was performed three-fold, separately for each level of the accompanying orchestral sound (i.e. for 10%, 20%, 30%, 40%, and 50%):

- on clean singular sound data only (singular instrument sounds)
- on both singular and accompanied sound data (i.e. mixed with orchestral recording)
- on accompanied sound data only

In each case, the testing was performed on the data obtained via mixing of the initial clean data with other instrument sound (diminished to 10% of original amplitude), as described above.

Summary of results for all these experiments is presented in tables 1–4.

The improvement of correctness for each classifier, trained on both clean singular sound and accompanied sound data, in comparison with the training on clean singular sound data only, is presented in Figure 1. Negative values indicate decrease of correctness, when the mixes with accompanied sounds were added to the training set.

As we can see, for LWL classifier the disturbances in the training data always caused decrease of the correctness of the instrument recognition. However, in most other cases (apart from decision trees) we observe improvement of classification correctness, when mixed sound data are added to the training set.

The improvement of correctness for our classifiers, but trained on mixed sound data only, in comparison with the training on clean singular sound data only, is presented in Figure 2.

As we can see, in this case we only have improvement of correctness for low levels of accompanying sounds. Therefore we can conclude that clean singular sound data are rather necessary to train classifiers for instrument recognition purposes.

When starting these experiments, we hoped to observe some dependencies between the added disturbances (i.e. accompanying sounds) to the training sound data, the level of the disturbance, and change of the classification correctness. As we can see, there are no such clear linear dependencies. On the other hand,

Table 1. Results of experiments for Bayesian network

Classifier	Disturbance level	Training on data:	Correctness %
BayesNet	10%	Singular sounds only	71,41%
		Both singular and accompanied sounds	80,98%
		Accompanied sounds only	75,40%
	20%	Singular sounds only	71,41%
		Both singular and accompanied sounds	76,33%
		Accompanied sounds only	68,62%
	30%	Singular sounds only	71,41%
		Both singular and accompanied sounds	77,39%
		Accompanied sounds only	62,23%
	40%	Singular sounds only	71,41%
		Both singular and accompanied sounds	76,73%
		Accompanied sounds only	61,30%
50%	Singular sounds only	71,41%	
	Both singular and accompanied sounds	75,13%	
	Accompanied sounds only	56,38%	

Table 2. Results of experiments for decision trees (Tree J48)

Classifier	Disturbance level	Training on data:	Correctness %
TreeJ48	10%	Singular sounds only	79,65%
		Both singular and accompanied sounds	76,46%
		Accompanied sounds only	74,47%
	20%	Singular sounds only	79,65%
		Both singular and accompanied sounds	79,65%
		Accompanied sounds only	53,46%
	30%	Singular sounds only	79,65%
		Both singular and accompanied sounds	91,62%
		Accompanied sounds only	62,10%
	40%	Singular sounds only	79,65%
		Both singular and accompanied sounds	66,36%
		Accompanied sounds only	53,59%
	50%	Singular sounds only	79,65%
		Both singular and accompanied sounds	71,94%
		Accompanied sounds only	46,94%

Table 3. Results of experiments for Logistic Regression Model

Classifier	Disturbance level	Training on data:	Correctness %
Logistic	10%	Singular sounds only	78,19%
		Both singular and accompanied sounds	85,11%
		Accompanied sounds only	69,55%
	20%	Singular sounds only	78,19%
		Both singular and accompanied sounds	89,36%
		Accompanied sounds only	62,23%
	30%	Singular sounds only	78,19%
		Both singular and accompanied sounds	85,90%
		Accompanied sounds only	56,91%
	40%	Singular sounds only	78,19%
		Both singular and accompanied sounds	84,18%
		Accompanied sounds only	68,22%
50%	Singular sounds only	78,19%	
	Both singular and accompanied sounds	82,98%	
	Accompanied sounds only	50,40%	

Table 4. Results of experiments for Locally Weighted Learning

Classifier	Disturbance level	Training on data:	Correctness %
LWL	10%	Singular sounds only	67,82%
		Both singular and accompanied sounds	67,42%
		Accompanied sounds only	67,15%
	20%	Singular sounds only	67,82%
		Both singular and accompanied sounds	66,36%
		Accompanied sounds only	68,35%
	30%	Singular sounds only	67,82%
		Both singular and accompanied sounds	62,63%
		Accompanied sounds only	60,77%
	40%	Singular sounds only	67,82%
		Both singular and accompanied sounds	55,85%
		Accompanied sounds only	55,85%
	50%	Singular sounds only	67,82%
		Both singular and accompanied sounds	53,86%
		Accompanied sounds only	54,12%

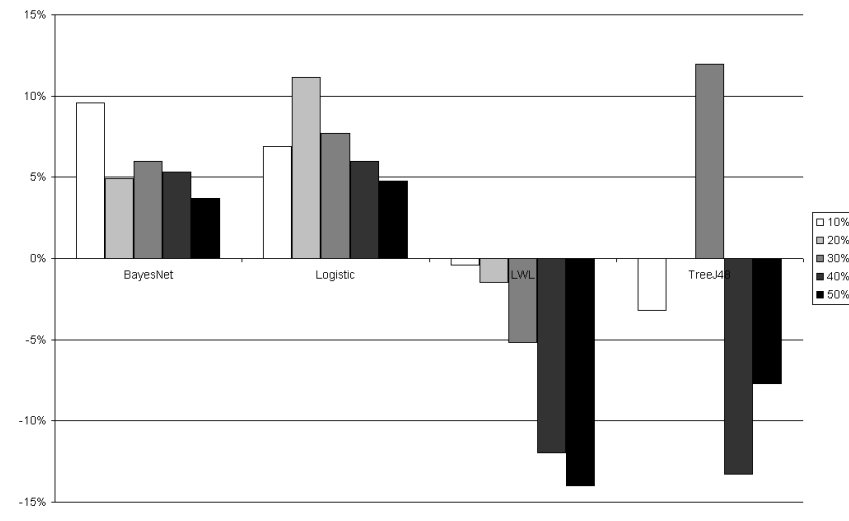


Fig. 1. Change of correctness of musical instrument sound recognition for classifiers built on both clean singular musical instrument sound and accompanied sound data, i.e. with added (mixed) orchestral excerpt of various levels (10%, 20%, 30%, 40%, 50% of original amplitude), and tested on the data distorted through adding other instrument sound to the initial clean sound data. Comparison is made with respect to the results obtained for classifiers trained on clean singular sound data only.

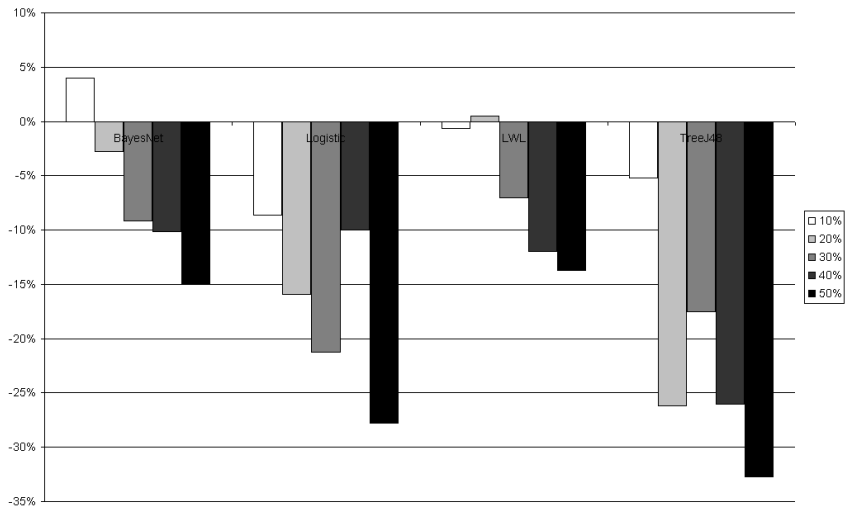


Fig. 2. Change of correctness of musical instrument sound recognition for classifiers built on the sounds of singular instruments with added (mixed) orchestral excerpt of various levels (10%, 20%, 30%, 40%, 50% of original amplitude), and tested on the data with added other instrument sound to the main instrument sound. Comparison is made with respect to the results obtained for classifiers trained on clean singular sound data only.

the type of the disturbance/accompaniment (for example, its harmonic contents, and how it overlaps with the initial sound) may also influence the results. Also, when sound mixes were produced, both sounds in any mix were changing in time, what is natural and unavoidable in case of music. Therefore, in some frames the disturbing, accompanying sounds could be louder than the sound of interest, so mistakes regarding identification of the dominant instrument in the mix also may happen as well.

4 Summary and Conclusions

The experiments described in this paper aimed at observing if (and how) adding disturbance (i.e. accompanying sound added) to the clean (i.e. singular musical instrument sound) data influences correctness of classification of musical instrument sound, dominating in the polyphonic recording. The disturbances added represented various levels of orchestral recordings, added to singular monophonic musical instrument sounds. Tests performed on pairs of instruments sounds shown that in most cases the use of disturbed data, together with initial clean singular sound data, increases the correctness of the classifier, thus increasing its quality. However, no clear linear relationships can be observed.

We plan to continue our experiments, with using various levels of added orchestral sounds for training and for testing the classifiers. Also, since the set of sound features is very important for the correct classification, we plan to check how changes in the feature set influence the quality of classification for distorted in such a way data set.

5 Acknowledgments

This work was supported by the National Science Foundation under grant IIS-0414815, and also by the Research Center of PJIIT, supported by the Polish National Committee for Scientific Research (KBN).

The authors would like to express thanks to Xin Zhang from the University of North Carolina at Charlotte for help with preparing the initial data.

References

1. Brown, J. C.: Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. *Journal of the Acoustical Society of America* **105** (1999) 1933–1941
2. Dziubinski, M., Dalka, P., and Kostek, B.: Estimation of Musical Sound Separation Algorithm Effectiveness Employing Neural Networks. *Journal of Intelligent Information Systems*, **24**:2/3 (2005) 133–157
3. Herrera, P., Amatriain, X., Batlle, E., and Serra X.: Towards instrument segmentation for music content description: a critical review of instrument classification techniques. In *Proc. of International Symposium on Music Information Retrieval ISMIR 2000*, Plymouth, MA

4. ISO/IEC JTC1/SC29/WG11: MPEG-7 Overview. (2004) Available at <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>
5. Kaminskyj, I.: Multi-feature Musical Instrument Sound Classifier w/user determined generalisation performance. Proceedings of the Australasian Computer Music Association Conference ACMC 2002, 53–62
6. Martin, K. D., and Kim, Y. E.: Musical instrument identification: A pattern-recognition approach. 136-th meeting of the Acoustical Society of America, Norfolk, VA (1998)
7. Opolko, F., Wapnick, J.: MUMS - McGill University Master Samples. CD's (1987)
8. Wierzchowska, A.: Towards Musical Data Classification via Wavelet Analysis. In: Foundations of Intelligent Systems, (Eds. Z. W. Ras, S. Ohsuga), Proceedings of ISMIS'00, Charlotte, NC, USA, LNCS/LNAI, No. 1932, Springer-Verlag 2000, 292–300
9. Wierzchowska, A., Kolczyńska, E., Raś, Z., Zhang, X.: Correctness of classification of musical timbre under influence of accompanying sounds. Submitted to The 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology IAT (2007)
10. Zhang, X. and Ras, Z. W.: Analysis of Sound Features for Music Timbre Recognition. International Conference on Multimedia and Ubiquitous Engineering MUE 2007, 26-28 April 2007, Seoul, Korea. Edited by S. Kim, J. H. Park, N. Pissinou, T. Kim, W. C. Fang, D. Slezak, H. Arabnia, D. Howard. IEEE Computer Society, 3–8

Estimating Semantic Distance Between Concepts for Semantic Heterogeneous Information Retrieval

Ahmad El Sayed, Hakim Hacid, Djamel Zighed

University of Lyon 2
ERIC Laboratory- 5, avenue Pierre Mendès-France
69676 Bron cedex - France
{asayed, hhacid, dzighed}@eric.univ-lyon2.fr

Abstract. This paper brings two contributions in relation with the semantic heterogeneous (documents composed of texts and images) information retrieval: (1) A new context-based semantic distance measure for textual data, and (2) an IR system providing a conceptual and an automatic indexing of documents by considering their heterogeneous content using a domain specific ontology. The proposed semantic distance measure is used in order to automatically fuzzify our domain ontology. The two proposals are evaluated and very interesting results were obtained. Using our semantic distance measure, we obtained a correlation ratio of 0.89 with human judgments on a set of words pairs which led our measure to outperform all the other measures. Preliminary combination results obtained on a specialized corpus of web pages are also reported.

1 Introduction

An important lack in the current IRS, is that most of them deal with homogeneous data types. We can find those dealing with text content, others with visual content but rarely with both. Let's take the example of a web page. If we compose a web page by its different components, (where each component represents a data type), we'll find that all parts do not necessarily represent the same piece of information. Even on a single document, each component will have a distinct meaning for the user. So if we treat only text for example, we wash out all the information contained in images, and vice versa. Let's mention that indexing an image by the text surrounding it, as most search engines do, is not the real solution since text does not necessarily represent the image content.

Any IRS contains mainly two important components: a data representation structure which is generally translated by the use of an index for capturing the semantic of the data and accelerating the access to the low level data, and a querying strategy which enables the end user to express his/her query to the system. To be efficient, these two components necessitate another important element which is a semantic similarity measure able to capture the semantic proximity between pieces of information (concepts, words, images, etc...).

In this paper, we present a novel retrieval system that represent documents by their text and image content and thus, by multiple information sources. We bring two main contributions in relation with the semantic heterogeneous (documents composed of texts and images) information retrieval: (1) A new context-based semantic distance¹ measure for textual data (since we are dealing with keyword-based information retrieval), and (2) a IR system providing a conceptual and automatic indexing of documents by considering their heterogeneous content using a domain specific ontology. In order to maximize our system's performances, we automatically fuzzify our knowledge unit using our proposed semantic distance measure.

The rest of this paper is organized as follows: In Section 2, we present and evaluate the new semantic distance measure between concepts. Our heterogeneous information retrieval system is introduced and detailed in Section 3. We conclude and give some future work in Section 4.

2 Semantic Similarity

In text-based applications, beyond managing synonymies and polysemies, one need to measure the degree of semantic similarity between two words/concepts²; let's mention: Information retrieval, question answering, automatic text summarization and translation, etc. Many semantic similarity measures³ have been proposed in the literature. We can distinguish between knowledge-based measures and corpus-based measures.

On the one hand, knowledge-based measures offer reliable results given their hand-crafted 'semantic' logical structure. Taxonomies, like WordNet[8] and Mesh⁴, are widely used for such purpose. These measures can be divided into an edge-based measures [11][6][19], a features-based measures [16], or an Information Content (IC) measures [12][5][7].

On the other hand, corpus-based measures are based on a statistical analysis of large text corpora. They have the advantage of being self-independent; they don't need any external knowledge resources, which can overcome the coverage problem in taxonomies. In this category, we can find co-occurrence based measures [2][15] and context-based measures[1][4].

2.1 A MultiSource Context-Dependent Semantic Distance Between Concepts

Our Context-Dependent Measure A major lack in existing semantic similarity methods is that no one takes into account the context or the considered

¹ We consider distance by its dissimilarity which is the inverse of similarity. Then, greater distance values imply greater difference between compared objects

² In the rest of the paper, 'words' is used when dealing with text corpora and 'concepts' is used when dealing with taxonomies where each concept is represented by a list of words holding a sense.

³ For a more detailed state of art, readers are invited to read our previous paper [13]

⁴ <http://www.nlm.nih.gov/mesh/>

domain. However, two concepts similar in one context may appear completely unrelated in another context. Let's take the example of *heart* and *blood*. In a general context, these two concepts can be judged to be very similar. However, if we put ourselves in a medical context, *heart* and *blood* define two largely separated concepts. They will be even more distant if the context is more specifically related to *cardiology*. Our context-dependent approach⁵ suggest to adapt semantic similarities to the target corpus since it's the entity representing the context or the domain of interest in most text-based applications. This method is inspired by the Information Content (IC) theory of Resnik[12] and by the Jiang[5] measure.

In spite of using a text corpus, the IC proposed measures are unable to capture the target context since they rely uniquely on the probability of encountering a concept in a corpus which is not a sufficiently adaptive measure to reflect its dependency to a given context. A concept very frequent in some few documents and absent in many others cannot be considered to be "well" representative for the corpus. Thus, the number of documents where the concept occurs is another important factor that must be considered. In addition to that, it's most likely that a concept c_1 -with a heterogeneous distribution among documents - is more discriminative than a concept c_2 with a monotone repartition which can reveal less power of discrimination over the target domain (Experimentations made assess our thesis).

Instead of assigning IC values to concepts, we assign weights for taxonomy's concepts according to a Context-Dependency CD measure for a given corpus C . The goal is to obtain a weighted taxonomy, where 'heavier' subtrees are more context representative than 'lighter' subtrees. This will allow us to calculate semantic similarities by considering the actual context. Therefore, lower similarity values will be obtained in 'heavy' subtrees than 'light' subtrees. Thus, in our *heart/blood* example, we tend to give a high similarity for the concept couple in a general context, and a low similarity in a specific context like medicine.

Consequently, we introduce our CD measure which is an adapted version of the standard $tf - idf$. Given a concept c , $CD(c)$ is a function of its total frequency $freq(c)$, the number of documents containing it $d(c)$, and the variance of its frequency distribution $var(c)$ over a corpus C :

$$CD(c) = \frac{\log(1 + freq(c))}{\log(N)} * \frac{\log(1 + d(c))}{\log(D)} * (1 + \log(1 + var(c))) \quad (1)$$

Where N denotes the total number of concepts in C and D is the total number of documents in C . The log likelihood seems adaptive to such purpose since it helps to reduce the big margins between values. This formula ensures that if a concept frequency is 0, its CD will equals 0 too. It ensures also that if c have an instance in C , its CD will never be 0 even if $var(c) = 0$.

Note that the CD of a concept c is the sum of its individual CD value with the CD of all its subconcepts in the taxonomy. The weights propagation from

⁵ The approach is presented in more detail in our previous paper[13]

the bottom to the top of the hierarchy is a natural way to ensure that a parent even with a low individual CD will be considered as highly context-dependent if its children are well represented in the corpus

To compare two concepts using the CD values, we assign a Link Strength (LS) to each 'is-a' link in the taxonomy. Assume that c_1 subsumes c_2 , the LS between c_1 and c_2 is then calculated as follows: $LS(c_1, c_2) = CD(c_1) - CD(c_2)$. Then our Context-Dependent Semantic Sistance (CDSD) is calculated by summing the log likelihood of LS along the shortest path separating the two concepts in the taxonomy:

$$Dist(c_1, c_2) = \sum_{c \in SPath(c_1, c_2)} \log(1 + LS(c, parent(c)))$$

Where $SPath$ denotes the shortest path between c_1 and c_2 .

Combinations with Other Measures First, at the corpus level, the promising rates attained by the corpus-based word similarities techniques and especially for the co-occurrence based ones has pushed us to combine them with our context-dependent measure in order to reach the best possible rates. However, two similar words can appear in the same document, paragraph, sentence, or a fixed-size window. It's true that smaller window size can help identifying relations that hold over short ranges with good precisions, larger window size, yet too coarse-grained, allows to detect large-scale relations that could not been detected with smaller windows. We can say that if small windows improve precision, a large windows improve recall.

We have chosen to combine both techniques in order to view relations at different-scales. At the low scale, we use the PMI measure described above with a window size of 10 words. At the large scale, we calculate the Euclidian distance between words vectors where each word is represented by its tf.idf values over the documents.

Secondly, at the taxonomy level, a feature based measure is used. A part of their simple conceptual structure, taxonomies like Wordnet provide users with additional resources which describe most entities. Information in Wordnet is organized around logical groupings called synsets. Each synset is attached to a description set, a list of synonyms, antonyms, etc..In order to take advantage of the full information package in such rich resources, we have chosen to combine our CD measure also with the feature-based measure proposed by Tversky [16] which assumes that the more common characteristics (i.e. synonyms, antonyms, meronyms, etc..) two objects have and the less non common characteristics they have, the more similar the objects are.

2.2 Evaluation and Results

To evaluate our approach, a benchmark composed of a corpus of 30,000 web pages along with the WordNet taxonomy is used. The web pages are crawled from a set of News web sites (reuters.com, cnn.com, nytimes.com...).

The most intuitive way to evaluate a semantic similarity/distance is to compare machine ratings and human ratings on a same data set. A very common set of 30 word pairs is given by Miller and Charles [9]. M&C asked 38 undergraduate students to rate each pair on a scale from 0 (no similarity) to 4 (perfect synonymy). The average rating of each pair represents a good estimate on how similar the two words are. The correlation between individual ratings of human replication was 0.90 which led many researchers to take 0.90 as the upper bound ratio. For our evaluations, we’ve chosen the M&C subset of 28 words pairs which is the most commonly used subset for that purpose. Note that since our measure calculates distance, the M&C distance will be: $dist = 4 - sim$ where 4 represent the maximum degree of similarity.

When comparing our distance results with the M&C human ratings, the context-dependency *CD* method alone gave a correlation of 0.83 which seems to be a very promising rate. Then, we have combined our measure with others by trying multiple combination strategies(See the previous section). By doing this, we could increase our correlation ratio to 0.89 which is not too far from human correlations of 0.905. This obtained rate led our approach to outperform the existing approaches for semantic similarity (see Table 1).

Similarity method	Type	Correlation with M&C
Human replication	Human	0,901
Rada	Edge-based	0,59
Hirst and St-Onge	Edge-based	0,744
Leacock and Chodorow	Edge-based	0,816
Resnik	Information Content	0,774
Jiang	Information Content	0,848
Lin	Information Content	0,821
CDSB	Context-Dependent	0,830
our multisource measure	Hybrid	0,890

Table 1. Comparison between the principal measures and our two-level measure

Our method shows an interesting result whether on an individual or on a combination scale. A part of its interesting correlation coefficient of 0.83, our CD method has the advantage to be context-dependent, which means that our results are flexible and can vary from one context to another. We argue that our measure could perform better if we "place" human subjects in our corpus context. In other terms, our actual semantic distance values reflect a specific context that does not necessarily match with the context of the human subjects during the R&C experiments.

We presented in this section our new multisource context dependent semantic distance measure between concepts. In the next section, we detail the architecture as well as the different components of our content-based heterogeneous data retrieval system. We will also demonstrate the use of the proposed similarity measure in the global architecture.

3 Our Content-based Retrieval System

We propose an approach that enables semantic retrieval on documents containing heterogeneous information sources. The approach we are proposing is based on the translation of all the data forms into a textual form (following an annotation process for images and a normalization process for text documents for example). That’s where measuring a distance between two words/concepts takes all its importance in the whole retrieval process. Our system architecture (See Figure 1) is composed of three different layers: a low level layer, a high level semantic layer, and a querying layer.

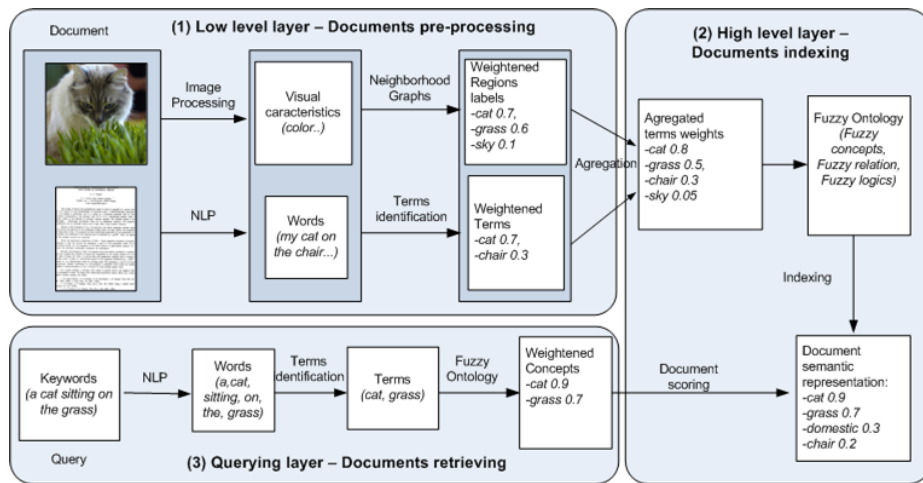


Fig. 1. General Architecture of our semantic based heterogeneous data retrieval framework

3.1 Low level layer – document pre-processing

Image processing One of the most important challenges in imagery is semantics association to an image. Indeed, image processing methods associate for each image a features vector (or vectors) calculated on the image. These features are known as "low level features" (color, texture, etc.). The interrogation of an image database is then done by introducing an image query into the system and its comparison to the available ones using similarity measures [17]. Thus, no semantics is associated to images with this process.

The common way for semantics assignment to an image is annotation. Multimedia data annotation is the task of assigning, for each multimedia document, or for a part of the multimedia document, a keyword or a list of keywords describing its semantic contents. This function can be considered as a mapping between the visual aspects of the multimedia data and their high level characteristics.

To handle images semantics in this work, we adopt the proposed method in [3]. This method is based on an interesting geometrical structure, a relative neighborhood graph [14]. This structure combines, at the same time, a distance measure and the topology of the multidimensional space to determine the neighbors of each point (image in this case) in the considered space. The annotation process is performed into two levels: a) the *indexing level* where images are structured as a neighborhood graph using only their low level features (color, texture, etc.), and b) the *Decision making level* where the neighbors of an unlabeled image are located in the graph and the potential annotations, based on score calculation, are affected to the unknown image. More details about this method are available in [3].

Text processing As images, the raw text of the document is treated separately as well. In order to extract terms from text, classic Natural Language Processing (NLP) techniques are applied. A tokenizer is used first to localize words, numbers, punctuations with their different positions in text. A sentence splitter is used next. Then, a morphological and a syntactic analysis are performed in order to identify respectively grammatical Part Of Speech (POS) for each word which will serve for the lemmatisation process. Lemmatisation involves the reduction of words to their basic lexeme. This normalization step is necessary in order to 1) treat the inflected forms of words and to 2) facilitate the matching with ontology concepts that are usually presented in their lemmatised forms.

After that, we apply the trigram approach on words along with a set of grammatical rules in order to identify the candidate terms to be an ontological concepts. Candidate and non candidate terms are then assigned a normalized tf.idf weight (term frequency/inverse document frequency). At the end, the text part of the document is represented simply by a set of terms and weights.

3.2 High level layer – document indexing

The main goal in this layer is to reach a semantic interpretation of the document. However, keywords or terms extracted at the low level layer are not enough for that purpose. These keywords, are still on an intermediate or object level, and need further treatments to be on a semantic machine understandable level. That’s where our knowledge-unit is involved. Extracted terms from text along with deduced labels from images are all redirected to a domain ontology in order to provide a semantic annotation for document content. Let’s mention that our system purposes are not for a generic domain. Our system deals with specialized corpora along with domain specific ontologies.

Before the concept mapping, an aggregation step is necessary in order to merge obtained lists from the low level layer into one single list. We use the following formula:

$$w_{ki} = \frac{\sum_{j=1}^n w_{kji}}{n} \quad (2)$$

Where:

- w_{ki} is the weight of term w_k in document D_i ;
- n is the total number of parts in document D_i ;
- w_{kji} is the weight of term w_k in part P_j of document D_i .

As we've said earlier, each term is associated to a weight representing its importance in the document. Since we treat particular domains, concepts weights should be a function of their document importance and their domain relative importance. Obviously, in a domain ontology, not all concepts represent the same importance for the target domain. One concept can be more discriminative or more domain-dependent from others, and thus, should be assigned different weights.

That's the reason why we've chosen to use fuzzy ontologies which are an extension of the crisp ontologies [18][10]. Since knowledge can be fuzzy, its representation should be fuzzified. Fuzzification can be integrated to an ontology by using fuzzy concepts, fuzzy relations, and fuzzy logics. It consists of assigning weights to concepts, relation and logical rules. (see figure 2).

Ontology fuzzification is done in an automated manner by making use of our semantic distance method described above. On the one hand, concepts in our domain ontology are assigned weights which correspond to the CD values (described above) that represent concept's dependency to a particular context represented by a text corpus. On the other hand, the 'is-a' relations in the ontology are assigned weights which represent the semantic similarity (as described above) between the two target linked concepts. The semantic similarity is calculated by inverting the semantic distance: $SIM = 1/Dist$.



Fig. 2. Illustration Example of a fuzzy ontology

The aggregated terms with their weights are then sent to the ontology in order to pass from the low level layer to the semantic layer. Certainly, not all terms will be found in the domain ontology. Thus, the result is a set of non-ontological and ontological terms (concepts). The non-ontological terms are then used to semi-automatically enrich the ontology, a process out of the scope of this paper⁶. Terms weight at the low level layer and the concepts weights in the ontology are both used to recalculate each concept weight in the document at the semantic layer using the following formula:

⁶ This part will be detailed in our future publications

$$fw_{ki} = \frac{(2 \times ow_k) + w_{ki}}{3} \quad (3)$$

Where:

- fw_{ki} is the final weight of concept k in document D_i ;
- ow_k is the weight of concept k in the ontology (its CD value);
- w_{ki} is the weight of term k in document D_i (calculated using the above formula);

Note that if a term has been found in the ontology i.e. Ontological term, its ontological weight (ow_k) corresponds to the weight of the concept in that ontology. Otherwise, if the term isn't in the ontology i.e. non-ontological term, its ontological weight is set to 0. So, its weight is divided by 3 in order to penalize this term since we consider that it's not a domain close term.

The interest of this formula is that the calculated concepts weights take into account the term importance in the document and its importance for the considered domain.

We've decided not to make any concept expansion at this indexing level. Experimentations have shown that expanding both documents and queries can result to a lot of sense deviations and imprecisions.

3.3 Querying layer – document retrieving

As we mentioned before, we deal only with ontology-based keywords augmenting. The same process done for text indexing is applied for queries. Query keywords are mapped to the ontology in order to extract concepts. Query will then be divided into terms (non-ontological terms) and concepts (ontological terms). These concepts are then expanded to another linked concepts sharing a link weight greater than a fixed threshold ∂ . Relations weights (which are semantic similarities) in the ontology are used to calculate the deduced concepts weights. Consider the Figure 2. Assume that the concepts *tiger*, *cat*, *dog* were identified in a query q using a threshold $\partial = 0.2$. The concept *dangerous* will be used to expand the query q to q' by using the relations weights between $R(\text{dangerous} - \text{tiger})$ and $R(\text{dangerous} - \text{dog})$ only since $R(\text{dangerous} - \text{cat}) < \partial$.

Finally, the following formula is used to calculate the weight of a document in the database according to the query:

$$w_{iq} = \sum_{l=1}^t (w_{lq} \times fw_{li}) \quad (4)$$

Where:

- w_{iq} is the weight of the document D_i according to the query q ;
- t is the total number of terms within the query q ;
- w_{lq} is the weight of term l in a query q ;
- fw_{li} is the final weight of concept l in the document D_i .

The weight of term l in a query q is determined according to three situations:

- if the query term l is an ontological term, $w_{lq} = 1$;
- if the query term l is inferred, w_{lq} will be the weight of the relation between the origin query concept and the deduced one;
- if the query term l is a non ontological term, its weight will be the maximum of the weights of the query terms obtained by expansion.

Our objective by setting up these weights query is to create a hierarchy of importance between terms. Thus, the query ontological terms are at the top and the expanded ontological ones are at the bottom.

3.4 System evaluation

In this section we present some preliminary results of our approach. To perform the experiments, we built a small corpus of 50 web pages. Each web page contains texts and images. All the pages are related to the domain of animals. We have used an animal domain ontology that we fuzzified.

Each web page is then decomposed into two parts, the first part containing images and the second part containing text. Each document is automatically analyzed and annotated by two lists of keywords: a list of keywords describing the image content and another one describing the text content. These lists are merged using the proposed framework described beforehand.

Semantic based systems evaluation is a very hard task. Since in this case the classical evaluation measures (recall and precision for example) are neither efficient nor significant, we make up a user driven evaluation protocol. We considered ten keyword based queries. The user send his query to the system and obtains a list of documents. At each iteration, he selects the pertinent documents to his query. For each selected document, we take into account the part of interest (image, text, image and text) or the manner of obtaining the result (query expansion or not).

Generally speaking, 79% of the returned documents contain interesting information for the user, which seems to be an interesting rate. The graphic of figure 3 illustrates the average of contribution rate of each data type to the global result.

By considering the graphic, we can note that the different parts of the system: data types (images and texts) and query expansion contribute to the whole result. Text constitutes the most important contribution. We can also note that the combination of image and text gives also interesting results which constitutes a major point.

4 Conclusion and Future Work

Nowadays, retrieving information becomes more and more difficult. This is due especially to the huge volume and the heterogeneity of the modern databases. To interact with these kinds of data, one needs tools which can semantically process them.

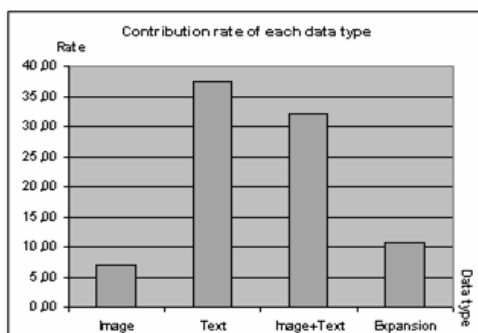


Fig. 3. Contribution rate of each data type to the final results

In this paper, we have shown the importance of considering the context when calculating semantic distance between words/concepts. We've proposed a context-dependent method that takes the taxonomy as a principal knowledge resource, and a text corpus as a distance adaptation resource for the target context. We've proposed also to combine it with other taxonomy-based and corpus-based methods. The results obtained from the experiments show the effectiveness of our approach which led it to outperform the other approaches. We have also proposed a new framework to handle the heterogeneous data retrieval problem. Each document is then decomposed into different components (text, image, etc.) analyzed separately using appropriate techniques. An indexing level ensures the assignment of a significant labels describing the semantic content of each document. The approach supports keywords based querying. Document indexing and query understanding is guided by a domain ontology fuzzified by means of our semantic distance measure. The obtained results show the effectiveness and the interest of the proposed approach.

As for future work, we aim at evaluating the distance measure and comparing it with others by performing a context-driven human ratings, where human subjects will be asked to rank a same set of words pairs in different contexts. The machine correlation computed next according to each context will be able to show more significantly the added-value of our approach. We plan also to test the retrieval approach on more large databases and compare it with other approaches (text retrieval and image retrieval approaches), and to extend this approach by affecting weights for each part of the document reflecting the relative importance for each data type according to the treated domain.

References

1. H. S. Christopher MANNING. *Foundations of statistical natural language processing*. 1999".
2. K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th. Annual Meeting of the Association for*

- Computational Linguistics*, pages 76–83, Vancouver, B.C., 1989. Association for Computational Linguistics.
3. H. Hacid. Neighborhood graphs for semi-automatic annotation of large image databases. In *The 13th International MultiMedia Modeling Conference (MMM'07)*, Singapore., page 586595, 2007.
 4. D. Hindle. Noun classification from predicate-argument structures. In *Meeting of the Association for Computational Linguistics*, pages 268–275, 1990.
 5. J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy, 1997.
 6. C. Leacock, M. Chodorow, and G. A. Miller. Using corpus statistics and wordnet relations for sense identification. *Computational Linguistics*, 24(1):147–165, 1998.
 7. D. Lin. An information-theoretic definition of similarity. In *Proc. 15th International Conf. on Machine Learning*, pages 296–304. Morgan Kaufmann, San Francisco, CA, 1998.
 8. G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, 1995.
 9. G. A. Miller and W. Charles. Contextual correlated of semantic similarity. *Language and Cognitive Processes*, 6:1–28, 1991.
 10. D. Parry. A fuzzy ontology for medical document retrieval. In *ACSW Frontiers '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation*, pages 121–126, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
 11. R. Rada, H. Mili, E. Bicknell, and M. Blettner. Development and application of a metric on semantic nets. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(1):17–30, 1989.
 12. P. Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res. (JAIR)*, 11:95–130, 1999.
 13. A. E. Sayed, H. Hacid, and D. Zighed. A multisource context-dependent approach for semantic distance between concepts. In *DEXA*. Springer, 2007.
 14. G. T. Toussaint. The relative neighborhood graphs in a finite planar set. *Pattern recognition*, 12:261–268, 1980.
 15. P. D. Turney. Mining the Web for synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–??, 2001.
 16. A. Tversky. Features of similarity. *Psychological Review*, 84:327–352, 1977.
 17. R. C. Veltkamp and M. Tanase. Content-based image retrieval systems : A survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, 2000.
 18. D. H. Widyantoro. A fuzzy ontology-based abstract search engine and its user studies. *FUZZ-IEEE*, pages 1291–1294, 2001.
 19. Z. Wu and M. Palmer. Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico, 1994.

Clustering Individuals in Ontologies: a Distance-based Evolutionary Approach

Nicola Fanizzi, Claudia d'Amato, Floriana Esposito

LACAM – Dipartimento di Informatica, Università degli Studi di Bari
Campus Universitario, Via Orabona 4 – 70125 Bari, Italy
{fanizzi|claudia.damato|esposito}@di.uniba.it

Abstract. A clustering method is presented which can be applied to semantically annotated resources in the context of ontological knowledge bases. This method can be used to discover interesting groupings of structured objects through expressed in the standard languages employed for modeling concepts in the Semantic Web. The method exploits an effective and language-independent semi-distance measure over the space of resources, that is based on their semantics w.r.t. a number of dimensions corresponding to a committee of features represented by a group of concept descriptions (discriminating features). A maximally discriminating group of features can be constructed through a feature construction method based on genetic programming. The evolutionary clustering algorithm employed is based on the notion of medoids applied to relational representations. It is able to induce a set of clusters by means of a proper fitness function based on a discernibility criterion. An experimentation with some ontologies proves the feasibility of our method.

1 Introduction

In this work, unsupervised learning is tackled in the context of the standard concept languages used for representing ontologies which are based on Description Logics [1]. In particular, we focus on the problem of *conceptual clustering* [14] of semantically annotated resources. The benefits of clustering in the context of semantically annotated knowledge bases are manifold. Clustering annotated resources enables the definition of new emerging concepts (*concept formation*) on the grounds of the primitive concepts asserted in a knowledge base [6]; supervised methods can exploit these clusters to induce new concept definitions or to refining existing ones *ontology evolution*; intensionally defined groupings may speed-up the task of search and *discovery*; clustering may also suggest criteria for *ranking* the retrieved resources.

Essentially, many existing clustering methods are based on the application of similarity (or density) measures defined over a fixed set of attributes of the domain objects. Classes of objects are taken as collections that exhibit low interclass similarity (density) and high intraclass similarity (density). Often these methods cannot into account any form of *background knowledge* that could characterize object configurations by means of global concepts and semantic relationship. This hinders the interpretation of the outcomes of these methods which is crucial in the Semantic Web perspective.

Thus, clustering methods have aimed at defining groups of objects through conjunctive descriptions based on selected attributes [14]. In the perspective, the expressiveness of the language adopted for describing objects and clusters (concepts) is equally important. Alternative approaches, for terminological representations [1], pursued a different way for attacking the problem, devising logic-based methods for specific languages [11, 6]. The main drawback is that these methods may suffer from noise in the data. This motivates our investigation on similarity-based clustering methods which can be more noise-tolerant, and as language-independent as possible. Specifically we propose a multi-relational extension of effective clustering techniques, which is tailored for the Semantic Web context. It is intended for grouping similar resources w.r.t. a semantic dissimilarity measure.

From a technical viewpoint, upgrading existing algorithms to work on complex representations, like the concept languages used in the Semantic Web, requires fully semantic similarity measures that are suitable for such representations. In particular, as for the original method, one may fix a given number k of clusters of interest, yet this may be hard when scarce knowledge about the domain is available. As an alternative, a partitioning method may be employed up to reaching a minimal threshold value for cluster *quality* (many measures have been proposed in the literature [9, 2]) which makes any further subdivisions useless.

Recently, dissimilarity measures for specific DLs have been proposed [4]. Although they turned out to be quite effective for the inductive tasks, they were still partly based on structural criteria which makes them fail to fully grasp the underlying semantics and hardly scale to any standard ontology language. Therefore, we have devised a family of dissimilarity measures for semantically annotated resources, which can overcome the mentioned limitations [5]. Following the criterion of semantic discernibility of individuals, these measures are suitable for a wide range of concept languages since they are merely based on the discernibility of the input individuals with respect to a fixed committee of features represented by concept definitions. As such the new measures are not absolute, yet they depend on the knowledge base they are applied to. Thus, also the choice of the optimal feature sets deserves a preliminary feature construction phase, which may be performed by means of a randomized search procedure based on *genetic programming*, whose operators are borrowed from recent works on ontology evolution [8].

In this setting, instead of the notion of *centroid* that characterizes the algorithms descending from K-MEANS [9] originally developed for numeric or ordinal features, we recur to the notion of *medoids* [10] as central individuals in a cluster.

Another theoretical problem is posed by the *Open World Assumption* (OWA) that is generally made on the language semantics, differently from the *Closed World Assumption* (CWA) which is standard in machine learning or database query-answering contexts. As pointed out in a seminal paper on similarity measures for DLs [3], most of the existing measures focus on the similarity of atomic concepts within hierarchies or simple ontologies. Moreover, they have been conceived for assessing *concept* similarity, whereas, for other tasks, a notion of similarity between *individuals* is required.

Also the clustering algorithm that we propose employs genetic programming as a learning schema. The clustering problem is solved by considering populations made up

of strings of medoids with different lengths. The medoids are computed according to the semantic measure induced with the methodology mentioned above. On each generation, the strings in the current population is evolved by mutation and cross-over operators, which are also able to change the number of medoids. Thus, this algorithm is also able to suggest autonomously a promising number of clusters.

The paper is organized as follows. Sect. 2 presents the basics representation and the novel semantic similarity measure adopted with the clustering algorithm. This algorithm is presented and discussed in Sect. 3. We report in Sect. 4 an experimental session aimed at assessing the validity of the method on populated ontologies available in the Web. Conclusions and extensions are finally examined in Sect. 5.

2 Semantic Distance Measures

One of the advantages of our method is that it does not rely on a particular language for semantic annotations. Hence, in the following, we assume that resources, concepts and their relationship may be defined in terms of a generic ontology language that may be mapped to some DL language with the standard model-theoretic semantics (see the handbook [1] for a thorough reference).

In this context, a *knowledge base* $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ contains a *TBox* \mathcal{T} and an *ABox* \mathcal{A} . \mathcal{T} is a set of concept definitions. \mathcal{A} contains assertions (facts, data) concerning the world state. Moreover, normally the *unique names assumption* is made on the ABox individuals¹ therein. The set of the individuals occurring in \mathcal{A} will be denoted with $\text{Ind}(\mathcal{A})$.

As regards the inference services, like all other instance-based methods, our procedure may require performing *instance-checking*, which amounts to determining whether an individual, say a , belongs to a concept extension, i.e. whether $C(a)$ holds for a certain concept C .

2.1 A Semantic Semi-Distance for Individuals

Moreover, for our purposes, we need a function for measuring the similarity of individuals rather than concepts. It can be observed that individuals do not have a syntactic structure that can be compared. This has led to lifting them to the concept description level before comparing them (recurring to the approximation of the *most specific concept* of an individual w.r.t. the ABox).

We have developed new measures whose definition totally depends on semantic aspects of the individuals in the knowledge base [5]. On a semantic level, similar individuals should behave similarly with respect to the same concepts. We introduce a novel measure for assessing the similarity of individuals in a knowledge base, which is based on the idea of comparing their semantics along a number of dimensions represented by a committee of concept descriptions. Following the ideas borrowed from ILP [13] and *multi-dimensional scaling*, we propose the definition of totally semantic distance measures for individuals in the context of a knowledge base.

¹ Each individual can be assumed to be identified by its own URI. A Unique Names Assumption can also be made.

The rationale of the new measure is to compare them on the grounds of their behavior w.r.t. a given set of hypotheses, that is a collection of concept descriptions, say $F = \{F_1, F_2, \dots, F_m\}$, which stands as a group of discriminating *features* expressed in the language taken into account.

In its simple formulation, a family of distance functions for individuals inspired to Minkowski's distances can be defined as follows:

Definition 2.1 (dissimilarity measures). Let $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ be a knowledge base. Given set of concept descriptions $F = \{F_1, F_2, \dots, F_m\}$, a family of functions $d_p^F : \text{Ind}(\mathcal{A}) \times \text{Ind}(\mathcal{A}) \mapsto [0, 1]$ defined as follows:

$\forall a, b \in \text{Ind}(\mathcal{A})$

$$d_p^F(a, b) := \frac{1}{m} \left(\sum_{i=1}^m |\pi_i(a) - \pi_i(b)|^p \right)^{1/p}$$

where $p > 0$ and $\forall i \in \{1, \dots, m\}$ the projection function π_i is defined by:

$\forall a \in \text{Ind}(\mathcal{A})$

$$\pi_i(a) = \begin{cases} 1 & \mathcal{K} \models F_i(x) \\ 0 & \mathcal{K} \models \neg F_i(x) \\ 1/2 & \text{otherwise} \end{cases} \quad (1)$$

The case of $\pi_i(a) = 1/2$ corresponds to the case when a reasoner cannot give the truth value for a certain membership query. This is due to the OWA normally made in this context.

It can be proved that these functions have almost all standard properties of distances [5]:

Proposition 2.1 (semi-distance). For a fixed feature set F and $p > 0$ the function d_p^F is a semi-distance.

It cannot be proved that $d_p(a, b) = 0$ iff $a = b$. This is the case of *indiscernible* individuals with respect to the given set of hypotheses F .

Compared to other proposed distance (or dissimilarity) measures [3], the presented function does not depend on the constructors of a specific language, rather it requires only retrieval or instance-checking service used for deciding whether an individual is asserted in the knowledge base to belong to a concept extension (or, alternatively, if this could be derived as a logical consequence).

Note that the π_i functions ($\forall i = 1, \dots, m$) for the training instances, that contribute to determine the measure with respect to new ones, can be computed in advance thus determining a speed-up in the actual computation of the measure. This is very important for the measure integration in algorithms which massively use this distance, such as all instance-based methods.

The underlying idea for the measure is that similar individuals should exhibit the same behavior w.r.t. the concepts in F . Here, we make the assumption that the feature-set F represents a sufficient number of (possibly redundant) features that are able to discriminate really different individuals.

2.2 Feature Set Optimization

Experimentally, we could obtain good results by using the very set of both primitive and defined concepts found in the ontology. The choice of the concepts to be included – *feature selection* – may be crucial. We have devised a specific optimization algorithms founded in *genetic programming* which are able to find optimal choices of discriminating concept committees.

Various optimizations of the measures can be foreseen as concerns its definition. Among the possible sets of features we will prefer those that are able to discriminate the individuals in the ABox.

Since the function is very dependent on the concepts included in the committee of features F , two immediate heuristics can be derived:

- control the number of concepts of the committee, including especially those that are endowed with a real discriminating power;
- finding optimal sets of discriminating features, by allowing also their composition employing the specific constructors made available by the representation language of choice.

Both these objectives can be accomplished by means of randomized optimization techniques especially when knowledge bases with large sets of individuals are available. Namely, part of the entire data can be drawn in order to learn optimal F sets, in advance with respect to the successive usage for all other purposes.

Specifically, we experimented the usage of genetic programming for constructing optimal sets of features. Thus we devised the algorithm depicted in Fig. 1. Essentially the algorithm searches the space of all possible feature committees starting from an initial guess (determined by $\text{MAKEINITIALFS}(\mathcal{K})$) based on the concepts (both primitive and defined) currently referenced in the knowledge base \mathcal{K} .

The outer loop gradually augments the cardinality of the candidate committees. It is repeated until the algorithm realizes that employing larger feature committees would not yield a better fitness value with respect to the best fitness recorded in the previous iteration (with fewer features).

The inner loop is repeated for a number of generations until a stop criterion is met, based on the maximal value of generations maxGenerations or, alternatively, when a minimal threshold for the fitness value minFitness is reached by some feature set in the population, which can be returned.

As regards the $\text{BESTFITNESS}()$ routine, it computes the best feature committee in a vector in terms of their *discernibility* [12, 7]. For instance, given the whole set of individuals $IS = \text{Ind}(\mathcal{A})$ (or just a sample to be used to induce an optimal measure) the fitness function may be:

$$\text{DISCERNIBILITY}(F) = \frac{1}{|IS|^2} \sum_{(a,b) \in IS^2} \sum_{i=1}^{|F|} \frac{|\pi_i(a) - \pi_i(b)|}{2 \cdot |F|}$$

As concerns finding candidate sets of concepts to replace the current committee ($\text{GENERATEOFFSPRINGS}()$ routine), the function was implemented by recurring to simple transformations of a feature set:

- choosing $F \in \text{currentFSs}$;
- randomly selecting $F_i \in F$;
 - replacing F_i with $F'_i \in \text{RANDOMMUTATION}(F_i)$ randomly constructed, or
 - replacing F_i with one of its refinements $F'_i \in \text{REF}(F_i)$

Refinement of concept description may be language specific. E.g. for the case of \mathcal{ALC} logic, refinement operators have been proposed in [8].

This is iterated till a suitable number of offsprings is generated. Then these offspring feature sets are evaluated and the best ones are included in the new version of the `currentFSs` array; the minimal fitness value for these feature sets is also computed. As mentioned, when the while-loop is over the current best fitness is compared with the best one computed for the former feature set length; if an improvement is detected then the outer repeat-loop is continued, otherwise (one of) the former best feature set(s) is selected for being returned as the result of the algorithm.

3 Evolutionary Clustering Around Medoids

The conceptual clustering procedure consists of two phases: one that detects the clusters in the data and the other that finds an intensional definition for the groups of individuals detected in the former phase.

The first clustering phase implements a genetic programming learning scheme, where the designed representation for the competing genes is made up of strings (lists) of individuals of different lengths, where each individual stands as prototypical for one cluster. Thus, each cluster will be represented by its prototype recurring to the notion of *medoid* [10, 9] on a categorical feature-space w.r.t. the distance measure previously defined. Namely, the medoid of a group of individuals is the individual that has the lowest distance w.r.t. the others. Formally, given a cluster $C = \{a_1, a_2, \dots, a_n\}$, the medoid is defined:

$$m = \text{medoid}(C) = \underset{a \in C}{\text{argmin}} \sum_{j=1}^n d(a, a_j)$$

The algorithm performs a search in the space of possible clusterings of the individuals optimizing a fitness measure maximizing discernibility of the individuals of the different clusters (inter-cluster separation) and the intra-cluster similarity measured in terms of our metric.

The second phase is more language dependent. The various cluster can be considered as training examples for a supervised algorithm aimed at finding an intensional DL definition for one cluster against the counterexamples, represented by individuals in different clusters [11, 6].

3.1 The Clustering Algorithm

The proposed clustering algorithm can be considered as an extension of methods based on genetic programming, where the notion of cluster prototypical instance of centroid, typical of the numeric feature-vector data representations, is replaced by that of medoid

```

FeatureSet OPTIMIZEFS( $\mathcal{K}$ , maxGenerations, minFitness)
input:
     $\mathcal{K}$ : current knowledge base
    maxGenerations: maximal number of generations
    minFitness: minimal fitness value
output:
    FeatureSet: FeatureSet
begin
currentBestFitness = 0; formerBestFitness = 0;
currentFSs = MAKEINITIALFS( $\mathcal{K}$ ); formerFSs = currentFSs;
repeat
    fitnessImproved = false;
    generationNumber = 0;
    currentBestFitness = BESTFITNESS(currentFSs);
    while (currentBestFitness < minFitness) or (generationNumber < maxGenerations)
        begin
        offsprings = GENERATEOFFSPRINGS(currentFSs);
        currentFSs = SELECTFROMPOPULATION(offsprings);
        currentBestFitness = BESTFITNESS(currentFSs);
        ++generationNumber;
        end
        if (currentBestFitness > formerBestFitness) and (currentBestFitness < minFitness) then
            begin
            formerFSs = currentFSs;
            formerBestFitness = currentBestFitness;
            currentFSs = ENLARGEFS(currentFSs);
            end
        else fitnessImproved = true;
        end
    until fitnessImproved;
return BEST(formerFSs);
end

```

Fig. 1. Feature set optimization algorithm based on Genetic Programming.

[10]: each cluster is represented by one of the individuals in the cluster, the medoid, i.e., in our case, the one with the lowest average distance w.r.t. all the others individuals in the cluster. In the algorithm, a genome will be represented by a list of medoids $G = \{m_1, \dots, m_k\}$. Per each generation those that are considered as best w.r.t. a fitness function are selected for passing to the next generation. Note that the algorithm does not prescribe a fixed length of these lists (as, for instance in K-MEANS and its extensions [9]), hence it should be able to detect an optimal number of clusters for the data at hand.

Fig. 2 reports a sketch of the clustering algorithm. After the call to the initialization procedure INITIALIZE() returning the randomly generated initial population of medoid strings (currentPopulation) in a number of popLength, it essentially consists of the typical generation loop of genetic programming.

```

medoidVector ECM(maxGenerations, minGap)
input:
    maxGenerations: max number of iterations;
    minGap: minimal gap for stopping the evolution;
output:
    medoidVector: list of medoids
begin
INITIALIZE(currentPopulation, popLength);
while (generation ≤ maxGenerations) and (gap > minGap)
    begin
        offsprings = GENERATEOFFSPRINGS(currentPopulation);
        fitnessVector = COMPUTEFITNESS(offsprings);
        currentPopulation = SELECT(offsprings, fitnessVector);
        gap = (FITNESS[popLength] - FITNESS[1]);
        generation++;
    end
return currentPopulation[0]; // best genome
end

```

Fig. 2. ECM: the EVOLUTIONARY CLUSTERING AROUND MEDOIDS algorithm.

At each iteration this computes the new offsprings of current best clusterings represented by `currentPopulation`. This is performed by suitable genetic operators explained in the following. The `fitnessVector` recording the quality of the various offsprings (i.e. clusterings) is then updated, which is used to select the best offsprings that survive, passing to the next generation.

The fitness of a genome $G = \{m_1, \dots, m_k\}$ is computed by distributing all individuals among the clusters ideally formed around the medoids in that genome. Per each medoid m_i , $i = 1, \dots, k$, let C_i be such a cluster. Then, the fitness is computed:

$$\text{FITNESS}(G) = \sqrt{k+1} \sum_{i=1}^k \sum_{x \in C_i} d_p(x, m_i)$$

This measure is to be minimized. The factor $\sqrt{k+1}$ is introduced in order to penalize those clusterings made up of too many clusters that could enforce the minimization in this way (e.g. by proliferating singletons).

The loop condition is controlled by two factors the maximal number of generation (the `maxGenerations` parameter) and the difference (`gap`) between the fitness of best and of the worst selected genomes in `currentPopulation` (which is supposed to be sorted in ascending order, 1 thru `popLength`). Thus another stopping criterion is met when this `gap` becomes less than the minimal `gap` `minGap` passed as a parameter to the algorithm, meaning that the algorithm has reached a (local) minimum.

It remains to specify the nature of the `GENERATEOFFSPRINGS` procedure function and the number of such offsprings, which may as well be another parameter of the ECM algorithm. Three mutation and one crossover operators are implemented:

DELETION(G) drop a randomly selected medoid:
 $G := G \setminus \{m\}, m \in G$
 INSERTION(G) select $m \in \text{Ind}(\mathcal{A}) \setminus G$ that is added to G :
 $G := G \cup \{m\}$
 REPLACEMENTWITHNEIGHBOR(G) randomly select $m \in G$ and replace it with $m' \in \text{Ind}(\mathcal{A}) \setminus G$ such that $\forall m'' \in \text{Ind}(\mathcal{A}) \setminus G d(m, m') \leq d(m, m'')$:
 $G' := (G \setminus \{m\}) \cup \{m'\}$
 CROSSOVER(G_A, G_B) select subsets $S_A \subset G_A$ and $S_B \subset G_B$ and exchange them between the genomes:
 $G_A := (G_A \setminus S_A) \cup S_B$ and $G_B := (G_B \setminus S_B) \cup S_A$

A (10+60) selection strategy has been implemented, with n_e numbers indicating, resp., the number of parents selected for survival and the number of their offsprings.

3.2 The Supervised Learning Phase

Each cluster may be labeled with an intensional concept definition which characterizes the individuals in the given cluster while discriminating those in other clusters [11, 6]. Labeling clusters with concepts can be regarded as a number of supervised learning problems in the specific multi-relational representation targeted in our setting [8]. As such it deserves specific solutions that are suitable for the DL languages employed.

A straightforward solution may be found, for DLs that allow for the computation of (an approximation of) the *most specific concept* (MSC) and *least common subsumer* (lcs) [1] (such as \mathcal{ALC}). This may involve the following steps:
 given a cluster of individuals node_j

- **for each** individual $a_i \in \text{node}_j$
 do compute $M_i := \text{msc}(a_i)$ w.r.t. \mathcal{A} ;
- **let** $\text{MSCs}_j := \{M_i \mid a_i \in \text{node}_j\}$;
- **return** $\text{lcs}(\text{MSCs}_j)$

As an alternative, algorithms for learning concept descriptions expressed in DLs may be employed [8]. Indeed, concept formation can be cast as a supervised learning problem: once the two clusters at a certain level have been found, where the members of a cluster are considered as positive examples and the members of the dual cluster as negative ones. Then any concept learning method which can deal with this representation may be utilized for this new task.

3.3 Discussion

The representation of centers by means of medoids has two advantages. First, it presents no limitations on attributes types, and, second, the choice of medoids is dictated by the location of a predominant fraction of points inside a cluster and, therefore, it is lesser sensitive to the presence of outliers. In K-MEANS case a cluster is represented by its centroid, which is a mean (usually weighted average) of points within a cluster. This works conveniently only with numerical attributes and can be negatively affected by a single outlier.

A PAM algorithm has several favorable properties. Since it performs clustering with respect to any specified metric, it allows a flexible definition of similarity. This flexibility is particularly important in biological applications where researchers may be interested, for example, in grouping correlated or possibly also anti-correlated elements. Many clustering algorithms do not allow for a flexible definition of similarity, but allow only Euclidean distance in current implementations. In addition to allowing a flexible distance metric, a PAM algorithm has the advantage of identifying clusters by the medoids. Medoids are robust representations of the cluster centers that are less sensitive to outliers than other cluster profiles, such as the cluster means of K-MEANS. This robustness is particularly important in the common context that many elements do not belong exactly to any cluster, which may be the case of the membership in DL knowledge bases, which may be not ascertained given the OWA.

4 Experimental Validation

A comparative evaluation of the method is not possible yet, since to the best of our knowledge, there is no similar algorithm which can cope with complex DL languages such as those indicated in the following Tab. 1. The only comparable (logical) approaches to clustering DL KBs are suitable for limited languages only (e.g. see [11, 6]).

The clustering procedure was validated through some standard internal indices [9, 2]. As pointed out in several surveys on clustering, it is better to use a different criterion for the clustering algorithm (e.g. for choosing the candidate cluster to bisection) and for assessing the quality of its resulting clusters.

To this purpose, we modify a generalization of Dunn's index [2] to deal with medoids. Let $P = \{C_1, \dots, C_k\}$ be a possible clustering of n individuals in k clusters. The index can be defined:

$$V_{GD}(P) = \min_{1 \leq i \leq k} \left\{ \min_{\substack{1 \leq j \leq k \\ i \neq j}} \left\{ \frac{\delta_p(C_i, C_j)}{\max_{1 \leq h \leq k} \{\Delta_p(C_h)\}} \right\} \right\}$$

where δ_p is the Hausdorff distance for clusters derived from d_p (defined: $\delta_p(C_i, C_j) = \max\{d_p(C_i, C_j), d_p(C_j, C_i)\}$, where $d_p(C_i, C_j) = \max_{a \in C_i} \{\min_{b \in C_j} \{d_p(a, b)\}\}$) while the cluster diameter measure Δ_p is defined:

$$\Delta_p(C_h) = \frac{2}{|C_h|} \left(\sum_{c \in C_h} d_p(c, m_h) \right)$$

which is more noise-tolerant w.r.t. other standard measures.

The other measures employed are more standard: the mean square error (WSS, a measure of cohesion) and the silhouette measure [10].

For the experiments, a number of different ontologies represented in OWL were selected, namely: FSM, SURFACE-WATER-MODEL, TRANSPORTATION and NEWTESTAMENTNAMES from the Protégé library², the FINANCIAL ontology³ employed as a

² <http://protege.stanford.edu/plugins/owl/owl-library>

³ <http://www.cs.put.poznan.pl/alawrynowicz/financial.owl>

Table 1. Ontologies employed in the experiments.

ontology	DL	#concepts	#object prop.	#data prop.	#individuals
FSM	<i>SO\mathcal{F}(D)</i>	20	10	7	37
S.-W.-M.	<i>ALCC$\mathcal{O}$$\mathcal{F}$(D)</i>	19	9	1	115
TRANSPORTATION	<i>ACC</i>	44	7	0	250
NTN	<i>SHLF(D)</i>	47	27	8	676
FINANCIAL	<i>ALCIF</i>	60	16	0	1000

Table 2. Results of the experiments: average value (\pm std. deviation) and min–max value ranges.

ONTOLOGY	DUNN'S	WSS	SILHOUETTE
FSM	.221 (\pm .003)	30.254 (\pm 11.394)	.998 (\pm .005)
	.212–.222	14.344–41.724	.985–1.000
S.-W.-M.	.333 (\pm .000)	11.971 (\pm 11.394)	1.000 (\pm .000)
	.333–.333	7.335–13.554	1.000–1.000
TRANSPORTATION	.079 (\pm .000)	46.812 (\pm 5.944)	.976 (\pm .000)
	.079–.079	39.584–57.225	.976–.976
NTN	.058 (\pm .003)	96.155 (\pm 24.992)	.986 (\pm .007)
	.056–.063	64.756–143.895	.974–.996
FINANCIAL	.237 (\pm .000)	130.863 (\pm 24.117)	.927 (\pm .034)
	.237–.237	99.305–163.259	.861–.951

testbed for the PELLET reasoner. Table 1 summarizes important details concerning the ontologies employed in the experimentation. A variable number of assertions (facts) was available per single individual in the ontology.

For each populated ontology, the experiments have been repeated for 10 times. In the computation of the distances between individuals (the most time-consuming operation) all concepts in the ontology have been used for the committee of features, thus guaranteeing meaningful measures with high redundancy. The PELLET 1.4 reasoner was employed to compute the projections. An overall experimentation of 10 repetitions on a single ontology took from a few minutes to less than one hour on a 2.5GHz (512Mb RAM) Linux Machine.

The outcomes of the experiments are reported in Tab. 2. It is possible to note that the silhouette measure is quite close its optimal value (1), thus providing an absolute indication for the quality of the obtained clusterings. The other two indices, Dunn's and WSS may be employed as a suggestion on whether to accept or not the (number of) clusters suggested by the algorithm. Namely, among the various repetitions, those final clusterings had to be preferred whose values maximize these two indices.

5 Conclusions and Future Work

This work has presented a clustering for (multi-)relational representations which are standard in the Semantic Web field. Namely, it can be used to discover interesting groupings of semantically annotated resources in a wide range of concept languages.

The method exploits a novel dissimilarity measure, that is based on the resource semantics w.r.t. a number of dimensions corresponding to a committee of features represented by a group of concept descriptions (discriminating features). The algorithm, is an adaptation of clustering procedures employing medoids since complex representations typical of the ontology in the Semantic Web are to be dealt with.

Better fitness functions may be investigated for both the distance optimization procedure and the clustering one. We are also devising extensions that are able to produce hierarchical clusterings.

References

- [1] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2003.
- [2] J.C. Bezdek and N.R. Pal. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics*, 28(3):301–315, 1998.
- [3] A. Borgida, T.J. Walsh, and H. Hirsh. Towards measuring similarity in description logics. In I. Horrocks, U. Sattler, and F. Wolter, editors, *Working Notes of the International Description Logics Workshop*, volume 147 of *CEUR Workshop Proceedings*, Edinburgh, UK, 2005.
- [4] C. d’Amato, N. Fanizzi, and F. Esposito. Reasoning by analogy in description logics through instance-based learning. In G. Tummarello, P. Bouquet, and O. Signore, editors, *Proceedings of Semantic Web Applications and Perspectives, 3rd Italian Semantic Web Workshop, SWAP2006*, volume 201 of *CEUR Workshop Proceedings*, Pisa, Italy, 2006.
- [5] N. Fanizzi, C. d’Amato, and F. Esposito. Induction of optimal semi-distances for individuals based on feature sets. In *Working Notes of the International Description Logics Workshop, DL2007*, volume 250 of *CEUR Workshop Proceedings*, Bressanone, Italy, 2007.
- [6] N. Fanizzi, L. Iannone, I. Palmisano, and G. Semeraro. Concept formation in expressive description logics. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *Proceedings of the 15th European Conference on Machine Learning, ECML2004*, volume 3201 of *LNAI*, pages 99–113. Springer, 2004.
- [7] S. Hirano and S. Tsumoto. An indiscernibility-based clustering method. In X. Hu, Q. Liu, A. Skowron, T. Y. Lin, R. Yager, and B. Zhang, editors, *2005 IEEE International Conference on Granular Computing*, pages 468–473. IEEE, 2005.
- [8] L. Iannone, I. Palmisano, and N. Fanizzi. An algorithm based on counterfactuals for concept learning in the semantic web. *Applied Intelligence*, 26(2):139–159, 2007.
- [9] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.
- [10] L. Kaufman and Rousseeuw. P.J. *Finding Groups in Data: an Introduction to Cluster Analysis*. John Wiley & Sons, 1990.
- [11] J.-U. Kietz and K. Morik. A polynomial approach to the constructive induction of structural knowledge. *Machine Learning*, 14(2):193–218, 1994.
- [12] Z. Pawlak. *Rough Sets: Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, 1991.
- [13] M. Sebag. Distance induction in first order logic. In S. Džeroski and N. Lavrač, editors, *Proceedings of the 7th International Workshop on Inductive Logic Programming, ILP97*, volume 1297 of *LNAI*, pages 264–272. Springer, 1997.
- [14] R. E. Stepp and R. S. Michalski. Conceptual clustering of structured objects: A goal-oriented approach. *Artificial Intelligence*, 28(1):43–69, Feb. 1986.

Data mining of Multi-categorized Data

Akinori Abe^{1),2)}, Norihiro Hagita^{1),3)}, Michiko Furutani¹⁾, Yoshiyuki Furutani¹⁾, and Rumiko Matsuoka¹⁾

1) International Research and Educational Institute for Integrated Medical Science (IREIIMS), Tokyo Women's Medical University
8-1 Kawada-cho, Shinjuku-ku, Tokyo 162-8666 JAPAN

2) ATR Knowledge Science Laboratories

3) ATR Intelligent Robotics and Communication Laboratories
2-2-2, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 JAPAN
ave@ultimaVI.arc.net.my, hagita@atr.jp, {michi,yoshi,rumiko}@imcir.twmu.ac.jp

Abstract. At the International Research and Educational Institute for Integrated Medical Sciences (IREIIMS) project, we are collecting complete medical data sets to determine relationships between medical data and health status. Since the data include many items which will be categorized differently, it is not easy to generate useful rule sets. Sometimes rare rule combinations are ignored and thus we cannot determine the health status correctly. In this paper, we analyze the features of such complex data, point out the merit of categorized data mining and propose categorized rule generation and health status determination by using combined rule sets.

1 Introduction

Medical science and clinical diagnosis and treatment has been progressing rapidly in recent years with each field becoming more specialized and independent. As a result, cooperation and communication among researchers in the two fields has decreased which has led to problems between both communities, not only in terms of medical research but also with regard to clinical treatment. Therefore, an integrated and cooperative approach to research between medical researchers and biologists is needed. Furthermore, we are living in a changing and quite complex society, so important knowledge is always being updated and becoming more complex. Therefore, integrated and cooperative research needs to be extended to include engineering, cultural science, and sociology. As for medical research, the integration of conventional (Western) and unconventional (Eastern) medical research, which should be fundamentally the same but in fact are quite different, has been suggested.

With this situation in mind, we propose a framework called Cyber Integrated Medical Infrastructure (CIMI) [Abe et al., 2007a] which is a framework of integrated management of clinical data on computer networks consisting of a database, a knowledge base, and an inference and learning component, which are connected to each other in the network. In this framework, medical information

(e.g. clinical data) is analyzed or data mined to build a knowledge base for the prediction of all possible diseases and to support medical diagnosis.

For medical data mining, several techniques such as Inductive Logic Programming (ILP), statistical methods, decision tree learning, Rough Sets and KeyGraph have been applied (e.g. [Ichise and Numao, 2005], [Ohsawa, 2003] and [Tsumoto, 2004] and acceptable results have been obtained. We have also applied C4.5 [Quinlan, 1993] to medical data and obtained acceptable results. However, we used incomplete medical data sets which lack many parts of the data, since the data were collected during clinical examination. To save costs, physicians do not collect unnecessary data. For instance, if certain data are not related to the patient's situation, physicians will not collect them. If parts of the data are missing, even if we can collect many data sets, some of them are ignored during simple data mining procedures. To prevent this situation, we need to supplement missing data sets. However, it is difficult to automatically supplement the missing data sets. In fact, to supplement missing data sets, Ichise proposed a non-linear supplemental method [Ichise and Numao, 2003], but when we collected data from various patients it was difficult to guess relationship among the data, so we gave up to the idea of introducing such a supplemental method. Instead, we introduced a boosting method which estimates the distribution of original data sets from incomplete data sets and increases data by adding Gaussian noise [Abe et al., 2004]. We obtained results with robustness but we could not guarantee the results. In addition, when we used data sets collected in clinical inspections, we could only collect a small number of incomplete data sets. Therefore, in the International Research and Educational Institute for Integrated Medical Science (IREHIMS) project, we decided to collect complete medical data sets. Even if we collect considerable size of complete data sets, we still have additional problems. The data include various types of data. That is, they contain data, for instance, of persons with lung cancer, those with stomach cancer etc. It is sometimes hazardous to use such mixed and complex data to perform data mining. In [Abe et al., 2007a], we pointed out that if we deal with multiple categorized (mixed) data, it is rather difficult to discover hidden or potential knowledge and we proposed integrated data mining. In [Abe et al., 2007b], we proposed an interface for medical diagnosis support which helps the user to discover hidden factors for the results. In this study, we introduce the interface to help to discover rare, hidden or potential data relationships.

In this paper, we analyze the collected medical data consisting of multiple categorized items then propose categorized rule generation (data mining) and a health level¹ determination method by applying the combined rule sets. In Section 2, we briefly describe the features of the collected medical data. In Section 3, we analyze (data mine) the collected data by C4.5 and apply generated rule sets to the medical data to determine the patients' situations. In section 4, we analyze the data mined results to point out the limitation of simple data mining of the collected data. In section 5, we suggest several strategies to deal with complex data that enable better health level determination.

¹ Health level is explained in section 2.

2 Features of the collected medical data

In this section, we analyze and describe features of the medical data collected in the IREHIMS project.






Health Level		Health Condition	(%)
I		Excellent	0
II		Good	10
III		Fair	60
IV		Needs an improvement in Lifestyle	25
V		Needs a precise examination and therapy	5

Fig. 1. Health levels

To construct the database in CIMI, we are now collecting various types of medical data, such as those obtained in blood and urine tests. We have collected medical data from about 1800 persons (For certain persons, the data were collected more than once.) and more than 130 items are included in the medical data of each person. Item sets in the data are, for instance, total protein, albumin, serum protein fraction- α 1-globulin, Na, K, Ferritin, total acid phosphatase, urobilinogen, urine acetone, mycoplasma pneumoniae antibody, cellular immunity, immunosuppressive acidic protein, Sialyl Le X-i antigen, and urine β 2-microglobulin. In addition, health levels are assigned by physicians resulting from the medical data and by clinical interviews. Health levels that express the health status of patients are defined based on *Tumor stages* [Kobayashi and Kawakubo, 1994] and modified by Matsuoka. Categorization of the health levels is shown in Fig. 1 (“%” represents a typical distribution ratio of persons in each level.). Persons at levels I and II can be regarded as being healthy, but those at levels III, IV, and V can possibly develop cancer. In [Kobayashi and Kawakubo, 1994], level III is defined as the stage before the shift to preclinical cancer, level IV is defined as conventional stage 0 cancer (G0), and level V is defined as conventional stages 1–4 cancer (G1–G4).

As shown in Fig. 1, Kobayashi categorized health levels into 5 categories. For more detailed analysis, Matsuoka categorized health levels into 8 categories which are 1, 2, 3, 4a, 4b, 4c, 5a, and 5b, since levels 4 and 5 include many clients’ data. Table 1 shows the distribution ratio of health levels of the collected

Table 1. Health levels.

health level	1	2	3	4a	4b	4c	5a	5b
ratio (%)	0.0	0.0	3.09	17.23	46.99	19.22	10.77	2.71
ratio (%)	0.0	0.0	3.09	83.44			13.48	

data. The distribution ratio is quite different from that shown in Fig. 1, as we are currently collecting data from office workers (aged 40 to 50 years old) but not from students or younger persons. Accordingly the data distribution shifts to level 5 and 80% of the data are assigned to level 4. This imbalance and distribution might influence the data mining results. However, in this study, we did not apply any adjustments as we have no idea or models for proper adjustment. Adjustments will be proposed after analysis of the data sets.

3 Analysis of the data

In this section, we analyze the collected medical data. First, we simply apply C4.5 to obtain relationships between the health levels and medical data sets. Then we apply the obtained relationships to medical data to estimate the patient's health situation. If we have the actual health level information, we can determine whether obtained rule sets are good or not. In addition, we can obtain the features of medical data sets.

3.1 Data analysis by C4.5

First to determine the features of the data, we simply applied C4.5 to the collected data. To check the effect of data size, we analyzed both 1200 and 1500 medical data sets that were chronologically² extracted from 1800 medical data sets. Both results are shown below.

– 1200 medical data sets

```

ICTP <= 5.8
| TK activity <= 5.4
| | CEA <= 4.1
| | | EBV-VCA-IgG <= 640
| | | |  $\gamma$ -seminoprotein <= 2.15
| | | | | Chloride (Cl) <= 96
| | | | | | CK <= 82 : 4b
| | | | | | CK > 82 : 4c
| | | | | | Chloride (Cl) > 96
...

```

² “Chronological” extraction is performed because we aim to use generated rule sets to determine health levels. It is natural to use previous data sets to generate models for future estimation.

– 1500 medical data sets

```
TK activity <= 5.4
| ICTP <= 5.8
| | CYFRA <= 2.1
| | |  $\gamma$ -seminoprotein <= 2.1
| | | | EBV-VCA-IgG <= 640
| | | | | Chloride (Cl) <= 96
| | | | | | B-Cell(CD20) <= 22 : 4b
| | | | | | B-Cell(CD20) > 22 : 4c
| | | | | | Chloride (Cl) > 96
| | | | | | CEA <= 4.2
...

```

Since the data set size will not be large enough for general data mining, there are some differences between the results of 1200 medical data sets and 1500 medical data sets. If we can collect more medical data sets, we will be able to obtain more stable results. Nevertheless, even from current data, acceptable results can be obtained. If we focus on the first few lines, they are almost the same.

3.2 Applying the result to determine health levels

Next, we applied the obtained rule sets (decision trees) to the rest of the collected medical data to determine the health levels. For the results from 1200 medical data sets, we can estimate the health levels of about 600 (=1800–1200) persons' health levels. For the result from 1500 medical data sets, we can estimate the health levels of about 300 (=1800–1500) persons' health levels. A series of nodes in a decision tree is used to determine the health level. Currently the combination of multiple decision tree clusters is not considered. We follow a decision tree from the root point (top of a decision tree) to a leaf. Table 2 shows accuracies of the results. For difference, we mean that if the estimated health level is 4b and the actual level is 4c, then the difference is -1 . If the estimated health level is 4b and the actual level is 3, the difference is $+2$. Of course, if the estimated and actual health levels are the same, the difference is 0. An exact estimation (0 estimation) ratio is about 40%. Generally, this is not a good result. However, if we regard both $+1$ and -1 as correct estimations, the ratio becomes about 85% which is usually regarded as a good result. Even for us, it is sometimes rather difficult to distinguish level 4a from 4b, so it might be acceptable to extend the correct estimation to $+1$ and -1 . In addition, we could not find superiority due to the size of the data sets. In fact, as for an exact estimation, rules generated from 1200 data sets are better than those generated from 1500 data sets. The difference in number is only 300, so it might be difficult to find a superiority due to the size of the data sets. From the accuracy ratio, in medical situations, it might be difficult to use the generated rule sets as they are. However, with a certain modification or improvement, they can help to determine the health levels of patients during medical diagnosis.

Table 2. Health level estimation

	from 1200 data	from 1500 data
Difference	Correct ratio (%)	Correct ratio (%)
-3	1.4	1.1
-2	6.3	3.7
-1	19.2	23.1
0	42.4	38.4
+1	23.9	25.4
+2	4.8	6.7
+3	1.8	1.1
+4	0.2	0.0

After obtaining the results, for us, it is more interesting and significant for us to find the reasons for incorrect estimations. From these reasons, we can propose a proper strategy for reducing incorrect estimations. In the next section, we analyze the results in detail and try to find the reasons for incorrect estimation.

4 Analysis of results

In this section, we analyze the reasons for incorrect estimations by using the interface proposed in [Abe et al., 2007b] which can deal with data interactively. Figure 2 shows a result (decision tree) obtained in the web interface. In the browser, the left tab shows ID lists of a person such as 1186 and the right tab shows a decision tree. In the decision tree "White blood cell differentiation:Neutro > 66.1: 5a(2/1/0)" can be interpreted "... and if White blood cell differentiation:Neutro > 66.1 then the health level is 5a." In addition, the generated rule set (series of nodes) can explain three persons of which two explanations are correct and one is incorrect. When the user clicks the link point "5a(2/1/0)," another browser appears (Fig. 3). In the browser, <1199,2007-02-21,4a> shows that an estimation of ID 1199 is incorrect (blue colored and the assigned level is 4a; the actual level is 5a). When the user clicks the link point "<1199,2007-02-21,4a>," another browser appears (Fig. 4). In this case, even if we review the medical data, if we are not physicians we cannot determine whether the person is in level 4a or 5a. However we can ask physicians for reasons or confirmation. In contrast, we also found a case where, in spite of the assigned health level being 4b, the estimated level was 3. Similarly to the above case, we checked the data to find that NSE is 7.8 but it was not considered during the estimation. Actually in the decision tree the following pattern appears but in the estimation process, the system cannot refer to the rule. That is, the pattern does not appear in the inference path to determine the health levels.

```
NSE > 7.2 .....
  Lipase <= 20
    Albumin <= 4.3: 4a
```

```

Albumin > 4.3: ....
.....
Acid Phosphatase(ACP) <= 8.3 : 4b

```

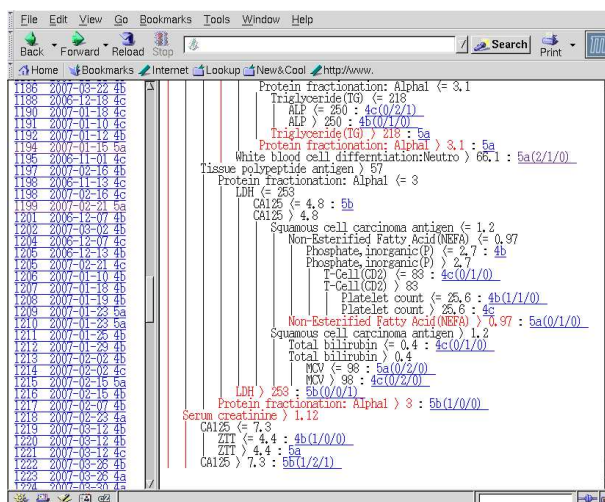


Fig. 2. Analysis result shown in the proposed interface.

In fact, as shown in Table 1, only 3% are in level 3. Thus the number of examples is too small to generate proper rules or models, so it will be necessary to collect more data in level 3. In fact, none of the persons in health level 3 have been estimated correctly. Cases where the an estimation of the health level is 4a may be satisfactory, but some cases are estimated as 4b or 4c which cannot be regarded as correct. From these cases, we can say that the number of examples in health level 3 is too small to generate a proper model.

First, for this type of phenomenon, we assume that a rare case might be present in level 4x. That is, some of the items that would place a person in level 4x are rare. However, as shown above, the decision tree includes rules of NSE for level 4x. Accordingly, since a combination of NSE and certain factors for level 4 are rare, the system cannot determine the health levels correctly by using such rules.

As suggested in section 2, imbalance and the non-standard distribution of data play a negative role in modeling. In fact, we intentionally remove any imbalance in the data, that is, we reduce the data according to the number of persons in level 3. The data contain only 300 samples (all levels have about 50 data), so the number is too small to generate a proper model. The result is not satisfactory as a diagnosis system but shows that balanced data will generate

lower bound	Inspection items	upper bound	Value Range	Value Range
5.4	White blood cell counts(WBC)			TK activity
	Red blood cell counts(RBC)			Protein fractionation:Albumin
	Hemoglobin			Protein fractionation:α1
	Hematocrit			Protein fractionation:α2
	Platelet count			Protein fractionation:β
	MCV			Protein fractionation:γ
	MCH			CK-MB
	MCHC			ALP type1
68.1	White blood cell differentiation:Neutro			ALP type2
	White blood cell differentiation:Stab			ALP type3
	White blood cell differentiation:Seg			ALP type4
	White blood cell differentiation:Lymphocyte			ALP type5
	White blood cell differentiation:Mono			ALP type6
	White blood cell differentiation:Eosino			Lipase
	White blood cell differentiation:Baso			Pepsinogen1
				Pepsinogen2
				Pepsinogen1/2 ratio
				Pepsinogen total power
				Bolysamine urine

Fig. 3. Health level estimation result

Items	value	Item
White blood cell counts(WBC)	6.9	TK activity
Red blood cell counts(RBC)	3.68	Protein fractionation:Albumin
Hemoglobin	12.3	Protein fractionation:α1
Hematocrit	37.9	Protein fractionation:α2
Platelet count	22.2	Protein fractionation:β
MCV	103	Protein fractionation:γ
MCH	33.4	CK-MB
MCHC	32.5	ALP type1
White blood cell differentiation:Neutro	87	ALP type2
White blood cell differentiation:Stab	4	ALP type3
White blood cell differentiation:Seg	83	ALP type4
White blood cell differentiation:Lymphocyte	10	ALP type5
White blood cell differentiation:Mono	3	ALP type6
White blood cell differentiation:Eosino	0	Lipase
White blood cell differentiation:Baso	0	Pepsinogen1
		Pepsinogen2
		Pepsinogen1/2 ratio
		Pepsinogen total power
		Bolysamine urine

Fig. 4. Data of ID:1199

better models for medical diagnosis. If we can collect many data, we will be able to modify them. However currently this is not possible, so to overcome this problem, it is necessary to apply or develop another type of modeling.

5 Proposal for the treatment of complex data

5.1 Categorized rule generation and application

In the previous section, we analyzed the collected medical data and showed several problems with the data mining of the collected data and the application of generated rules to determine health levels. We found at least three features of the collected medical data as shown below;

- 1) The collected data showed imbalance and did not follow the standard distribution.
- 2) Proper models might not be generated for health level 3 due to the small number of examples in this level.
- 3) Parts of the generated rules cannot be referred to in health level determination.

Due to the features of the collected medical data, a simple data mining method cannot be applied to them to obtain a satisfying result. As for 3), for estimating health levels, our medical diagnosis system can only follow a series of nodes in a decision tree. In fact, a decision tree has many points of division, so after a certain division of the tree, the system cannot refer to rule sets on the other decision tree clusters.

To overcome these problems, in [Abe et al., 2007a], we proposed integrated data mining that categorizes the medical data into multiple categories, to discover relationships between the items in each category and the health levels, and integrates the results. In addition, we discovered an order of influential power of each category which controls the results. In the followings, we show the actual results of categorized data mining.

If we apply a rule generated from all the data, to person ID1035, for instance, the estimated health level is 4b, though the assigned health level is 4c. However, if we apply a rule generated from the liver, pancreas, and kidney test data, the estimated health level becomes 4c. The reason is that if we apply a rule generated from all the data, the effect of LDH (Lactate dehydrogenase) cannot be considered (LDH does not appear in the decision tree.), but if we apply a rule generated from the liver, pancreas, and kidney test data, the effect of LDH can be included during computational medical diagnosis. Since LDH is categorized into the liver, pancreas, and kidney test data, if we generate a rule only from these data, any influence from tumour markers³ will be cancelled and rules including effect of LDH can appear. In fact, we can obtain a part of a decision tree as shown below;

³ In [Abe et al., 2007a], we discovered that tumour marker is the most influential factor.

```

...
LDH > 241
| ALP type3 <= 52.4
| | Protein fractionation:  $\alpha$  2 <= 9.7
| | | Lipase <= 5
| | | | Blood urea nitrogen <= 12 : 4b
| | | | Blood urea nitrogen > 12 : 5b
| | | Lipase > 5
| | | | AST(GOT) <= 20 : 4b(1/2/0)
| | | | AST(GOT) > 20
.....

```

hus for these type of problems, a combination of multiple rules will work well. For person ID1035, even a single rule set that is generated from data of a single category can work well. Of course, in general, we cannot estimate the health level correctly by only using a rule set generated by categorized data mining, we cannot estimate health level correctly. In fact, correct estimation ratios are fewer when obtained in this manner than those obtained by rules generated from all the data. On the contrary, for NSE (neuron-specific enolase) which is categorized in tumour markers, as pointed out in section 4, since an effect of NSE cannot be considered, an estimated health level becomes 3, though the assigned health level is 4b. It might be better to apply a similar strategy to the above, but since NSE is categorized as a tumour marker, we cannot apply a similar strategy. Of course the generated model itself might not sufficiently proper, but it is necessary to introduce the other strategy to overcome such problems.

In fact, we categorized the data sets according to the standard classification including 1) liver, pancreas, and kidney test data, 2) metabolic function test data, 3) general urine test data, 4) blood and immunity test data, and 5) tumor markers. For the case of ID1035, categorized rule generation and application worked well. However, if we take the case of NSE into consideration, it might be necessary to introduce another classification or a more complicated or detailed classification. Zheng proposed committee learning [Zheng and Webb, 1998] which divide data set into several parts and perform data mining for each divided data set and generates a result after comparison of each result. The classification strategy is different from categorized data mining, but if we can collect enough medical data, it will be better to introduce committee learning. In fact, although we have only 1800 medical data, we apply committee learning to the medical data to obtain better results. Anyhow, we need to generate rules with properly categorized data sets and apply generated rules with a proper combination. Then we will be able to deal with complex or mixed data.

5.2 Health level estimation according to the patient's situation

When physicians assign a health level to a patient, they will focus on a part of the medical data according to the patients situation or clinical interview. Thus they do not take all the data into account. Since some of the data are not related to the patients situation, physicians usually ignore unnecessary data. For

a better estimation of health levels, it might be necessary to prepare or install such an intuitional reasoning as physicians do. That is, during the health level estimation procedure, the system should focus on proper data clusters, and apply rules related to the data clusters.

As shown above, we conducted categorized data mining and applied the generated rule sets to determine health levels. Currently we have not discovered general models for generation and applying rules. However, as shown above, we discovered several case that can estimate health levels correctly. Therefore, it is necessary to develop an automatic categorization method that can properly categorize medical data. Simple Principal Component Analysis could not properly categorize the medical data sets. In fact, if we know the patient's situation, we can focus on the data category relating to the patient's organ which is the source of the health problem. In the future, we will construct a data categorization model and a rule set combination model by analyzing data sets and physicians' health level determination models.

Finally, we can also say that the combination of categorized learning and committee learning will generate better result in data mining.

6 Conclusions

In this paper, we discussed the treatment of complex medical data which are collected in the International Research and Educational Institute for Integrated Medical Science (IREIIMS) project. Our main aim is to determine relationships between health levels and medical data. By applying C4.5, we could obtained acceptable results, but for even better results, we suggested and introduced several strategies including categorized data mining and combined rule application. We have not discovered general models for categorization and combination. However, we point out that a general model can be obtained by referring to physicians' determination patterns. We need to discover more strong relationships between medical data and health status.

For relationship or association, Agrawal proposed an association rule that represents relationships between items in databases [Agrawal et al., 1993]. The association rule is frequently used when analyzing POS data to discover tendencies of users' shopping patterns (basket analysis). However, from the analysis, we can only discover frequently co-occurring patterns. Also, relational data mining has recently been proposed [Džroski and Lavrač, 2001]. This paradigm also discovers relationships between items in a (relational) database by using ILP techniques. Their approaches are important for complex data mining. However our major aim is not to discover relationships between each category but to determine an effective classification for data mining of complex data. Nevertheless, their concept can be introduced to discover relationships.

Finally, we emphasize that our approach is based on a concept of chance discovery [Ohsawa and McBurney, 2003]. A rare relationship that cannot be extracted by simple C4.5 application can play a significant role in a proper health

level determination. Accordingly, our main aim is to discover such rare and significant relationships that can be used for accurate health level determination.

Acknowledgments

This research was supported in part by the Program for Promoting the Establishment of Strategic Research Centers, Special Coordination Funds for Promoting Science and Technology, Ministry of Education, Culture, Sports, Science and Technology (Japan). We thank Mr. Ken Chang (NTT-AT) for supporting to develop analysis tools.

References

- [Abe et al., 2004] Abe A., Naya F., Kogure K., and Hagita N.: Rule Acquisition from small and heterogeneous data set, *Technical Report of JSAI, SIG-KBS-A304-32*, pp. 189–194 (2004) in Japanese
- [Abe et al., 2007a] Abe A., Hagita N., Furutani M., Furutani Y., and Matsuoka R.: Possibility of Integrated Data Mining of Clinical Data, *Data Science Journal*, Vol. 6, Supplement, pp. S104–S115 (2007)
- [Abe et al., 2007b] Abe A., Hagita N., Furutani M., Furutani Y., and Matsuoka R.: An interface for medical diagnosis support, *Proc. of KES2007* (2007) to appear
- [Agrawal et al., 1993] Agrawal R., Imielinski T., and Swami A.: Mining association rules between sets of items in large databases, *Proc. of ACM SIGMOD Int'l Conf. on Management of Data*, pp. 207–216 (1993)
- [Džroski and Lavrač, 2001] Džroski S and Lavrač N eds.: *Relational Data Mining*, Springer Verlag (2001)
- [Ichise and Numao, 2003] Ichise R., Numao M.: A Graph-based Approach for Temporal Relationship Mining, *Technical Report of JSAI, SIG-FAI-A301*, pp. 121–126 (2003)
- [Ichise and Numao, 2005] Ichise R. and Numao M.: First-Order Rule Mining by Using Graphs Created from Temporal Medical Data, *LNAI*, Vol. 3430, pp. 112–125 (2005)
- [Kobayashi and Kawakubo, 1994] Kobayashi T. and Kawakubo T.: Prospective Investigation of Tumor Markers and Risk Assessment in Early Cancer Screening, *Cancer*, Vol. 73, No. 7, pp. 1946–1953 (1994)
- [Ohsawa, 2003] Ohsawa Y., Okazaki N., and Matsumura N.: A Scenario Development on Hepatics B and C, *Technical Report of JSAI, SIG-KBS-A301*, pp. 177–182 (2003)
- [Ohsawa and McBurney, 2003] Osawa Y. and McBurney P. eds.: *Chance Discovery*, Springer Verlag (2003)
- [Tsumoto, 2004] Tsumoto S.: Mining Diagnostic Rules from Clinical Databases Using Rough Sets and Medical Diagnostic Model, *Information Sciences*, Vol. 162, No. 2, pp. 65–80 (2004)
- [Quinlan, 1993] Quinlan J.R.: *C4.5: Programs for Machine Learning*, Morgan Kaufman (1993)
- [Zheng and Webb, 1998] Zheng Z. and Webb G.I.: Stochastic Attribute Selection Committees, *Proc. of AI98*, pp. 321–332 (1998)

POM Centric Multiaspect Data Analysis for Investigating Human Problem Solving Function

Shinichi Motomura¹, Akinori Hara¹, Ning Zhong^{2,3}, and Shengfu Lu³

¹ Graduate School, Maebashi Institute of Technology, Japan

² Department of Life Science and Informatics, Maebashi Institute of Technology, Japan

³ The International WIC Institute, Beijing University of Technology, China
motomura@maebashi-it.org

Abstract. In the paper, we propose an approach of POM (peculiarity oriented mining) centric multiaspect data analysis for investigating human problem solving related functions, in which computation tasks are used as an example. The proposed approach is based on Brain Informatics (BI) methodology, which supports studies of human information processing mechanism systematically from both macro and micro points of view by combining experimental cognitive neuroscience with advanced information technology. We describe how to design systematically cognitive experiments to obtain multi-ERP data and analyze spatiotemporal peculiarity of such data. Preliminary results show the usefulness of our approach.

1 Introduction

Problem-solving is one of main capabilities of human intelligence and has been studied in both cognitive science and AI [9], where it is addressed in conjunction with reasoning centric cognitive functions such as attention, control, memory, language, reasoning, learning, and so on. We need to better understand how human being does complex adaptive, distributed problem solving and reasoning, as well as how intelligence evolves for individuals and societies, over time and place [3, 11–13, 17]. Then, we catch problem solving from the standpoint of Brain Informatics, and address systematically for the solution of a process.

Brain Informatics (BI) is a new interdisciplinary field to study human information processing mechanism systematically from both macro and micro points of view by cooperatively using experimental, theoretical, cognitive neuroscience and advanced information technology [16, 17]. It attempts to understand human intelligence in depth, towards a holistic view at a long-term, global vision to understand the principles, models and mechanisms of human information processing system.

Our purpose is to understand activities of human problem solving system by investigating the spatiotemporal features and flow of human problem solving system, based on functional relationships between activated areas of human brain. More specifically, at the current stage, we want to understand:

- how a peculiar part (one or more areas) of the brain operates in a specific time;
- how the operated part changes along with time;
- how the activated areas work cooperatively to implement a whole problem solving system;
- how the activated areas are linked, indexed, navigated functionally, and what are individual differences in performance.

Based on this point of view, we propose a way of peculiarity oriented mining (POM) for knowledge discovery in multiple human brain data.

The rest of the paper is organized as follows. Section 2 provides a BI Methodology for multiaspect human brain data analysis of human problem solving system. Sections 3 explain how to design the experiment of an ERP mental arithmetic task with visual stimuli standing on BI Methodology. Sections 4 describe how to do multiaspect analysis in the obtained ERP data, respectively, as an example to investigate human problem solving and to show the usefulness of the proposed mining process. Finally, Section 5 gives concluding remarks.

2 Brain Informatics Methodology

Brain informatics pursues a holistic understanding of human intelligence through a systematic approach to brain research. BI regarded the human brain as an information processing system (HIPS) and emphasizes cognitive experiments to understand its mechanisms for analyzing and managing data. Such systematic study includes the following 4 main research issues:

- systematic investigation of human thinking centric mechanisms;
- systematic design of cognitive experiments;
- systematic human brain data management;
- systematic human brain data analysis.

The first issue is based on the observation for Web intelligence research needs and the state-of-the-art cognitive neuroscience. In cognitive neuroscience, although many advanced results with respect to “perception oriented” study have been obtained, only a few of preliminary, separated studies with respect to “thinking oriented” and/or a more whole information process have been reported [1].

The second issue is with respect to how to design the psychological and physiological experiments for obtaining various data from HIPS, in a systematic way. In other words, by systematic design of cognitive experiments in BI methodology, the data obtained from a cognitive experiment and/or a set of cognitive experiments may be used for multi-task/purpose.

The third issue relates to manage human brain data, which is based on a conceptual model of cognitive functions that represents functional relationships among multiple human brain data sources for systematic investigation and understanding of human intelligence.

The last issue is concerned with how to extract significant features from multiple brain data measured by using fMRI and EEG in preparation for multispect data analysis by combining various data mining methods with reasoning [4, 6, 10, 11, 15].

An investigation flow based on BI methodology is shown in Figure 1, in which various tools can be cooperatively used in the multi-step process for experimental design, pre-processing (data extraction, modeling and transformation), multispect data mining and post-processing.

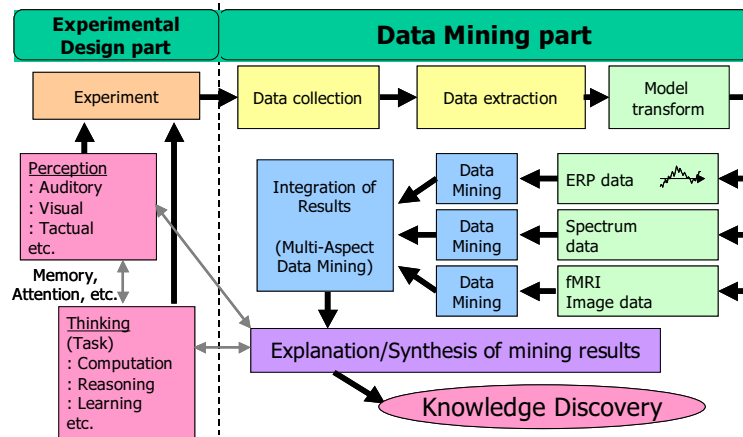


Fig. 1. A flow based on BI methodology

3 The Experiment of Mental Arithmetic Task with Visual Stimuli

As mentioned above, based on BI methodology, the data obtained from a cognitive experiment and/or a set of cognitive experiments may be used for multi-task/purpose, including for investigating both lower and higher functions of HIPS. For example, it is possible that our experiment can meet the following requirements: investigating the mechanisms of human visual and auditory systems, computation, problem-solving (i.e. the computation process is regarded as an example of problem-solving process), and the spatiotemporal feature and flow of HIPS in general. Figure 2 gives a computation process from the macro viewpoints, with respect to several component functions of human solving problem, such as attention, interpretation, short-term memory, understanding of work, computation, checking.

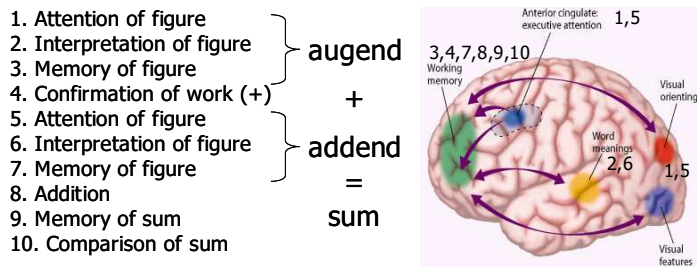


Fig. 2. Computation as an example of problem-solving

In this work, the ERP (event-related potential) human brain waves are derived by carrying out a mental arithmetic task with visual stimuli, as an example to investigate human problem solving process. ERP is a light, sound, and brain potential produced with respect to the specific phenomenon of spontaneous movement [2].

3.1 Outline of Experiments

The experiment conducted this time shows a numerical calculation problem to a subject, and asks the subject to solve it in mental arithmetic, and the shown sum has hit, or it pushes a button, and performs a judging of corrigenda. The form of the numerical calculation to be shown is the addition problem of “augend + addend = sum”. The wrong sum occurs at half the probability, and the distribution is not uniform.

In the experiments, three states (tasks), namely, *visual on-task*, *visual off-task*, and *no-task*, exist by the difference in the stimulus given to a human subject. *Visual on-task* is the state which is calculating by looking a number. *Visual off-task* is the state which is looking the number that appears at random. *no-task* is the relaxed state which does not work at all.

Figure 3 gives an example of the screen state transition. Type 1 is two digits addition that the figure remains on the screen. Type 2 is also two digits addition, but the figure doesn’t remain on the screen.

We try to compare and analyze what is the relationship between tasks. Figure 4 gives an example of comparing tasks. By this design, it is possible to analyze the influence, in the different levels of difficulty, with the same *on-task* and *off-task*, and to make a comparison between *on-task* and *off-task* in the same difficulty.

3.2 Trigger Signal and Timing Chart

It is necessary to measure EEG relevant to a certain event to the regular timing in measurement of ERP repeatedly. In this research, since the attention was paid

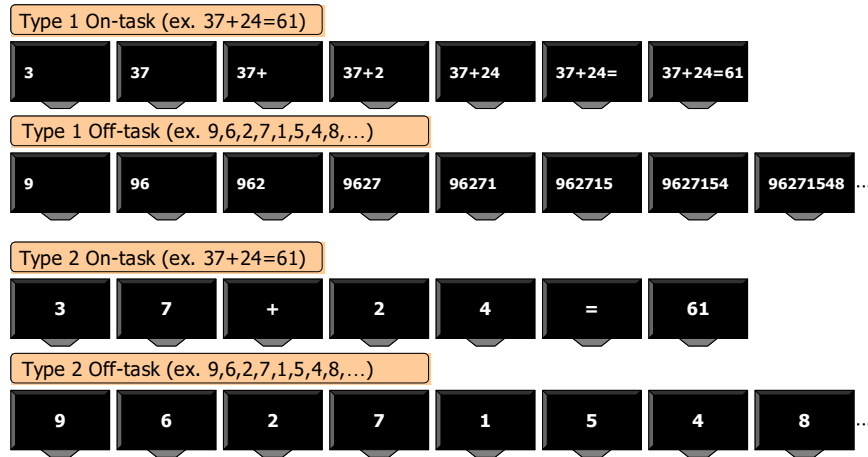


Fig. 3. Experimental design with two levels of difficulty

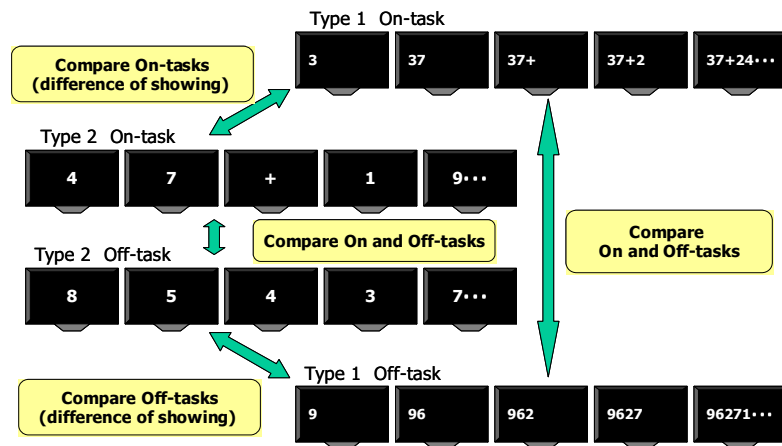


Fig. 4. Viewpoints for experiments

to each event of augend, addend, and sum presentation in calculation activities. Pre-trigger was set to 200 [msec], and addition between two digits are recorded in 10000 [msec], respectively. Figure 5 gives an example of the time chart. “au” is augend, “ad” is addend, and “su” is sum. Therefore “au2” is MSD (last 2-digits) of augend, and “au1” is LSD (last 1-digits) of augend. “n” is the random number (1-digits).

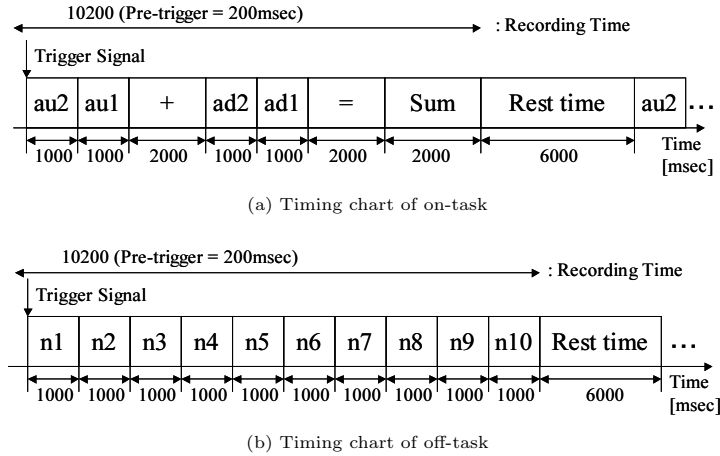


Fig. 5. Timing chart

3.3 Experimental Device and Measurement Conditions

Electroencephalographic activity was recorded using a 64 channel BrainAmp amplifier (Brain Products, Munich, Germany) with a 64 electrode cap. The electrode cap is based on an extended international 10-20 system. Furthermore, eye movement measurement (2ch) is also used. The sampling frequency is 2500Hz to be processed. The number of experimental subjects is 20.

4 Multispect Data Analysis (MDA)

4.1 Topography Analysis

For the measured EEG data, a maximum of 40 addition average processing were performed, and the ERPs were derived by using Brain Vision Analyzer (Brain Products, Munich, Germany). Generally speaking, the Wernicke area of a left temporal lobe and the prefrontal area are related to the calculation

process [5]. In this study, we pay attention to recognition of the number, short-term memory and attentiveness, as well as compare Type 1 and Type 2 by focusing on important channels (C5).

Figure 6 shows the topography. After displayed symbol, the display time is 250 milliseconds. We can see some difference between Type 1 and Type 2 at the appearance part (the left and right brain) of the positive potential. This phenomenon means the process of recognition of number and expression are different between Type 1 and Type 2.

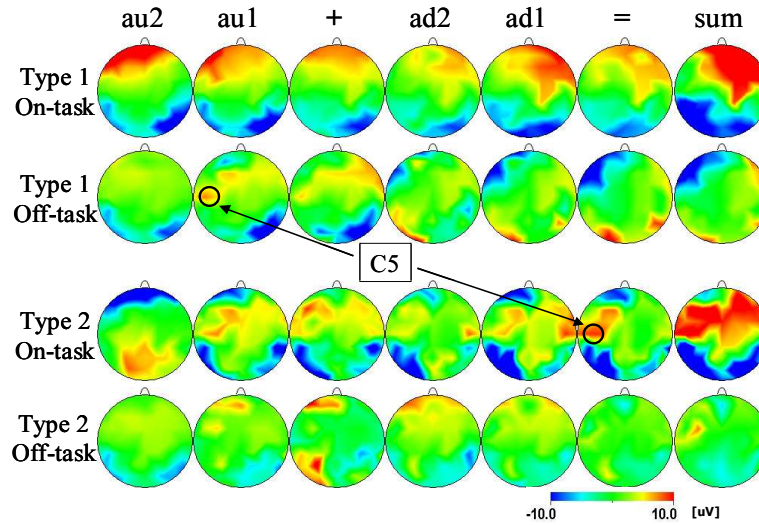


Fig. 6. Topography of ERP data

First, we discuss the on-task and off-task. We can see some difference between on-task and off-task at the positive potential. It is guessed that on-task is linked to the activating of frontal area. Next, we discuss the frontal area. We can see some difference between Type 1 and Type 2 at the distribution of the positive potential. Type 1 shows high positive potential in almost time. It is guessed that Type 1 is easier than Type 2 to recognize the number, then the experimental subject repeated computation in Type 1. Finally, we discuss the channel C5, which is part of the left temporal lobe with respect to the logical interpretation. Generally speaking, when human interprets and calculates the expression, the left brain is activated. In particular, the area with respect to short-term memory is strongly activated for Type 2. In addition, the experimental subject logically interprets an each figure. Therefore, it is guessed that the potential distribution of Type 2 is different from the one of Type 1 in C5. It is necessary to investigate

this phenomenon deeply additionally with the change of the time series around the visual area.

4.2 Peculiarity Oriented Mining

It is clear that a specific part of the brain operates in a specific time and the operations change over time. Although detecting the concavity and convexity (P300 etc.) of ERP data is easy by using the existing tool, it is difficult to find a peculiar one in multiple channels with the concavity and convexity [7, 8]. In order to discover new knowledge and models of human information processing activities, it is necessary to pay attention to the peculiar channel and time in ERPs for investigating the spatiotemporal features and flow of a human information processing system.

peculiarity oriented mining (POM) is a proposed knowledge discovery methodology [14, 15]. The main task of POM is the identification of peculiar data. The attribute-oriented method of POM, which analyze data from a new view and are different from traditional statistical methods, has been recently proposed by Zhong *et al.* and applied in various real-world problems [14, 15]. Unfortunately, such POM is not totally fit for ERP data analysis. The reason is that the useful aspect for ERP data analysis is not amplitude, but the latent time. In order to solve this problem, we extend POM to Peculiarity Vector Oriented Mining (PVOM). After smoothing enough by moving average processing, in the time series, we pay the attention to each potential towards N pole or P pole. Furthermore, the channel with the direction different from a lot of channels is considered to be a peculiar channel at that time. Hence, the distance between the attribute-values is expressed at the angle. And this angle can be obtained from the inner product and the norm in the vector. Let inclination of wave i in a certain time t be x_{it} . The extended PF (Peculiarity Factor) corresponding to ERP can be defined by the following Eq. (1).

$$PF(x_{it}) = \sum_{k=1}^n \theta(x_{it}, x_{kt})^\alpha. \quad (1)$$

$\alpha = 0.5$ as default. In normally POM, PF is obtained by distance between two attribute values. However, θ in Eq. (1) is an angle which the wave in time t makes. For the θ , we can compute for an angle using Eq. (2).

$$\cos\theta = \frac{1 + x_{it} \cdot x_{kt}}{\sqrt{1 + x_{it}^2} \sqrt{1 + x_{kt}^2}}. \quad (2)$$

Based on the peculiarity factor, the selection of peculiar data is simply carried out by using a threshold value. More specifically, an attribute value is peculiar if its peculiarity factor is above minimum peculiarity p , namely, $PF(x_{it}) \geq p$. The threshold value p may be computed by the distribution of PF as follows:

$$\begin{aligned} \text{threshold} = & \text{mean of } PF(x_{it}) + \\ & \beta \times \text{standard deviation of } PF(x_{it}) \end{aligned} \quad (3)$$

where β can be adjusted by a user, and $\beta = 1$ is used as default. The threshold indicates that a data is a peculiar one if its PF value is much larger than the mean of the PF set. In other words, if $PF(x_{it})$ is over the threshold value, x_{it} is a peculiar data. By adjusting the parameter β , a user can control and adjust the threshold value.

4.3 Application of the Extended POM Method

In this work, we want to mine four kinds of patterns, which are classified into two types of peculiarity with respect to the temporal and channel axes, respectively, as shown in Figure 7. Mining 1 and Mining 2 are used to find temporal peculiarity, and Mining 3 and Mining 4 are used to find channel peculiarity, respectively.

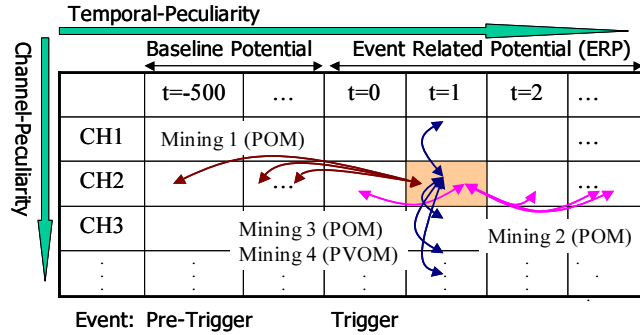


Fig. 7. Views of ERP peculiarity

More specifically, Mining 1 examines whether the potential at arbitrary time is peculiar compared with a baseline potential in channel C5. Mining 2 examines whether the potential at arbitrary time is peculiar compared with an ERP in channel C5. Mining 3 examines whether the potential of C5 is peculiar compared with the potential on other channels in a specific time. Furthermore, Mining 4 examines whether a potential change of C5 is peculiar compared with a potential change on other channels in a specific time. As shown in Figure 7, the POM method is used for the Mining 1 to Mining 3 and the extended POM method (PVOM) stated in Section 4.2 is used for the Mining 4. PF axis 0 in Figs. 8 and 9 denotes the threshold and the part is peculiar if it is over the threshold value.

Figure 8 shows an ERP and the result of mining on it, in which the peculiarity in ERP data with respect to addition Type 1 (Channel: C5, Task: off-task) is presented. We can see there are some higher values of peculiarity and the potential changes over the time. Although the off-task potential is about zero, Viewpoint 1 is nevertheless judged to be peculiar because it is a higher positive

at this time. Furthermore, Viewpoint 2 represents that the potential of C5 is peculiar at this time in the channel. Viewpoint 3 represents that a potential change of C5 is peculiar at this time in the channel.

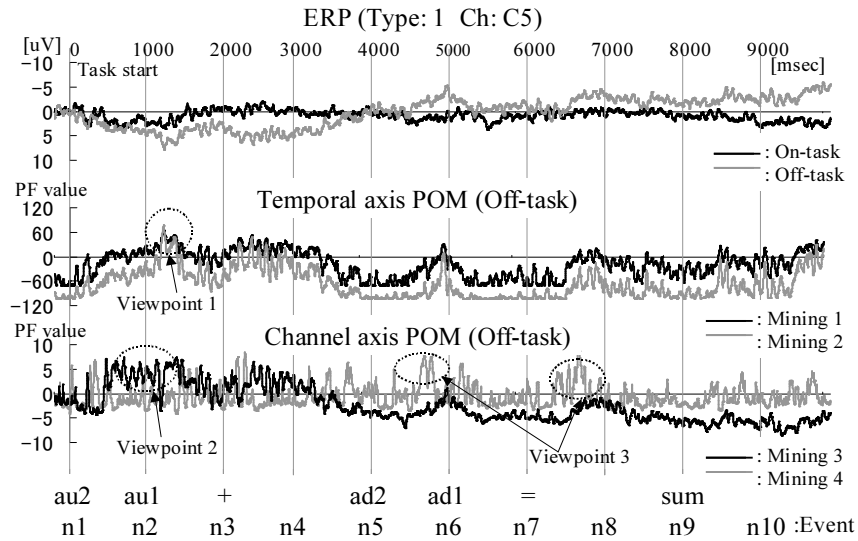


Fig. 8. ERP and mining results (Type 1 off-task)

Figure 9 shows an ERP and the result of mining on it, in which the peculiarity in ERP data with respect to addition Type 2 (Channel: C5, Task: on-task) is presented. Although the on-task pencil is a higher positive one, Viewpoint 4 is judged to be peculiar because it is near to zero. Furthermore, although other on-task potentials are high, Viewpoint 5 is judged to be peculiar because it is remarkably high in comparison with others.

Figure 10 illustrates how to integrate the mining results, not only POM for ERP data, but also frequency data and fMRI data. The system collects multiple data sources from several event-related time points, and transformed into various forms for POM centric MDA. Furthermore, the results of separate analysis can be explained and combined into a whole flow of information processing with respect to problem solving.

5 Conclusion

In this paper, we investigated human problem solving related functions by using computation as an example, which demonstrate what is BI methodology and

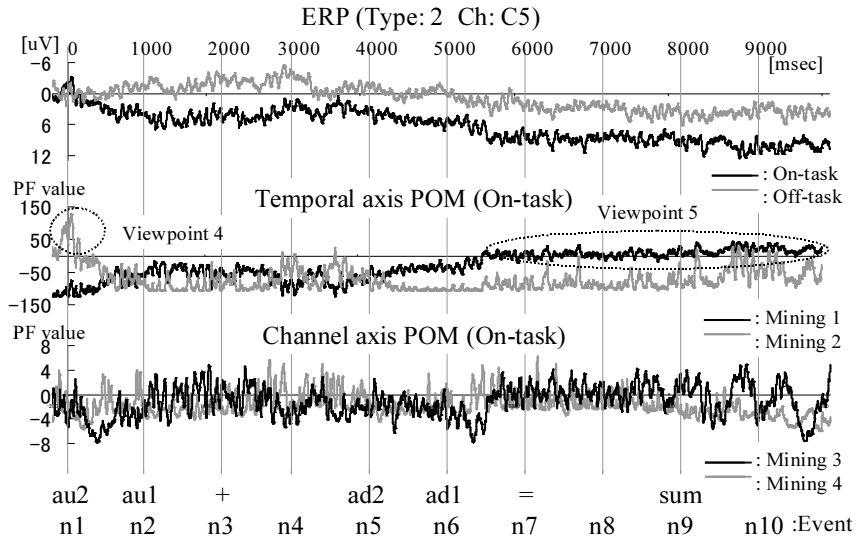


Fig. 9. ERP and mining results (Type 2 on-task)

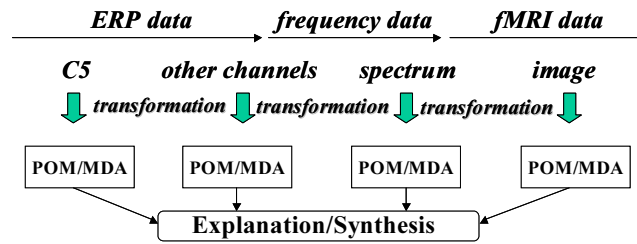


Fig. 10. Integration of mining results

its usefulness. The proposed POM centric multiaspect ERP data analysis based on BI methodology shifts the focus of cognitive science from a single type of experimental data analysis towards a deep, holistic understanding of human information processing principles, models and mechanisms.

Our future work includes obtaining and analyzing more subject data, combining with fMRI human brain image data for multi-aspect analysis in various approaches of data mining and reasoning.

References

1. M.S. Gazzaniga (ed.) *The Cognitive Neurosciences III*, The MIT Press (2004).
2. T.C. Handy, *Event-Related Potentials, A Methods Handbook*, The MIT Press (2004).
3. J. Liu, X. Jin, and K.C. Tsui, *Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling*, Springer (2005).
4. V. Megalooikonomou and E.H. Herskovits, "Mining Structure-Function Associations in a Brain Image Database", K.J. Cios (ed.) *Medical Data Mining and Knowledge Discovery*, Physica-Verlag (2001) 153-179.
5. H. Mizuhara, L. Wang, K. Kobayashi, Y. Yamaguchi, "Long-range EEG Phase-synchronization During an Arithmetic Task Indexes a Coherent Cortical Network Simultaneously Measured by fMRI", *NeuroImage*, Vol.27, No.3 (2005) 553-563.
6. T.M. Mitchell, R. Hutchinson, R.S. Niculescu, F.Pereira, X. Wang, M. Just, and S. Newman, "Learning to Decode Cognitive States from Brain Images", *Machine Learning*, 57(1-2) (2004) 145-175.
7. H. Nittono, Y. Nageishi, Y. Nakajima, and P. Ullsperger, "Event-related Potential Correlates of Individual Differences in Working Memory Capacity", *Psychophysiology*, 36 (1999) 745-754.
8. T.W. Picton, S. Bentin, P. Berg, E. Donchin, S.A. Hillyard, R. Johnson, et al. "Guidelines for using human event-related potentials to study cognition: Recording standards and publication criteria", *Psychophysiology*, 37, (2000) 127-152.
9. A. Newell and H.A. Simon, *Human Problem Solving*, Prentice-Hall (1972).
10. F.T. Sommer and A. Wichert (eds.) *Exploratory Analysis and Data Modeling in Functional Neuroimaging*, The MIT Press (2003).
11. R.J. Sternberg, J. Lautrey, and T.I. Lubart, *Models of Intelligence*, American Psychological Association (2003).
12. Y.Y. Yao, "A Partition Model of Granular Computing", *Springer LNCS Transactions on Rough Sets*, 1 (2004) 232-253.
13. L.A. Zadeh, "Precisiated Natural Language (PNL)", *AI Magazine*, 25(3) (Fall 2004) 74-91.
14. N. Zhong, Y.Y. Yao, and M. Ohshima, "Peculiarity Oriented Multi-Database Mining", *IEEE Transaction on Knowledge and Data Engineering*, 15(4) (2003) 952-960.
15. N. Zhong, J.L. Wu, A. Nakamaru, M. Ohshima, H. Mizuhara, "Peculiarity Oriented fMRI Brain Data Analysis for Studying Human Multi-Perception Mechanism", *Cognitive Systems Research*, 5(3), Elsevier (2004) 241-256.
16. N. Zhong, "Building a Brain-Informatics Portal on the Wisdom Web with a Multi-Layer Grid: A New Challenge for Web Intelligence Research", V. Torra et al. (eds.) *Modeling Decisions for Artificial Intelligence*, LNAI 3558, Springer (2005) 24-35.
17. N. Zhong, "Impending Brain Informatics (BI) Research from Web Intelligence (WI) Perspective", *International Journal of Information Technology and Decision Making*, World Scientific, Vol. 5, No. 4 (2006) 713-727.

Author Index

- Abe, Akinori, 209
Abe, Hidenao, 49
- Bahri, Emna, 151
Basile, T.M.A., 13
Bathoorn, Ronnie, 25
Biba, M., 13
- Ceci, Michelangelo, 83
Ciesielski, Krzysztof, 128
- d'Amato, Claudia, 197
Delteil, Alexandre, 106
Dembczyński, Krzysztof, 163
Di Mauro, N., 13
- El Sayed, Ahmad, 185
Elazmeh, William, 37
Esposito, Floriana, 13, 197
- Fanizzi, Nicola, 197
Farion, Ken, 37
Ferilli, S., 13
Furutani, Michiko, 209
Furutani, Yoshiyuki, 209
- Grcar, Miha, 1
Grobelnik, Marko, 1
- Hacid, Hakim, 185
Hagita, Norihiro, 209
Hara, Akinori, 221
Hirabayashi, Satoru, 49
Hirano, Shoji, 139
- Kolczyńska, Elżbieta, 175
Kontkiewicz, Aleksandra, 106
Kotłowski, Wojciech, 163
Kryszkiewicz, Marzena, 106
Kłopotek, Mieczysław A., 128
- Lu, Shengfu, 221
- Maddouri, Mondher, 151
Malerba, Donato, 83
Marcinkowska, Katarzyna, 106
Matsuoka, Rumiko, 209
Matwin, Stan, 37
Michalowski, Wojtek, 37
Mladenic, Dunja, 1
Motomura, Shinichi, 221
- Nicoloyannis, Nicolas, 151
- O'Sullivan, Dympna, 37
Ohsaki, Miho, 49
Okawa, Takenao, 71
Ozaki, Tomonobu, 71
- Peters, James F., 116
Protaziuk, Grzegorz, 106
- Raś, Zbigniew W., 59, 95
Ramanna, Sheela, 116
Rybinski, Henryk, 106
- Sehatkar, Morvarid, 37
Siebes, Arno, 25
Słowiński, Roman, 163
- Tsumoto, Shusaku, 139
- Wieczorkowska, Alicja, 175
Wierzchoń, Sławomir, 128
Wilk, Szymon, 37
Wyrzykowska, Elżbieta, 95
- Yamaguchi, Takahira, 49
Yamamoto, Tsubasa, 71
- Zhang, Xin, 59
Zhong, Ning, 221
Zighed, Djamel, 185