Workshop Notes

International Workshop

"What can FCA do for Artificial Intelligence?"

FCA4AI

International Joint Conference on Artificial Intelligence

IJCAI 2015

July 25, 2015

Buenos Aires, Argentina

Editors

Sergei O. Kuznetsov (NRU HSE Moscow)

Amedeo Napoli (LORIA Nancy)

Sebastian Rudolph (TU Dresden)

http://fca4ai.hse.ru/2015/

Preface

The three preceding editions of the FCA4AI Workshop showed that many researchers working in Artificial Intelligence are deeply interested by a well-founded method for classification and mining such as Formal Concept Analysis (see `http://www.fca4ai.hse.ru/`). The first edition of FCA4AI was co-located with ECAI 2012 in Montpellier and published as `http://ceur-ws.org/Vol-939/`, the second edition was co-located with IJCAI 2013 in Beijing and published as `http://ceur-ws.org/Vol-1058/`, and finally the third edition was co-located with ECAI 2014 in Prague and published as `http://ceur-ws.org/Vol-1257/`. Based on that, we decided to continue the series and we took the chance to organize a new edition of the workshop in Buenos Aires at the IJCAI 2015 Conference. This year, the workshop has again attracted many different researchers working on actual and important topics, e.g. recommendation, linked data, classification, biclustering, pattern mining, ontology design, and various applications. This shows the diversity and the richness of the relations between FCA and AI. Moreover, this is a good sign for the future and especially for young researchers that are at the moment working in this area or who will do.

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis and classification. FCA allows one to build a concept lattice and a system of dependencies (implications) which can be used for many AI needs, e.g. knowledge discovery, learning, knowledge representation, reasoning, ontology engineering, as well as information retrieval and text processing. As we can see, there are many "natural links" between FCA and AI.

Recent years have been witnessing increased scientific activity around FCA, in particular a strand of work emerged that is aimed at extending the possibilities of FCA w.r.t. knowledge processing, such as work on pattern structures and relational context analysis. These extensions are aimed at allowing FCA to deal with more complex than just binary data, both from the data analysis and knowledge discovery points of view and as well from the knowledge representation point of view, including, e.g., ontology engineering.

All these investigations provide new possibilities for AI activities in the framework of FCA. Accordingly, in this workshop, we are interested in two main issues:

- How can FCA support AI activities such as knowledge processing (knowledge discovery, knowledge representation and reasoning), learning (clustering, pattern and data mining), natural language processing, and information retrieval.
- How can FCA be extended in order to help AI researchers to solve new and complex problems in their domains.

The workshop is dedicated to discuss such issues. This year, the papers submitted to the workshop were carefully peer-reviewed by three members of the program committee and 10 papers with the highest scores were selected. We thank all the PC members for their reviews and all the authors for their contributions.

The Workshop Chairs

Sergei O. Kuznetsov
National Research University, Higher Schools of Economics, Moscow, Russia

Amedeo Napoli
LORIA (CNRS – Inria Nancy Grand Est – Université de Lorraine), Vandoeuvre les Nancy, France

Sebastian Rudolph
Technische Universität Dresden, Germany

Program Committee

Mathieu D'Aquin (Open University, UK)

Gabriela Arevalo (Universidad Nacional de Quilmes, Argentina)

Jaume Baixeries, UPC Barcelona, Catalunya

Karell Bertet (Université de La Rochelle, France, Germany)

Claudio Carpineto (Fondazione Ugo Bordoni, Roma, Italy)

Florent Domenach (University of Nicosia, Cyprus)

Sébastien Ferré (IRISA, Rennes, France)

Marianne Huchard (LIRMM/Université de Montpellier, France)

Dmitry I. Ignatov (NRU Higher School of Economics, Moscow, Russia)

Mehdi Kaytoue (INSA-LIRIS Lyon, France)

Florence Le Ber, Université de Strasbourg, France

Nizar Messai (Université de Tours, France)

Rokia Missaoui (Université du Québec en Outaouais, Ottawa, Canada)

Sergei A. Obiedkov (NRU Higher School of Economics, Moscow, Russia)

Jean-Marc Petit (INSA-LIRIS Lyon, France)

Uta Priss (Ostfalia University of Applied Sciences, Wolfenbüttel, Germany)

Chedy Raïssi (Inria/LORIA Nancy, France)

Artem Revenko, Technische Universität Dresden, Germany

Christian Săcărea (Babes-Bolyai University, Cluj-Napoca, Romania)

Baris Sertkaya (SAP Dresden, Germany)

Henry Soldano (Université de Paris-Nord, France)

Laszlo Szathmary, University of Debrecen, Hungary

Petko Valtchev (Université du Québec à Montréal, Montréal, Canada)

Renato Vimiero (UFPE Recife, Brazil)

Contents

Invited Talk
Using trust networks to improve data quality and recommendations

Hernán Astudillo
Universidad Técnica Federico Santa María (UTFSM)
Avenida España 1680, Valparaíso, Chile
hernan@inf.utfsm.cl

Abstract

The boom in social computing has left us with large amounts of information, some of it from automated sensors and some from humans, most of it incomplete, inconsistent, wrong and/or stale. Doctorow noticed that people are lazy, dumb, and at times deceitful, but we still want to use their data rather than none. We will introduce the related notions of Reputation, Trust, Confidence and Reliability, and will show how they can be used to improve the quality of data and of recommendations. We will pay special attention to explicit record of trust relationships among agents (human and otherwise), illustrate its usage with some ongoing recommender system projects, and highlight recent advances in trust aging.

Bridging DBpedia Categories and DL-Concept Definitions using Formal Concept Analysis

Mehwish Alam, Aleksey Buzmakov, Victor Codocedo, Amedeo Napoli

LORIA (CNRS – Inria Nancy Grand Est – Université de Lorraine)
BP 239, Vandoeuvre-lès-Nancy, F-54506, France
{firstname.lastname@loria.fr}

Abstract. The popularization and quick growth of Linked Open Data (LOD) has led to challenging aspects regarding quality assessment and data exploration of the RDF triples that shape the LOD cloud. Particularly, we are interested in the completeness of data and its potential to provide concept definitions in terms of necessary and sufficient conditions. In this work we propose a novel technique based on Formal Concept Analysis which organizes RDF data into a concept lattice. This allows the discovery of implications, which are used to automatically detect missing information and then to complete RDF data.

Keywords: Formal Concept Analysis, Linked Open Data, Data Completion.

1 Introduction

The World Wide Web has tried to overcome the barrier of data sharing by converging data publication into Linked Open Data (LOD) [3]. The LOD cloud stores data in the form of *subject-predicate-object* triples based on the RDF language[1], a standard formalism for information description of web resources. In this context, DBpedia is the largest reservoir of linked data in the world currently containing more than 4 million triples. All of the information stored in DBpedia is obtained by parsing Wikipedia, the largest open Encyclopedia created by the collaborative effort of thousands of people with different levels of knowledge in several and diverse domains.

More specifically, DBpedia content is obtained from semi-structured sources of information in Wikipedia, namely *infoboxes* and *categories*. Infoboxes are used to standardize entries of a given type in Wikipedia. For example, the infobox for "automobile" has entries for an image depicting the car, the name of the car, the manufacturer, the engine, etc. These *attributes* are mapped by the DBpedia parser to a set of "properties" defined in an emerging ontology[2] [2] (infobox dataset) or mapped through a hand-crafted lookup table to what is called the DBPedia Ontology. Categories are another important tool in Wikipedia used to organize information. Users can freely assign a category name to an article relating it to other articles in the same category. Example of categories for cars are "Category:2010s automobiles", "Category:Sports cars" or

[1] Resource Description Framework - http://www.w3.org/RDF/
[2] Emerging in the sense of "dynamic" or "in progress".

"Category:Flagship vehicles". While we can see categories in Wikipedia as an emerging "folksonomy", the fact that they are curated and "edited" make them closer to a controlled vocabulary. DBpedia exploits the Wikipedia category system to "annotate"[3] objects using a taxonomy-like notation. Thus, it is possible to query DBpedia by using *annotations* (e.g. all cars annotated as "Sport cars"). While categorical information in DBpedia is very valuable, it is not possible to use a category as one could expect, i.e. as a definition of a class of elements that are instances of the class or, alternatively, that are "described" by the category. In this sense, such a category violates the actual spirit of semantic Web.

Let us explain this with an example. The Web site of DBpedia in its section of "Online access" contains some query examples using the SPARQL query language. The first query has the description "People who were born in Berlin before 1900" which actually translates into a graph-based search of entities of the type "Person", which have the property "birthPlace" pointing to the entity representing the "city of Berlin" and another property named "birthDate" with a value less than 1900. We can see here linked data working at "its purest", i.e. the form of the query provides the right-hand side of a definition for "People who were born in Berlin before 1900". Nevertheless, the fourth query named "French films" does not work in the same way. While we could expect also a graph-based search of objects of the type "Film" with maybe a property called "hasCountry" pointing to the entity representing "France", we have a much rougher approach. The actual SPARQL query asks for objects (of any type) annotated as "French films".

In general, categorization systems express "information needs" allowing human entities to quickly access data. French films are annotated as such because there is a need to find them by these keywords. However, for a machine agent this information need is better expressed through a *definition*, like that provided for the first query (i.e. "People who were born in Berlin before 1900"). Currently, DBPedia mixes these two paradigms of data access in an effort to profit from the structured nature of categories, nevertheless further steps have to be developed to ensure coherence and completeness in data.

Accordingly, in this work we describe an approach to bridge the gap between the current syntactic nature of categorical annotations with their semantic correspondent in the form of a concept definition. We achieve this by mining patterns derived from entities annotated by a given category, e.g. All entities annotated as "Lamborghini cars" are of "type automobile" and "manufactured by Lamborghini", or all entities annotated as "French films" are of "type film" and of "French nationality". We describe how these category-pattern equivalences can be described as "definitions" according to *implication rules* among attributes which can be mined using Formal Concept Analysis (FCA [7]). The method considers the analysis of heterogeneous complex data (not necessarily binary data) through the use of "pattern structures" [6], which is an extension of FCA able to process complex data descriptions. A concept lattice can be built from the data and then used for discovering *implication rules* (i.e. association rules whose confidence is 100%) which provide a basis for "subject definition" in terms of necessary and sufficient conditions. For more details read the complete version of this paper [1].

[3] Notice that in DBPedia the property used to link entities and categories is called "subject". We use "annotation" instead of "subject" to avoid confusions with the "subject" in an RDF triple.

This article is structured as follows: Section 2 gives a brief introduction to the theoretical background necessary to sustain the rest of the paper. Section 3 describes the approach used for data completion in the DBpedia knowledge base. Finally, Section 4 concludes the paper.

2 Preliminaries

Linked Open Data (LOD) [3] is a formalism for publishing structured data on-line using the resource description framework (RDF). RDF stores data in the form of statements represented as $\langle subject, predicate, object \rangle$. The profile of an RDF triple $\langle s, p, o \rangle$ is given by $(U \cup B) \times (U \cup B) \times (U \cup B \cup L)$ where a set of RDF triples is an RDF graph, denoted by $\mathcal{G}$. Here, U denotes a set of URI references, B refers to the blank node and L to literals. For the sake of simplicity, in the current study we do no take into account blank nodes (B). An RDF triple is represented as $U \times U \times (U \cup L)$. For convenience, in the following we denote the set of predicate names as P and the set of object names as O. LOD can then be queried and accessed through SPARQL[4], which is a standard query language for RDF data. SPARQL is based on matching graph patterns (present in the *WHERE* clause of a query) against RDF graphs. For example, let us consider the SPARQL query given in Listing 1.1, for all the entities of type Automobile manufactured by *Lamborghini*, annotated as "Sport_cars" and as "Lamborghini_vehicles",

```
SELECT ?s WHERE {
?s dc:subject dbpc:Sports_cars .
?s dc:subject dbpc:Lamborghini_vehicles .
?s rdf:type dbo:Automobile .
?s dbo:manufacturer dbp:Lamborghini }
```

Listing 1.1: SPARQL for the formal context in Figure 1. Prefixes are defined in Table 1.

Formal Concept Analysis (FCA) is a mathematical framework introduced in [7], but in the following we assume that the reader already has necessary background of FCA. We only explain it with the help of an example. For example, consider the formal context in Figure 1 where $G = U$, $M = (P \times O)$ and $(u, (p, o)) \in I \iff \langle u, p, o \rangle \in \mathcal{G}$, i.e. $\langle u, p, o \rangle$ is a triple built from different triples manually extracted from DBpedia about nine different Lamborghini cars (35 RDF triples in total). Given a subject-predicate-object triple, the formal context contains subjects in rows, the pairs predicate-object in columns and a cross in the cell where the triple subject in row and predicate-object in column exists. Figure 1 depicts the concept lattice in reduced notation calculated for this formal context and contains 12 formal concepts. Consider the first five cars (*subjects*) in the table for which the maximal set of attributes they share is given by the first four *predicate-object* pairs. Actually, they form a formal concept depicted by the gray cells in Figure 1 and labelled as "Islero, 400GT" in Figure 1 (actually, the extent of this concept is "Islero, 400GT, 350GT, Reventon"). Given a concept lattice, rules can be extracted from the intents of concepts which are comparable.

[4] http://www.w3.org/TR/rdf-sparql-query/

Predicates		Objects	
Index	URI	Index	URI
A	dc:subject	a	dbpc:Sport_Cars
		b	dbpc:Lamborghini_vehicles
B	dbp:manufacturer	c	dbp:Lamborghini
C	rdf:type	d	dbo:Automobile
D	dbp:assembly	e	dbp:Italy
E	dbo:layout	f	dbp:Four-wheel_drive
		g	dbp:Front-engine

Namespaces:	
dc:	`http://purl.org/dc/terms/`
dbo:	`http://dbpedia.org/ontology/`
rdf:	`http://www.w3.org/1999/02/22-rdf-syntax-ns\#`
dbp:	`http://dbpedia.org/resource/`
dbpc:	`http://dbpedia.org/resource/Category:`

Table 1: Index of pairs predicate-object and namespaces.

	A		B	C	D		E
	a	b	c	d	e	f	g
Reventon	×	×	×	×	×	×	
Countach	×	×	×	×	×		
350GT	×	×	×	×			×
400GT	×	×	×	×			
Islero	×	×	×	×			
Veneno	×	×					
Aventador Roadster	×	×					
Estoque			×	×		×	×
Gallardo			×	×	×		

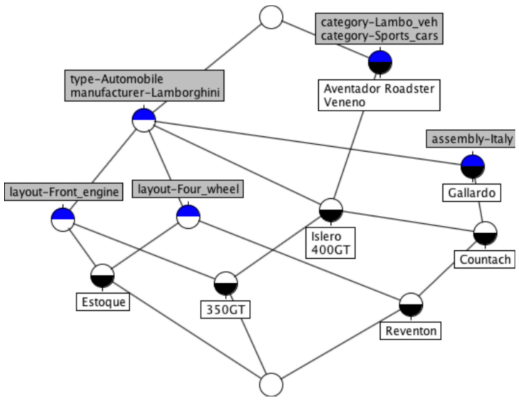


Fig. 1: The formal context shown on the left is built after scaling from DBpedia data given in Table 1. Each cross (×) corresponds to a triple subject-predicate-object. On the right the corresponding concept lattice is shown.

3 Improving DBpedia with FCA

3.1 Problem context

Consider the following fictional scenario. You are a bookkeeper in a library of books written in a language you do not understand. A customer arrives and asks you for a book about "Cars". Since you do not know what the books are about (because you cannot read them), you ask the customer to browse the collection on his own. After he finds a book he is interested to read, you will mark the symbol $\star$ on that book for future references. Then, in an empty page you will write ($\star$ - *Cars*). After several cases like this, you will probably end up with a page full of symbols representing different topics or categories of your books, among them ($\ominus$ - *Sports*), ($\diamond$ - *Football*) and ($\circ$ - *History*). Now you can even combine symbols when customers ask you for "Sport Cars" which you translate into $\star\ominus$. Actually, the demand for books about "Sport Cars" is so high that you create a

new symbol † just for it. So doing, you have created your own categorization system of a collection of books you do not understand.

In general, given a topic, you are able to retrieve books without many troubles, however since you do not understand the books, you are restricted to the set of symbols you have for doing this. Furthermore, if you are not careful some problems start to arise, such as books marked with $\diamond$ and without $\ominus$. Finally, people do not get books marked with † when they look for "Cars", since they only search for the symbol $\ominus$.

It is easy to stablish an analogy on how DBpedia profits from Wikipedia's categorization system and the above scenario. DBpedia is able to retrieve entities when queried with an annotation (as the example of "French films"), however any information need not initially provided as a category is unavailable for retrieval (such as "French films about the Art Nouveau era"). Incoherences in categorical annotations are quite frequent in DBpedia, for example there are over 200 entities annotated as "French films" which are not typed as "Films". Finally, DBpedia is not able to provide inferencing. For example, in Figure 1, the entities Veneno and Aventador, even though they are annotated as "Lamborghini vehicles", cannot be retrieved when queried simply by "vehicles". In such a way, it is exactly as if they were marked with a symbol such as †.

3.2 The completion of DBpedia data

Our main concern in this case lies in two aspects. Firstly, are we able to complete data using logical inferences? For example, can we *complete* the information in the dataset by indicating that the entities "Estoque" and "Gallardo" should be categorized as "Lamborghini vehicles" and "Sport cars"? Secondly, are we able to *complete* the descriptions of a given type? For example, DBpedia does not specify that an "Automobile" should have a "manufacturer". In the following, we try to answer these two questions using implications and association rules.

Consider rules provided in Table 2. Of course, the first three implications are only true in our dataset. This is due to the fact that we use the "closed world" assumption, meaning that our rules only apply in "our world of data" where all cars are of "Lamborghini" brand, i.e. all other information about cars that we do not know can be assumed as false [5]. While these implications are trivial, they provide a good insight of the capabilities of our model. For instance, including a larger number of triples in our dataset would allow discovering that, while not all automobiles are manufactured by Lamborghini, they are manufactured by either a Company, an Organization or an Agent. These three *classes*[5] are types of the entity Lamborghini in DBpedia. Such a rule would allow providing a *domain* characterization to the otherwise empty description of the predicate "dbo:manufacturer" in the DBpedia schema.

The association rule given in the fourth row in Table 2 shows the fact that 29% of the subjects of type "Automobile" and manufactured by "Lamborghini" should be categorized by "Sports cars" and "Lamborghini vehicles" to complete the data. This actually corresponds to the entities "Estoque" and "Gallardo" in Figure 1. Based on this fact, we can use association rules also to create new triples that allow the completion of the information included in DBpedia.

[5] *In the OWL language sense.*

Rule	Confidence	Support	Meaning
d $\Longrightarrow$ c	100%	7	Every automobile is manufactured by Lamborghini.
c $\Longrightarrow$ d	100%	7	Everything manufactured by Lamborghini is an automobile.
e $\Longrightarrow$ b,c	100%	3	All the entities assembled in Italy are Lamborghini automobiles.
c,d $\Longrightarrow$ a,b	71%	7	71% of the Lamborghini automobiles are categorized as "sport cars" and "Lamborghini vehicles"

Table 2: Association rules extracted from formal context in Figure 1.

3.3 Pattern structures for the completion process

The aforementioned models to support linked data using FCA are adequate for small datasets as the example provided. Actually, LOD do not always consists of triples of resources (identified by their URIs) but contains a diversity of *data types* and structures including dates, numbers, collections, strings and others making the process of data processing much more complex. This calls for a formalism able to deal with this diversity of complex and heterogeneous data.

Accordingly, pattern structures are an extension of FCA which enables the analysis of complex data, such as numerical values, graphs, partitions, etc. In a nutshell, pattern structures provide the necessary definitions to apply FCA to entities with complex descriptions. The basics of pattern structures are introduced in [6]. Below, we provide a brief introduction using interval pattern structures [8].

Let us consider Table 3 showing the predicate *dbo:productionStartYear* for the subjects in Figure 1. In such a case we would like to extract a pattern in the year of production of a subset of cars. Contrasting a formal context as introduced in Section 2, instead of having a set M of attributes, interval pattern structures use a semi-lattice of interval descriptions ordered by a subsumption relation and denoted by $(D, \sqsubseteq)$[6]. Furthermore, instead of having an incidence relation set I, pattern structures use a mapping function $\delta : G \rightarrow D$ which assigns to any $g \in G$ the corresponding interval description $\delta(g) \in D$. For example, the entity "350GT" in Table 3 has the description $\delta(350GT) = \langle[1963, 1963]\rangle$.

Let us consider two descriptions $\delta(g_1) = \langle[l_i^1, r_i^1]\rangle$ and $\delta(g_2) = \langle[l_i^2, r_i^2]\rangle$, with $i \in [1..n]$ where n is the number of intervals used for the description of entities. The similarity operation $\sqcap$ and the associated subsumption relation $\sqsubseteq$ between descriptions are defined as the convex hull of two descriptions as follows:

$$\delta(g_1) \sqcap \delta(g_2) = \langle[min(l_i^1, l_i^2), max(r_i^1, r_i^2)]\rangle$$
$$\delta(g_1) \sqsubseteq \delta(g_2) \iff \delta(g_1) \sqcap \delta(g_2) = \delta(g_1)$$
$$\delta(350GT) \sqcap \delta(Islero) = \langle[1963, 1967]\rangle$$
$$(\delta(350GT) \sqcap \delta(Islero)) \sqsubseteq \delta(400GT)$$

Finally, a pattern structure is denoted as $(G, (D, \sqsubseteq), \delta)$ where operators $(\cdot)^\square$ between $\wp(G)$ and $(D, \sqsubseteq)$ are given below:

$$A^\square := \prod_{g \in A} \delta(g) \qquad\qquad d^\square := \{g \in G \mid d \sqsubseteq \delta(g)\}$$

[6] It can be noticed that standard FCA uses a semi-lattice of set descriptions ordered by inclusion, i.e. $(M, \subseteq)$.

An interval pattern concept (A, d) is such as $A \subseteq G$, $d \in D$, $A = d^{\square}$, $d = A^{\square}$. Using interval pattern concepts, we can extract and classify the actual pattern (and pattern concepts) representing the years of production of the cars. Some of them are presented in the lower part of Table 3. We can appreciate that cars can be divided in three main periods of time of production given by the intent of the interval pattern concepts.

Entity	*dbo:productionStartYear*
Reventon	2008
Countach	1974
350GT	1963
400GT	1965
Islero	1967
Veneno	2012
Aventador Roadster	-
Estoque	-
Gallardo	-
Interval Pattern Concepts	
Reventon, Veneno	$\langle[2008, 2012]\rangle$
Countach,	$\langle[1974, 1974]\rangle$
350GT,400GT,Islero	$\langle[1963, 1967]\rangle$

Table 3: Upper table shows values of predicate *dbp:productionStartYear* for entities in Figure 1. The symbol - indicates that there are no values present in DBpedia for those subjects. Lower table shows the derived interval pattern concepts .

3.4 Heterogeneous pattern structures

Different instances of the pattern structure framework have been proposed to deal with different kinds of data, e.g. graph, sequences, interval, partitions, etc. For linked data we propose to use the approach called "heterogeneous pattern structure" framework introduced in [4] as a way to describe objects in a heterogeneous space, i.e. where there are relational, multi-valued and binary attributes. It is easy to observe that this is actually the case for linked data where the set of literals L greatly varies in nature depending on the predicate. For the sake of simplicity we provide only the most important details of the model used for working with linked data.

When the range of a predicate (hereafter referred to as "relation") $p \in P$ is such that $range(p) \subseteq U$, we call p an "object relation". Analogously, when the range is such that $range(p) \subseteq L$, p is a "literal relation". For any given relation (object or literal), we define the pattern structure $\mathcal{K}_p = (G, (D_p, \sqcap), \delta_p)$, where $(D_p, \sqsubseteq)$ is an ordered set of descriptions defined for the elements in $range(p)$, and δ_p maps entities $g \in G$ to their descriptions in D_p. Based on that, the triple (G, H, Δ) is called a "heterogeneous pattern structure", where $H = \times D_p(p \in P)$ is the Cartesian product of all the descriptions sets D_p, and Δ maps an entity $g \in G$ to a tuple where each component corresponds to a description in a set D_p.

For an "object relation", the order in $(D_p, \sqsubseteq)$ is given by standard set inclusion and thus, the pattern structure $\mathcal{K}_p$ is just a formal context. Regarding "literal relations", such as numerical properties, the pattern structure may vary according to what is more appropriate to deal with that specific kind of data. For example, considering the predicate

dbo:productionStartYear discussed in the previous section, $\mathcal{K}_{\text{dbo:productionStartYear}}$ should be modelled as an interval pattern structure. For the running example, the heterogeneous pattern structure is presented in Table 4. Cells in grey mark a *heterogeneous pattern concept* the extent of which contains cars "350GT, 400GT, Islero". The intent of this heterogeneous pattern concept is given by the tuple $(\{a, b\}, \{c\}, \{d\}, \langle[1963, 1967]\rangle)$, i.e. *"Automobiles manufactured by Lamborghini between 1963 and 1967"*.

	$\mathcal{K}_A$ a	b	$\mathcal{K}_B$ c	$\mathcal{K}_C$ d	$\mathcal{K}_D$ e	$\mathcal{K}_E$ f	g	$\mathcal{K}_{\text{dbo:productionStartYear}}$
Reventon	×	×	×	×	×	×		$\langle[2008, 2008]\rangle$
Countach	×	×	×	×	×			$\langle[1974, 1974]\rangle$
350GT	×	×	×	×		×		$\langle[1963, 1963]\rangle$
400GT	×	×	×	×				$\langle[1965, 1965]\rangle$
Islero	×	×	×	×				$\langle[1967, 1967]\rangle$
Veneno	×	×						$\langle[2012, 2012]\rangle$
Aventador Roadster	×	×						-
Estoque			×	×		×	×	-
Gallardo			×	×	×			-

Table 4: Heterogeneous pattern structure for the running example. Indexes for properties are shown in Table 1.

4 Conclusion

To conclude, in the current study we introduce a mechanism based on association rule mining for the completion of the RDF dataset. Moreover, we use heterogeneous pattern structures to deal with heterogeneity in LOD. This study shows the capabilities of FCA for completing complex RDF structures.

References

1. Mehwish Alam, Aleksey Buzmakov, Victor Codocedo, and Amedeo Napoli. Mining definitions from rdf annotations using formal concept analysis. In *IJCAI 2015, Proceedings of the 24th International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, July 25-31, 2015*, 2015.
2. Dominik Benz, Andreas Hotho, and Gerd Stumme. Semantics made by you and me: Self-emerging ontologies can capture the diversity of shared knowledge. In *Proceedings of the 2nd Web Science Conference*, 2010.
3. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data - the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.
4. Víctor Codocedo and Amedeo Napoli. A Proposition for Combining Pattern Structures and Relational Concept Analysis. In *12th International Conference on Formal Concept Analysis*. 2014.
5. Christian Fürber and Martin Hepp. Swiqa - a semantic web information quality assessment framework. In *19th European Conference on Information Systems*, 2011.
6. Bernhard Ganter and Sergei O. Kuznetsov. Pattern structures and their projections. In *ICCS*, volume 2120 of *Lecture Notes in Computer Science*, pages 129–142. Springer, 2001.
7. Bernhard Ganter and Rudolf Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Berlin/Heidelberg, 1999.
8. Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli, and Sébastien Duplessis. Mining gene expression data with pattern structures in formal concept analysis. *Information Sciences*, 181(10):1989–2001, 2011.

A Conceptual-KDD tool for ontology construction from a database schema

Renzo Stanley and Hernán Astudillo

Universidad Técnica Federico Santa María, Av. España 1680, Valparaíso, Chile
{rstanley,hernan}@inf.utfsm.cl

Abstract. The UNESCO convention on Intangible Cultural Heritage (ICH) requires countries to document their oral traditions, performing arts, traditional festivities, and so forth. Several institutions gather ICH, traditionally by hand, and record and disseminate it through conventional information systems (static knowledge in relational databases, RDB). Two difficulties are that (1) review/refinement of their underlying database schemata by domain experts becomes disruptive, and (2) contribution from community, non-expert users becomes hard, even impossible. This article presents an interactive tool that implements a recent technique to perform Knowledge Discovery in Databases (KDD) guided by Formal Concept Analysis (FCA). The tool takes an RDB schema (in SQL), translates it into a formal context and later in a concept lattice using the CORON platform, allows domain experts to manipulate it and produces a formal ontology (in RDFS). Later, the ontology can be used to instantiate a semantic wiki as community collaboration tool, for example. The technique and tool are illustrated with an example from the ICH domain, using Chile's Culture Ministry online data. The tool is also available online.

Keywords: Formal Concept Analysis, Knowledge Discovery in Databases, Ontologies, Intangible Cultural Heritage

1 Introduction

The Chilean National Council of Culture and Arts[1] (CNCA) has undergone the mission of documenting the ICH of different areas of the country in the context of a world-wide UNESCO[2] convention to incentive the *states parties*[3] and NGOs to properly maintain their cultural knowledge. Considering the dynamic structure (data, concepts and relations) of this domain, the conventional information management systems should be sufficiently flexible in order to support changes and community collaboration such as well-known wikis [5]. For these reasons, CNCA needs a tool that allows to simplify the process of refinement of their current relational database model. KDD emerged as a tool to support humans

[1] http://cultura.gob.cl
[2] United Nations Educational, Scientific, and Cultural Organization
[3] http://whc.unesco.org/en/statesparties/

in the discovery and extraction of knowledge from large collections of data (usually stored in databases) where a manual approach for such task is very difficult (or nearly impossible) [3]. Thus, the *human-centered* nature of the approach is a key factor in any KDD process [1] since it has to ensure that knowledge is not only successfully found, but also understood by the final user. For this reason, FCA proved to be a good support for a KDD process given its two-folded manner of representing knowledge, i.e. as concepts containing an extent (instances of the concept) and an intent (the attributes of the concept) [8]. To stress this fact, we quote [7] in the relation of FCA and KDD: "the process of concept formation in FCA is a KDD *par excellence*". FCA has been used to support KDD in several tasks for different domains. For example, [4] states that nearly 20% of the papers in the FCA domain consist on knowledge discovery related approaches. Furthermore, in [2] FCA is presented as the cornerstone of *Conceptual Knowledge Discovery in Databases (CKDD)* described as a human-centered process supporting the visual analysis of a conceptual structure of data for a given context of information. Since the principal difficulty of CNCA (reviewing and refinement of ICH model) are rooted in a database schema analysis and amelioration which heavily requires human domain expertise, we rely on a CKDD tool to redesign the data schema already in use and to elicit an ontological schema from it.

In this article, we show a tool that implements an iterative and human-centred approach based on KDD and FCA. This method uses the concept lattice generated as a support for guiding the redesign process, considering the relevant knowledge of experts. This approach was proposed in an earlier work [6], however applies it in a web-based tool that allows any user work with his own schema.

The reminder of this article is organized as follows: Section 2 resumes the method proposed, Section 3 describes in detail the principal functionalities of the tool developed, Section 4 outlines an example for validating the tool with a domain expert. Finally, Section 5 presents a discussion on future work and concludes the paper.

2 Method

Figure 1 presents a 3-step CKDD process designed to take a database schema and translating it into an ontological schema. In the following, we provide a general view of the tasks at each step.

2.1 First step: Data Preprocessing

The first step starts by extracting the database schema and ends when it is converted to a formal context. This step consists of three tasks: (1) Schema processing, (2) Attribute integration and (3) Relational attribute scaling. However, this process is fully automatized by the tool, and does not require expert intervention.

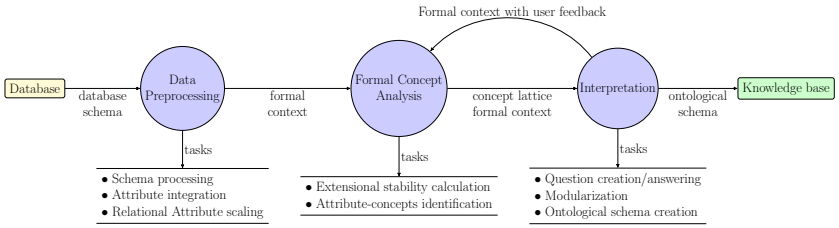


Fig. 1: FCA-based KDD process

2.2 Second step: Formal Concept Analysis

This step receives a formal context and ends when a concept lattice is constructed. The tasks performed are: (1) Extensional stability calculation and (2) Attribute-concepts identification and these are calculated using the Coron Platform. The extensional stability value and the attribute-concept calculated are shown to a domain expert for the next step.

2.3 Third step: Interpretation

The final step receives a formal context and its associated concept lattice where each attribute concept has been identified and each formal concept contains an *extensional stability* value. The tasks performed for this step are: (1) Question creation/answering, (2) Modularization, (3) Ontological schema creation. The options (1) and (2) allow the user to make another iteration sending a modified version of the formal concept received according to user feedback, but option (3) allows the user to end the process, an "ontological schema" will be created and it will be downloaded by the user in RDF file format.

2.4 Ontological schema creation

The final task of the process converts the concept lattice into an ontological schema which can be used for data integration and linked data publication. This schema is obtained by creating a set of RDF triples for the elements of the concept lattice. Table 1 shows a overview of the rules used to create the ontological schema. This table is based on an adapted definition of the relational data schema model.

Relational data schema model: A relational schema $S = \{R_1, R_2, ..., R_{|S|}\}$ is defined as a set of tables or "relation schemas" $R_i(A_1, A_2, ..., A_n)$ consisting of a table name R_i and a list of fields A_j which are value assignments of the domain $dom(A_j)$ to an *entry* in the table. The notation $R_i.A_j$ stands for the field A_j in table R_i.

Table 1: Formal concepts translation into an ontological schema [6].

Concept	Element	Actions
$\top = (S, S')$	$R_i \in S$	R_i `rdf:type rdfs:Class` *e.g. cnca:*Agent *rdf:type rdfs:Class*
$\bot = (A', A)$	$A_j \in A$	A_j `rdf:type rdfs:Property` A_j `rdfs:range rdfs:Literal` *e.g. cnca:establishment rdf:type rdfs:Property* *cnca:establishment rdfs:range rdfs:Literal*
$\bot = (A', A)$	`related_to:`$R_i \in A$	`related_to:`R_i `rdf:type rdfs:Property` `related_to:`R_i `rdf:range rdfs:`R_i *e.g. cnca:participant rdf:type rdfs:Property* *cnca:participant rdfs:range cnca:Agent*

$\bot = (A', A)$	`domain:Label` $\in A$	`cnca:Label rdf:type cnca:Domain`
		`cnca:Domain rdf:type rdfs:Class`
		`cnca:in_domain rdf:type rdfs:Property`
		e.g. cnca:People rdf:type cnca:Domain

$\mu A_j = (A'_j, A''_j)$	$R_i \in A'_j$	`cnca:`A_j `rdfs:domain cnca:`R_i *e.g. cnca:participant rdfs:domain cnca:Ritual*
$\mu A_j = (A'_j, A''_j)$	($A_j =$`domain:Label` $\wedge$ $R_i \in A'_j$)	`cnca:`R_i `cnca:in_domain cnca:Label` *e.g. cnca:Agent cnca:in_domain cnca:People*

This task is also interactive allowing the user to take most of the decisions w.r.t. how the ontological schema should be created. In the following, we refer to *cnca:* as the prefix used for the schema to be created.

Top Concept $\top = (S, S')$: All tables are modelled using the resource description framework schema (RDFS) element *rdfs:Class* by default (e.g. *cnca:Agent a rdfs:Class*). The user may choose to annotate some of them with the element *rdfs:Resource*. For the set of attributes in S', we provide a list of properties from RDFS and the *dublin core* ontology [4] where the user can select mappings going from the attributes to the ontology. For example, the attribute *name* is mapped to the property rdfs:label. The special attribute *id* is disregarded as its value in each entry is only considered to create a unique an valid URI [5].

Bottom Concept $\bot = (A', A)$**:** All fields in A are modelled according to their nature: *relational*, *non-relational attributes* or *special attributes*.

- *Regular attributes* are modelled by default using the *rdfs:Property* while the *cnca:* prefix is added to its name (e.g. *cnca:establishment a rdfs:Property*). In addition, the range of the property is set to rdfs:Literal (e.g. *cnca:establishment* rdfs:range rdfs:Literal).
- *Relational attributes* of the form *related_to:table* are modelled with *rdfs:Property* and the range is set to the table they refer to. Additionally, the user is asked to rename the relation (e.g *related_to:Agent* is modelled as *cnca:participant*

[4] `http://www.w3.org/wiki/Good_Ontologies#The_Dublin_Core_.28DC.29_ontology`

[5] Universal resource identifier.

a rdfs:Property; cnca:participant rdfs:range cnca:Agent). While the user may also be requested to create the inverse property, this feature is not available in RDFS and for the sake of simplicity we have disregard the use of OWL for now.
- *Special attributes* of the form *domain:Label* are modelled differently. For each different *domain:Label* we create a resource *cnca:Label a cnca:Domain* where *cnca:Domain a rdfs:Class* (e.g. *cnca:People a cnca:Domain*). A single property *cnca:in_domain a rdfs:Property; rdfs:range cnca:Domain; rdfs:domain rdfs:Class* is created to annotate classes created from tables.

Attribute concepts $\mu A_i = (A_i', A_i'')$: For each attribute concept, we use its extent to set the domain of the already modelled properties in its intent creating *cnca:*A_i *rdfs:domain cnca:R* for all $R \in A_i'$ (e.g. *cnca:participant rdfs:domain (cnca:Festive_Event,cnca:Ritual)*). For the special attributes of the form *domain:Label*, objects are annotated using *cnca:R cnca:in_domain cnca:Label* for all $R \in A_i'$ (e.g. *cnca:Agent cnca:in_domain cnca:People*).

There are some other actions taken during modelling, however for the sake of space and simplicity we do not discuss these in here.

3 Tool

The web-based tool intended to construct an ontological schema for a specific SQL relational database schema is compound of two principals components: (1) the CORON platform to calculate concept lattices and the stabilities values of each attribute-concept, and (2) the python backend application connecting user interface with CORON in order to execute functions that manage formal contexts, attribute-concept detections and ontology generation. Thus, the tool allows domain experts obtain an ontology in RDF file format.

3.1 Technology

This tool was developed on Python 2.7 and Flask micro-framework[6]. For developing the following technologies are used, namely: SQLAlchemy ORM[7] to connect Python to the DB schema, *python concepts*[8] to translate the DB schema to a formal context for the first time. Also we used the Coron Platform[9] to calculate the concept lattices and their extensional stabilities in order to identify the attribute concept in each iteration. RDFLib[10] was used for working with RDF files in Python. At this moment, the tool is available in `http://dev.toeska.cl/rstanley/rdb2ontology`. Once there, you can create a user account and connect it with your own MySQL DB schema.

[6] Flask `http://flask.pocoo.org/`
[7] Python Object Relational Mapper (ORM) `http://www.sqlalchemy.org/`
[8] Concepts: a python library for Formal Concept Analysis `https://pypi.python.org/pypi/concepts`
[9] Coron System: a symbolic data-mining platform `http://coron.loria.fr/site/index.php`
[10] RDFLib `https://github.com/RDFLib/rdflib`

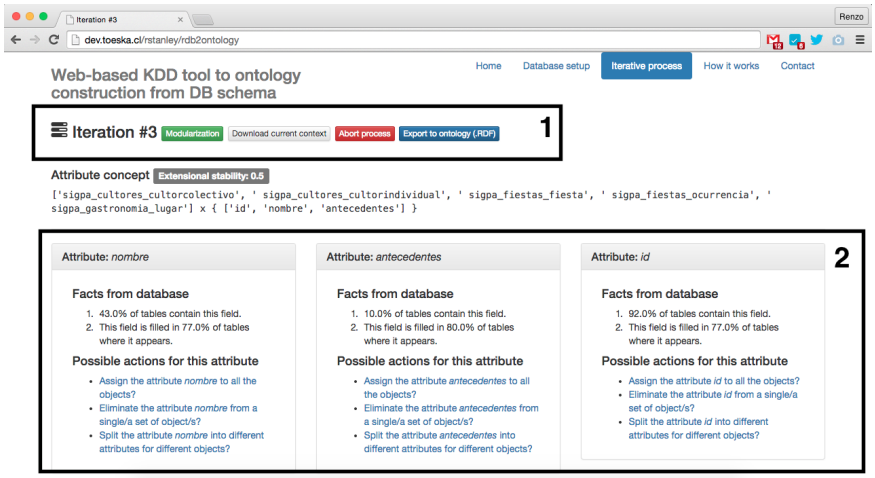


Fig. 2: Screen capture of an iteration

3.2 Functionalities

To provide a way to modify the underlying formal context for the domain expert we implemented some functionalities that can be looked at in figure 2. These actions are divided in two groups named *general options group* marked with #1 and *attribute-specific options* marked with #2. They are available for the domain expert in each iteration. Firstly, the *general options group* is composed by (1) Modularization, (2) Download current context, (3) Abort process, (4) Export to RDFS ontology. Secondly, the *attribute-specific options group* contains a set of actions to modify each attribute depending of a expert decision, namely: (1) Assign the attribute to all the objects?, (2) Eliminate the attribute from a single/a set of objects?, (3) Split the attribute into different attributes for different objects? For the sake of space and simplicity, we have left out the explanation of each of these options as it can be found in depth in our previous work [6].

4 Example

The database schema of CNCA[11] includes nearly 100 tables, however, for this example we have selected only 24 tables representing multi-disciplinary knowledge. These tables contain 24 objects, 53 attributes, and 13 relational attributes. The database schema for this example represents descriptions of *agents*, *collective agents*, *festive events*, *culinary manifestations*, *geolocations* and more. Figure 3 depicts the concept lattice obtained from the formal context generated by the

[11] Chilean National Council of Culture and Arts

database schema. Table 2 shows the decisions taken by the domain expert during 14 iterations. These decisions are based on question answering, domain labeling (modularization) or stoping the iterations.

Table 2: Iterations made by the domain expert

Iteration number	**Attribute**	**Action**
1	`name`	Assign to all tables
2	`background`	Split the attribute
3	`background`	Split the attribute
4	`views`	Eliminate from some tables
5	`published`	Eliminate from some tables
6	`description`	Assign to all tables
7	`founding_date`	Split the attribute
8	`related_to Agent`	Eliminate from *Ritual* table
9	-	Domain labelling: Culinary descriptors
10	`domain:culinary`	Eliminate from *CulinaryPlace* table
11	-	Domain labelling: ICH
12	-	Domain labelling: Agent descriptors
13	-	Domain labelling: Festive descriptors
14	-	Domain labelling Geo descriptors

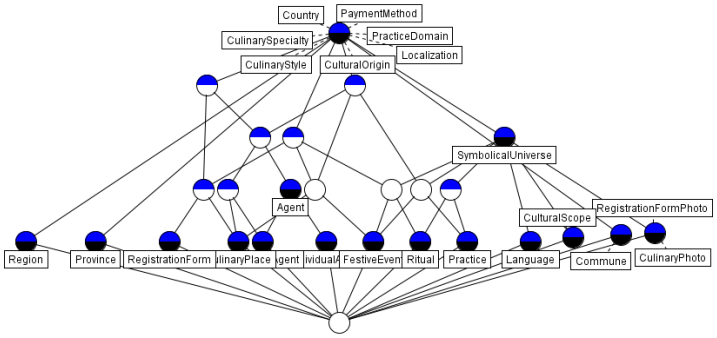


Fig. 3: Initial lattice obtained automatically from database schema

Figure 4 illustrates the final concept lattice presenting the refined structure after 14 iterations of the domain expert. We can distinguish several modules of information that have been marked. The expert called these modules as *ICH subdomains* identified from left to right, namely: *Festive Event descriptors sub-*

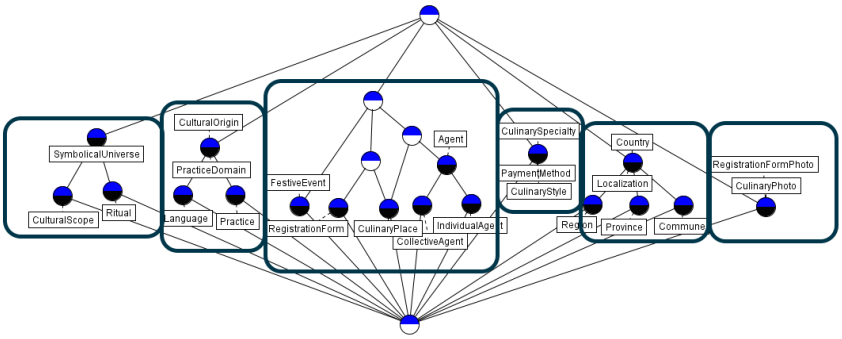


Fig. 4: Final lattice obtained after 14 iterations. Each ICH subdomain found have been marked.

domain, *Agent descriptors subdomain*, *ICH inventory subdomain*, *Culinary descriptors subdomain*, *Geographical subdomain*, *Photo subdomain*.

5 Conclusions and Future Work

To conclude, in this article we have presented a web-based tool fully functional based on an approach published in a previous work [6]. In this earlier work a case study was exposed obtaining interesting results, however these results were obtained executing calls to CORON platform in a manual way with the intervention of a knowledge engineer. The difference between the previous work and this work is that the tool allows a domain expert to get an ontological schema himself in RDFS. In the example showed in section 4 we obtained 14 iterations from a similar excerpt of a database schema, however in the previous case study executed in [6] we obtained 9 iterations, so the resulting concept lattices were very similar. In each lattice the same modules were found, however, the time to reach the same result was higher. We have to consider that the expert used the tool without the assistance of a knowledge engineer. Currently, we are implementing the next step of this tool related to construct a semantic wiki based on the ontological schema. So even though the ontology obtained was simple, the domain expert could enrich it by using annotations in a semantic wiki. Also, this wiki could aid a domain expert in order to collaborate in the documenting process.

References

1. Ronald J. Brachman and Tej Anand. The process of knowledge discovery in databases. In *Advances in Knowledge Discovery and Data Mining*, pages 37–57.

1996.

2. Joachim Hereth Correia, Gerd Stumme, Rudolf Wille, and Uta Wille. Conceptual knowledge discovery–a human-centered approach. *Applied Artificial Intelligence*, 17(3):281–302, March 2003.
3. Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: an overview. In *Advances in knowledge discovery and data mining*, pages 1–34. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.
4. Jonas Poelmans, Paul Elzinga, Stijn Viaene, and Guido Dedene. Formal concept analysis in knowledge discovery: a survey. In *Proceedings of the 18th international conference on Conceptual structures: from information to intelligence*, ICCS'10, pages 139–153, Berlin, Heidelberg, 2010. Springer-Verlag.
5. Renzo Stanley and Hernan Astudillo. Ontology and semantic wiki for an intangible cultural heritage inventory. In *Computing Conference (CLEI), 2013 XXXIX Latin American*, pages 1–12, Oct 2013.
6. Renzo Stanley, Hernan Astudillo, Victor Codocedo, and Amedeo Napoli. A conceptual-kdd approach and its application to cultural heritage. In Manuel Ojeda-Aciego and Jan Outrata, editors, *Concept Lattices and their Applications*, pages 163–174, La Rochelle, France, October 2013. L3i laboratory, University of La Rochelle.
7. Petko Valtchev, Rokia Missaoui, and Robert Godin. Formal concept analysis for knowledge discovery and data mining: The new challenges. In Peter W. Eklund, editor, *ICFCA*, volume 2961 of *Lecture Notes in Computer Science*, pages 352–371. Springer, 2004.
8. Rudolf Wille. Why can concept lattices support knowledge discovery in databases? *Journal of Experimental and Theoretical Artificial Intelligence*, 14(2-3):81–92, 2002.

Σοφια: how to make FCA polynomial?

Aleksey Buzmakov[1,2], Sergei Kuznetsov[2], and Amedeo Napoli[1]

[1]LORIA (CNRS – Inria NGE – Université de Lorraine), Vandœuvre-lès-Nancy, France
[2]National Research University Higher School of Economics, Moscow, Russia
aleksey.buzmakov@loria.fr, skuznetsov@hse.ru, amedeo.napoli@loria.fr

Abstract. In pattern mining, one of the most important problems is fighting exponential explosion of the set of patterns. A typical solution is generating only a part of all patterns satisfying some criteria. The most well-known criterion is support of a pattern, which has the monotonicity property allowing one to generate only frequent (highly supported) patterns. Many other useful criteria are not monotonic, which makes it difficult to generate best patterns efficiently. In this paper we introduce the notion of "generalized monotonicity" and Σοφια algorithm that allow to generate top patterns in polynomial time modulo basic operations, e.g., measure computation, for criteria that are not monotonic. This approach is applicable not only to itemsets, but to complex descriptions such as sequences, graphs, numbers or interval tuples, etc. In this paper we consider stability and Δ-measures which are not monotonic. In the experiments, we compute top best patterns w.r.t. these measures and obtain very promising results.

1 Introduction

To solve the problem of exponential explosion of patterns valid in a dataset many kinds of interestingness measures were proposed [1]. For example, pattern support, i.e., the number of objects covered by the pattern, is one of the most well-known measures of pattern quality. Among others stability of a formal concept [2] can be mentioned. Unlike support this measure is not monotonic w.r.t. the order of pattern inclusion and it is hard to generate only most interesting patterns w.r.t. these measures, so one has to find a large set of patterns and then postprocess it, choosing the best ones.

Due to the increasing importance of pattern mining, efficient approaches of finding best patterns are appearing. In [3] authors introduce an approach for efficiently searching the most interesting associations w.r.t. lift or leverage of a pattern. Another approach is searching for cosine interesting patterns [4]. The cosine interestingness of a pattern is not a monotonic measure but the authors take advantage of a conditional anti-monotonic property of cosine interestingness to efficiently mine interesting patterns. However, all of the mentioned approaches are not polynomial in the worst case.

In this paper we introduce a new algorithm Σοφια (Sofia, for Searching for Optimal Formal Intents Algorithm) for extracting top best patterns of different

kinds, i.e., itemsets, string, graph patterns, etc. Σοφια algorithm is applicable to a class of measures, including classical monotonic measures, stability, δ-freeness [5], etc. For itemset mining, our algorithm can find top best patterns w.r.t. a measure from this class in polynomial time, modulo complexity of measure computation. For more complex description the time is polynomial modulo complexity of basic operations (intersecting and testing containment on descriptions, computation of a measure).

2 Preliminaries

FCA is a formalism convenient for describing models of itemset mining and knowledge discovery [6]. Here we briefly define pattern structures and the corresponding notations [7]. A *pattern structure* is a triple $\mathbb{P} = (G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a meet-semilattice of descriptions such that $(\forall X \subseteq G) \bigsqcap X \in D$ and $\delta : G \to D$ maps an object to a description. The intersection $\sqcap$ gives similarity of two descriptions.

Let us denote the derivative operators of the Galois connection between 2^G and D by $(\cdot)^\diamond$ (see [7]). A *pattern concept* of a pattern structure $(G, (D, \sqcap), \delta)$ is a pair (A, d), where $A \subseteq G$, called *pattern extent* and $d \in D$, called *pattern intent*, such that $A^\diamond = d$ and $d^\diamond = A$. The set of all pattern concepts is partially ordered w.r.t. inclusion on extents, i.e., $(A_1, d_1) \leq (A_2, d_2)$ iff $A_1 \subseteq A_2$ (or, equivalently, $d_2 \sqsubseteq d_1$), making a lattice, called pattern lattice.

For real datasets, the number of patterns can be large. In order to reduce the most interesting concepts different measures can be used. In this paper we rely on stability [2], which measures the independence of a concept intent w.r.t. randomness in data. Because of limited space we do not discuss this measure in details here. Moreover, since concept stability is hard to compute, we rely on an estimate of concept stability which can be computed in polynomial time for a single concept [8].

The approach proposed in this paper is based on projections introduced for reducing complexity of computing pattern lattices [7]. A *projection operator* $\psi : D \to D$ is an "interior operator", i.e. it is (1) monotonic ($x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$), (2) contractive ($\psi(x) \sqsubseteq x$) and (3) idempotent ($\psi(\psi(x)) = \psi(x)$).

An *o-projected pattern structure* (projected pattern structure for simplicity) $\psi((G, (D, \sqcap), \delta))$ is a pattern structure $\psi(\mathbb{P}) = (G, (D_\psi, \sqcap_\psi), \psi \circ \delta)$, where $D_\psi = \psi(D) = \{d \in D \mid \exists \tilde{d} \in D : \psi(\tilde{d}) = d\}$ and $\forall x, y \in D, x \sqcap_\psi y := \psi(x \sqcap y)$ [9]. Given a projection ψ we say that the fixed set of ψ is the set of all elements from D which are mapped to themselves by ψ. The fixed set of ψ is denoted by $\psi(D) = \{d \in D \mid \psi(d) = d\}$. Any element outside of the fixed set of ψ is pruned from the description space. We say that a projection ψ_1 is *s*impler than a projection ψ_2, denoted by $\psi_1 < \psi_2$, if $\psi_1(D) \subset \psi_2(D)$, i.e., ψ_2 prunes less descriptions than ψ_1.

Our algorithm is based on this order on projections. The simpler a projection ψ is, the less patterns we can find in $\psi(\mathbb{P})$, and the less computational efforts one should take. Thus, we compute a set of patterns for a simpler projection, then

we remove unpromising patterns and extend our pattern structure and the found patterns to a more detailed projection. This allows to reduce the size of patterns within a simpler projection in order to reduce the computational complexity of more detailed projection.

3 Σοφια Algorithm

3.1 Monotonicity w.r.t. a Projection

Our algorithm is based on the projection monotonicity, a new idea introduced in this paper. Many interestingness measures for patterns, e.g., stability, are not monotonic w.r.t. subsumption order on patterns, i.e., given patterns X and Y such that $X \sqsubseteq Y$, and a nonmonotonic measure $\mathcal{M}$, one does not necessarily have $\mathcal{M}(X) \geq \mathcal{M}(Y)$. For instance, support is a monotonic measure w.r.t. pattern order and it allows for efficient generation of patterns with support higher than a threshold [10]. The projection monotonicity is a generalization of standard monotonicity and allows for efficient work with a wider set of interestingness measures.

Definition 1. *Given a pattern structure $\mathbb{P}$ and a projection ψ, a measure $\mathcal{M}$ is called* monotonic w.r.t. the projection ψ, *if*

$$(\forall p \in \psi(\mathbb{P}))(\forall q \in \mathbb{P}, \psi(q) = p)\mathcal{M}_\psi(p) \geq \mathcal{M}(q), \tag{1}$$

where $\mathcal{M}_\psi(p)$ is the measure $\mathcal{M}$ of pattern p computed in $\psi(\mathbb{P})$.

Here, for any pattern p of a projected pattern structure we check that a preimage q of p for ψ, e.g. $p = \psi(q)$, has a measure smaller than the measure of p. It should be noticed that a measure $\mathcal{M}$ for a pattern p can yield different values if $\mathcal{M}$ is computed in $\mathbb{P}$ or in $\psi(\mathbb{P})$. Thus we use the notation $\mathcal{M}_\psi$ for the measure $\mathcal{M}$ computed in $\psi(\mathbb{P})$.

An important example is given by binary data or formal contexts (G, M, I). In this case, a projection ψ_m corresponds to the removal of an attribute $m \in M$, i.e., $\psi_m(B) = B \cap (M \setminus \{m\})$ for any $B \subseteq M$. So Definition 1 means that the interestingness of an itemset p w.r.t. a measure $\mathcal{M}$ computed in $(G, M \setminus \{m\}, I \setminus G \times \{m\})$ should be higher than the interestingness of the itemsets p and $p \cup \{m\}$ (the preimages of p for ψ_m) w.r.t. the measure $\mathcal{M}$ computed in (G, M, I). If the value of a measure for a pattern does not depend on a projection this definition is related to a classical monotonic measure. Indeed, because of contractivity of ψ ($\psi(p) \sqsubseteq p$), for any monotonic measure one has $\mathcal{M}(\psi(p)) \geq \mathcal{M}(p)$.

Thus, given a measure $\mathcal{M}$ monotonic w.r.t. a projection ψ, if p is a pattern such that $\mathcal{M}_\psi(p) < \theta$, then $\mathcal{M}(q) < \theta$ for any preimage q of p for ψ. Hence, if, given a pattern p of $\psi(\mathbb{P})$, one can find all patterns q of $\mathbb{P}$ such that $\psi(q) = p$, it is possible to find the patterns of $\psi(\mathbb{P})$ and then to filter them w.r.t. $\mathcal{M}_\psi$ and a threshold, and finally to compute the preimages of filtered patterns.

3.2 Monotonicity w.r.t. a Chain of Projections

However, given just one projection, it can be hard to efficiently discover the patterns, because the projection is either hard to compute or the number of unpromising patterns that can be pruned is not high. Hence we introduce *a chain of projections* $\psi_0 < \psi_1 < \cdots < \psi_k = \mathbb{1}$, where the whole pattern lattice for $\psi_0(\mathbb{P})$ can be easily computed and $\mathbb{1}$ is the identity projection, i.e., $(\forall x)\mathbb{1}(x) = x$. For example, to find frequent itemsets, we typically search for small itemsets and, then, extend them to larger ones. It corresponds to extension to a more detailed projection.

Let us discuss what is a chain of projections in the case of a binary context $\mathbb{K} = (G, M, I)$ with $M = \{m_1, \cdots, m_N\}$. It can be seen that any subcontext $\mathbb{K}_s = (G, N, I \cap G \times N)$, where $N \subseteq M$, corresponds to a projection ψ such that $\psi(B \subseteq M) = B \cap N$. If we put $M_i = \{m_1, \cdots, m_i\}$, then we can consider a chain of projections corresponding to the subset of attributes $M_1, M_2, \cdots, M$. The corresponding projections are properly ordered. Now we define the projection monotonicity of $\mathcal{M}$ w.r.t. a chain of projections.

Definition 2. *Given a pattern structure* $\mathbb{P}$ *and a chain of projections* $\psi_0 < \psi_1 < \cdots < \psi_k = \mathbb{1}$*, a measure* $\mathcal{M}$ *is called* monotonic w.r.t. the chain of projections *if* $\mathcal{M}$ *is monotonic w.r.t. all* ψ_i *for* $0 \leq i \leq k$*.*

3.3 Algorithms

Given a measure monotonic w.r.t. a chain of projections, if we are able to find all preimages of any element in the fixed set of ψ_i that belong to a fixed set of ψ_{i+1}, then we can find all patterns of $\mathbb{P}$ with a value of $\mathcal{M}$ higher than a given threshold θ. We call this algorithm ϑ-Σοφια and its pseudocode is given in Fig. 1. In lines 11-12 we find all patterns for $\psi_0(\mathbb{P})$ satisfying the constraint that a value of $\mathcal{M}$ is higher than a threshold. Then in lines 13-15 we iteratively extend projections from smaller to bigger ones. The extension is done by constructing the set $\mathcal{P}_i$ of preimages of the set $\mathcal{P}_{i-1}$ (lines 2-5) and then by removing the patterns that do not satisfy the constraint from the set $\mathcal{P}_i$ (lines 6-9).

The algorithm is sound and complete, because first, when we compute the set of preimages of a pattern p, the pattern p is a preimage of itself ($\psi(p) = p$) and second, if we remove a pattern p from the set $\mathcal{P}$, then the value $\mathcal{M}(p) < \theta$ and, hence, the measure value of any preimage of p is less than θ by the projection chain monotonicity of $\mathcal{M}$.

The worst-case time complexity of ϑ-Σοφια algorithm is

$$\begin{aligned}\mathbb{T}(\vartheta\text{-Σοφια}) = \mathbb{T}(&FindPatterns(\psi_0))+ \\ &+ k \cdot \max_{0<i\leq k} |\mathcal{P}_i| \cdot (\mathbb{T}(Preimages) + \mathbb{T}(\mathcal{M})),\end{aligned} \tag{2}$$

where $\mathbb{T}(X)$ is time for computing operation X. Since projection ψ_0 can be chosen to be very simple, in a typical case the complexity of $FindPatterns(\theta, \psi_0)$ can be low or even constant. The complexities of $Preimages$ and $\mathcal{M}$ depend on the

Data: A pattern structure $\mathbb{P}$, a chain of projections $\Psi = \{\psi_0, \psi_1, \cdots, \psi_k\}$, a measure $\mathcal{M}$ monotonic for the chain Ψ, and a threshold θ for $\mathcal{M}$.

Function `ExtendProjection`(i, θ, $\mathcal{P}_{i-1}$)

  Data: i is the projection number to which we should extend ($0 < i \leq k$), θ is a threshold value for $\mathcal{M}$, and $\mathcal{P}_{i-1}$ is the set of patterns for the projection ψ_{i-1}.

  Result: The set $\mathcal{P}_i$ of all patterns with the value of measure $\mathcal{M}$ higher than the threshold θ for ψ_i.

  $\mathcal{P}_i \longleftarrow \emptyset$;

  `/* Put all preimages in` $\psi_i(\mathbb{P})$ `for any pattern` p `*/`

  foreach $p \in \mathcal{P}_{i-1}$ **do**

    $\mathcal{P}_i \longleftarrow \mathcal{P}_i \cup$ `Preimages(`i,p`)`

  `/* Filter patterns in` $\mathcal{P}_i$ `to have a value of` $\mathcal{M}$ `higher than` θ `*/`

  foreach $p \in \mathcal{P}_i$ **do**

    if $\mathcal{M}_{\psi_i}(p) \leq \theta$ **then**

      $\mathcal{P}_i \longleftarrow \mathcal{P}_i \setminus \{p\}$

Function `Algorithm_θ-Σοφια`

  Result: The set $\mathcal{P}$ of all patterns with a value of $\mathcal{M}$ higher than the threshold θ for $\mathbb{P}$.

  `/* Find all patterns in` $\psi_0(\mathbb{P})$ `with a value of` $\mathcal{M}$ `higher than` θ `*/`

  $\mathcal{P} \longleftarrow$ `FindPatterns(`θ, ψ_0`)`;

  `/* Run through out the chain` Ψ `and find the result patterns */`

  foreach $0 < i \leq k$ **do**

    $\mathcal{P} \longleftarrow$ `ExtendProjection(`$i, \theta, \mathcal{P}$`)`;

Algorithm 1: The ϑ-Σοφια algorithm for finding patterns in $\mathbb{P}$ with a value of a measure $\mathcal{M}$ higher than a threshold θ.

measure in use and on the instantiation of the algorithm. In many cases $\max\limits_{0<i\leq k} |\mathcal{P}_i|$ can be exponential in the size of the input, because the number of patterns can be exponential. It can be a difficult task to define the threshold θ *a priori* such that the maximal cardinality of $\mathcal{P}_i$ is not higher than a given number. Thus, we introduce Σοφια algorithm, which automatically adjusts threshold θ ensuring that $\max\limits_{0<i\leq k} |\mathcal{P}_i| < L$. Here L can be considered as a constraint on the memory used by the algorithm. It can be seen from Eq. (2) that Σοφια algorithm has polynomial time complexity if $\mathcal{M}$ and *Preimages* are polynomial. In the next subsection we consider an important partial case where Σοφια has polynomial complexity.

3.4 Σοφια Algorithm for Binary Data

In this subsection we have a formal context $\mathbb{K} = (G, M, I)$ with $M = \{m_1, \cdots, m_N\}$ and we want to find itemsets $X \subseteq M$ interesting w.r.t. a measure $\mathcal{M}$. First, we instantiate a chain of projections. In the case of binary data it corresponds to the chain of contexts $\mathbb{K}_i = (G, M_i, I \cap G \times M_i)$, where $M_i = \{m_1, \cdots, m_i\}$, i.e., M_i contains the first i attributes from M. It means that $\psi_i(X) = X \cap M_i$.

Then we define how the function *Preimages* works for this kind of chains of projections. A set $X \subseteq M_{i-1}$ has two preimages in the powerset of M_i, i.e. X and $X \cup \{m_i\}$. Hence, the computation complexity of finding preimages for any itemset X is constant. For the projection ψ_0 corresponding to the context $(G, \emptyset, \emptyset)$ there is only one itemset $\emptyset$. Thus, the worst case complexity for ϑ-Σοφια algorithm is

$$\mathbb{T}(\vartheta\text{-}\Sigma o\varphi\iota\alpha_{\text{binary}}) = |M| \cdot \max_{0<i\leq N} |\mathcal{P}_i| \cdot \mathbb{T}(\mathcal{M}). \tag{3}$$

In particular, the complexity of Σοφια for binary data is $|M| \cdot L \cdot \mathbb{T}(\mathcal{M})$, i.e., it is polynomial modulo complexity of the measure.

3.5 Σοφια Algorithm for Closed Patterns

Closed frequent itemsets are widely used as a condensed representation of all frequent itemsets since [10]. Here we show how one can adapt our algorithm for closed patterns. A closed pattern in $\psi_{i-1}(\mathbb{P})$ is not necessarily closed in $\psi_i(\mathbb{P})$. However, the extents of $\psi(\mathbb{P})$ are extents of $\mathbb{P}$ [7]. Thus, we associate the closed patterns with extents, and then work with extents instead of patterns, i.e., a pattern structure $\mathbb{P} = (G, (D, \sqcap), \delta)$ is transformed into $\mathbb{P}_C = (G, (D_C, \sqcap_C), \delta_C)$, where $D_C = 2^G$. Moreover, for all $x, y \in D_C$ we have $x \sqcap_C y = (x^\diamond \sqcap y^\diamond)^\diamond$, where diamond operator is computed in $\mathbb{P}$ and $\delta_C(g \in G) = \{g\}$. Hence, every pattern p in D_C corresponds to a closed pattern $p^\diamond$ in D.

A projection ψ of $\mathbb{P}$ induces a projection ψ_C of $\mathbb{P}_C$, given by $\psi_C(X \subseteq G) = \psi(X^\diamond)^\diamond$, where diamond is computed in $\mathbb{P}$. The function ψ_C is a projection because of the properties of $(\cdot)^\diamond$ operators and ψ mappings. We use this approach for representing closed patterns in our computer experiments.

4 Experiments and Discussion

Datasets	Decreasing order									Increasing order									Random order					
	$L = 10^3$			$L = 10^4$			$L = 10^5$			$L = 10^3$			$L = 10^4$			$L = 10^5$			$L = 10^3$			$L = 10^4$			$L = 10^5$		
	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ
Mushrooms	< 1	0.99	181	2	0.87	49	39	0.89	7	1	0.99	181	6	0.87	49	38	0.89	7	< 1	0.99	181	3	0.87	49	117	0.89	7
Chess	< 1	0.997	97	2	0.92	69	17	0.94	46	1	0.88	144	4	0.24	84	38	0.68	49	< 1	0.65	103	2	0.92	69	19	0.94	46
Plants	1	1	147	14	0.96	70	146	0.94	37	3	1	147	29	0.96	70	263	0.94	37	1	1	147	14	0.96	70	143	0.94	37
Cars	< 1	0.94	19	< 1	0.61	0	< 1	0.06	0	< 1	0.86	22	< 1	0.61	0	< 1	0.06	0	< 1	0.94	19	< 1	0.6	0	< 1	0.06	0

Table 1: Evaluation results of Σοφια algorithm for Δ-measure.

In the experiment we show how our algorithm in conjunction with stability estimate behaves on different datasets from UCI repository [11]. Here we should note that stability and its estimate is monotone w.r.t. any projection [12] and,

thus, we can combine it with Σοφια. The datasets `Mushrooms`[1] and `Cars`[2] are datasets having a relatively small number of closed patterns, which can be found in some seconds, while the datasets `Chess`[3] and `Plants`[4] have a lot of closed patterns, which can be hardly found.

There are two obvious orders for adding an attribute in Σοφια algorithm: the decreasing and increasing orders of attribute support. We consider also a random order of attributes allowing one to discard any bias in the order of attributes. Another point about our algorithm is that it does not ensure finding L-top best patterns. It finds no more than L patterns allowing to compute the result in polynomial time by adjusting the threshold θ of stable patterns.

Thus, in our experiment we have checked which order is better for the attributes and how many patterns we can find for a given L. Table 1 shows the results and is divided into three parts corresponding to the order in which attributes were added to the context. Then all parts are divided into three subparts corresponding to a value of $L \in \{10^3, 10^4, 10^5\}$. Hence, we have 9 experiments and for every experiment we measure the computation time in seconds (t), the ratio of found patterns to L (#) and the final θ corresponding to the found patterns. For example, in the `Mushrooms` dataset, adding the attributes in the decreasing order of their support for $L = 10000$, the total computational time is equal to 2 seconds; the algorithm found around $0.87 * L = 8700$ patterns representing all patterns with stability higher than 49.

In Table 1 we can see that our algorithm is efficient in the big and small datasets however the computational time and the number of found patterns depend on the order of attribute addition, i.e., on a projection chain. We can see that the computational time and the number of patterns for increasing order are never better than those of decreasing order and random order. Decreasing order and random order have nearly the same behavior, but in some cases the random order gives slightly worse results than the decreasing order. In fact, in the case of decreasing order we generate more patterns on earlier iterations of our algorithm, i.e., we have more chances to find an unstable pattern and filter it as earlier as possible. Since concepts are filtered earlier, we have more space for the computation, thus having smaller threshold θ and larger number of found patterns, and we should process less patterns, thus saving the computation time. We see that for the decreasing order of attributes the number of found patterns is always around or higher than $0.9 * L$, i.e., we find nearly as many patterns as the requested limit L.

[1] `https://archive.ics.uci.edu/ml/datasets/Mushroom`
[2] `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`
[3] `https://archive.ics.uci.edu/ml/datasets/Chess+(King-Rook+vs.+King-Knight)`
[4] `https://archive.ics.uci.edu/ml/datasets/Mushroom`

5 Conclusion

In this paper we have introduced a new kind of interestingness measures of patterns monotonic w.r.t. a chain of projections. Based on this monotonicity we introduce a new algorithm called Σοφια that finds the top best patterns for such kind of measures in polynomial time. Our experiments justified the efficiency of our algorithms. Many directions for future work are promising. First, we should work on adaptation of Σοφια for finding different kinds of patterns, e.g., itemset generators, sequences, graphs. Second, we should study the best chains of projections and the best order of attributes for Σοφια algorithm. Finally, the study of new measures that can be used with Σοφια is also very important.

References

1. Vreeken, J., Tatti, N.: Interesting Patterns. In Aggarwal, C.C., Han, J., eds.: Freq. Pattern Min. Springer International Publishing (2014) 105–134
2. Kuznetsov, S.O.: On stability of a formal concept. Ann. Math. Artif. Intell. **49**(1-4) (2007) 101–115
3. Webb, G.I., Vreeken, J.: Efficient Discovery of the Most Interesting Associations. ACM Trans. Knowl. Discov. from Data **8**(3) (2014) 15
4. Cao, J., Wu, Z., Wu, J.: Scaling up cosine interesting pattern discovery: A depth-first method. Inf. Sci. (Ny). **266**(0) (2014) 31–46
5. Hébert, C., Crémilleux, B.: Mining Frequent δ-Free Patterns in Large Databases. In Hoffmann, A., Motoda, H., Scheffer, T., eds.: Discov. Sci. Volume 3735 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2005) 124–136
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer (1999)
7. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In Delugach, H.S., Stumme, G., eds.: Concept. Struct. Broadening Base. Volume 2120 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2001) 129–142
8. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Scalable Estimates of Concept Stability. In Sacarea, C., Glodeanu, C.V., Kaytoue, M., eds.: Form. Concept Anal. Volume 8478 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014) 161–176
9. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Revisiting Pattern Structure Projections. In Baixeries, J., Sacarea, C., Ojeda-Aciego, M., eds.: Form. Concept Anal. Volume 9113 of LNAI 9113. Springer International Publishing (2015) 200–215
10. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient Mining of Association Rules Using Closed Itemset Lattices. Inf. Syst. **24**(1) (1999) 25–46
11. Frank, A., Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, Irvine, School of Information and Computer Sciences (2010)
12. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: On Mining Complex Sequential Data by Means of FCA and Pattern Structures. Int. J. Gen. Syst. (2016) IN PRESS

Pattern structures for news clustering

Tatyana Makhalova, Dmitry Ilvovsky, Boris Galitsky

School of Applied Mathematics and Information Science, National Research University Higher School of Economics, Moscow, Russia
Knowledge Trail Incorporated
t.makhalova@gmail.com, dilvovsky@hse.ru, bgalitsky@hotmail.com

Abstract. Usually web search results are represented as long list of document snippets. It is difficult for users to navigate through this collection of text. We propose clustering method that uses pattern structure constructed on augmented syntactic parse trees. In addition, we compare our method to other clustering methods and demonstrate the limitations of the competitive methods.

1 Introduction and related works

Document clustering problem has been widely investigated in many applications of text mining. One of the most important aspects of a text clustering problem is a structured representation of text. The common approach to text representation is the Vector Space Model [1], where the collection or corpus of documents is represented as a term-document matrix. The main drawback of this model is its inability to reflect the importance of words with respect to a document and a corpus. To tackle this issue the weighted scheme based on tf-idf score has been proposed.

However, a term-document matrix built on a large texts collection may be sparse and have high dimensionality. To reduce the feature space one may use PCA, truncated SVD (Latent Semantic Analysis), random projection and other methods. To handle synonyms as similar terms a Generalized Vector Space Model [2, 3], a Topic-based Vector Model [4] and Enhanced Topic-based Vector Space Model [5] were introduced. The most common ways to clustering of a term-document matrix are Hierarchical clustering, k-Means and also Bisecting k-Means.

Graph models are also used for text representation. Document Index Graph (DIG) was proposed by Hammouda [6]. Zamir and Etzioni [7] use suffix tree for representing web snippets, where words are used instead of characters. The more sophisticated model based on n-grams was introduced in [8].

In this paper, we consider a particular application of document clustering: representation of web search results that could make it easier for users to find the information they are looking for [9]. Clustering snippets on salient phrases (i.e. key phrases that characterize a cluster) are described in [10, 11]. But the most promising approach for document clustering is conceptual clustering, because it allows to obtain overlapping clusters and to organize them into a hierarchical

structure as well [12–17]. We present an approach to select the most significant clusters based on pattern structures [18]. This approach was introduced in [19]. The main idea is to construct a hierarchical structure of clusters using a reduced representation of syntactic trees with discourse relations between them. Leveraging discourse information allows to combine news articles not only by keyword similarity but by broader topicality and writing styles as well.

2 Clustering based on pattern structure

Parse Thickets Parse thicket [19] is defined as a set of parse trees for each sentence augmented with a number of arcs, reflecting inter-sentence relations. In this work we use parse thickets based on a limited set of relations: coreferences [20], Rhetoric structure relations [21] and Communicative Actions [22]. More information could be found in [19].

FCA A formal context is a triple (G, M, I), where G and M be sets, called the set of objects and attributes, respectively. Let I be a relation $I \subseteq G \times M$ between objects and attributes, i.e. $(g, m) \in I$ if the object g has the attribute m. The derivation operator $(\cdot)^{'}$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

$$A^{'} = \{m \in M | \forall g \in A : gIm\}$$

$$B^{'} = \{g \in G | \forall m \in B : gIm\}$$

$A^{'}$ is the set of attributes common to all objects of A and $B^{'}$ is the set of objects sharing all attributes of B. The double application of $(\cdot)^{'}$ is a closure operator, i.e., $(\cdot)^{''}$ is extensive, idempotent and monotone. Sets $(A)^{''}$ and $(B)^{''}$ are said to be closed. A formal concept is a pair (A, B), where $A \subseteq G$, $B \subseteq M$ and $A^{'} = B$, $B^{'} = A$. A and B are called the formal extent and the formal intent, respectively.

Pattern Structure and Projections Pattern Structures are generalization of formal contexts, where objects are described by more complex structures, rather than a binary data. A pattern structure [18] is defined as a triple $(G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a complete meet-semilattice of descriptions and $\delta : G \to D$ is a mapping an object to a description. The Galois connections between set of objects and their descriptions are defined as follows:

$$A^{\square} := \sqcap_{g \in A} \delta(g) \text{ for } A \subseteq G$$

$$d^{\square} := \{g \in G | d \sqsubseteq \delta(g)\} \text{ for } d \in D$$

A pair (A, d) for which $A^{\square} = d$ and $d^{\square} = A$ is called a pattern concept.

A projection ψ is a kernel operator, i.e. it is monotone ($x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$), contractive ($\psi(x) \sqsubseteq x$), and idempotent ($\psi(\psi(x)) = \psi(x)$). The mapping $\psi : D \to D$ is used to replace $(G, (D, \sqcap), \delta)$ by $(G, (D_\psi, \sqcap_\psi), \psi \circ \delta)$, where $D_\psi = \{d \in D | \exists d' \in D : \psi(d') = d\}$.

In our case, *an original paragraph of text* and *parse thickets constructed from this paragraph* correspond to *an object* and *a description of pattern concepts* respectively. To improve efficiency and decrease time complexity we use projection instead of a parse thicket itself. Projection on a parse thicket is defined as a set of its maximal sub-trees and the intersection operator takes the form of pairwise intersection of elements within noun and verb phrase groups.

3 Reduced pattern structures

A pattern structure constructed from the collection of short texts usually has a huge number of concepts. To reduce the computational costs and improve the interpretability of pattern concepts we introduce several metrics that are described below.

Average and Maximal Pattern Score The average and maximal pattern score indices are meant to assess how meaningful is the common description of texts in the concept. The higher the difference of text fragments from each other, the lower their shared content is. Thus, meaningfulness criterion of a pattern concept $\langle A, d\rangle$ is

$$Score^{max}\langle A, d\rangle := \max_{chunk\in d} Score\,(chunk)$$

$$Score^{avg}\langle A, d\rangle := \frac{1}{|d|}\sum_{chunk\in d} Score\,(chunk)$$

The score function $Score\,(chunk)$ estimates description d using its weights for different parts of speech.

Average and Minimal Pattern Loss Score This scores estimate how much information contained in the description of a text is lost with respect to the original text. The average pattern loss score calculates the average loss of a cluster content with respect to texts in this cluster, while minimal pattern score loss represents a minimal loss of content among all texts included in a concept.

$$ScoreLoss^{min}\langle A, d\rangle := 1 - \frac{Score^{max}\langle A, d\rangle}{\min_{g\in A} Score^{max}\langle g, d_g\rangle}$$

$$ScoreLoss^{avg}\langle A, d\rangle := 1 - \frac{Score^{avg}\langle A, d\rangle}{\frac{1}{|d|}\sum_{g\in A} Score^{max}\langle g, d_g\rangle}$$

We use a reduced pattern structure. We propose to create exactly meaningful pattern concepts. For arbitrary sets of texts A_1 and A_2, corresponding descriptions d_1, d_2 and candidate for a pattern concept $\langle A_1 \cup A_2\ , d_1 \cap d_2\rangle$ need to satisfy the following constrains

$$ScoreLoss^{*}\langle A_1 \cup A_2\ , d_1 \cap d_2\rangle \leq \theta$$

$$Score^{*}\langle A_1 \cup A_2\ , d_1 \cap d_2\rangle \geq \mu_1 \min\{Score^{*}\langle A_1\ , d_1\rangle, Score^{*}\langle A_2\ , d_2\rangle\}$$

$$Score^* \langle A_1 \cup A_2 \, , d_1 \cap d_2 \rangle \leq \mu_2 \max \{Score^* \langle A_1 \, , d_1 \rangle, Score^* \langle A_2 \, , d_2 \rangle\}$$

The first constraint provides condition for the construction of concepts with meaningful content, while two other constrains ensure that we do not use concepts with similar content.

4 Experiments

In this section we consider two examples for the proposed clustering method. The first one corresponds to the case when clusters are overlapping and distinguishable, the second one is the case of non-overlapping clusters.

4.1 User Study

In the most cases it is quite difficult to identify disjoint classes for a text collection [23]. To confirm this, we conducted experiments similar to the experiment scheme described in [11]. We took web snippets obtained by querying the Bing search engine API and asked a group of four experts to label ground truth for them. We performed news queries related to world's most pressing news (for example, "fighting Ebola with nanoparticles", "turning brown eyes blue", "F1 winners", "read facial expressions through webcam", "2015 ACM awards winners") to make labeling of data easier for the experts.

According to the experts, it was difficult to determine partitions, while overlapping clusters naturally stood out. As a result, in the case of non-overlapping clusters we usually got a small number of large classes or a sufficiently large number of classes consisting of 1-2 snippets. More than that, for the same set of snippets we obtained quite different partitions.

We used the Adjusted Mutual Information score to estimate pairwise agreement of non-overlapping clusters, which were identified by the experts. This metric allows one to estimate agreement of two clustering results with correction for randomness partition.

$$MI_{adj} = \frac{MI\,(U,V) - E\,[MI\,(U,V)]}{max\,(H(U), H(V)) - E\,[MI\,(U,V)]}$$

where U and V are partitions of the news set, $MI(U,V)$ - the mutual information between them and $E\,[MI\,(U,V)]$ is the expected mutual information between two random clusterings.

To study the behavior of the conventional clustering approach we consider 12 short texts on news query "The Ebola epidemic". Tests are available by link [1].

Experts identify quite different non-overlapping clusters. The pairwise Adjusted Mutual Information score was in the range of 0,03 to 0,51. Next, we

[1] `https://drive.google.com/file/d/0B7I9HM34b_62TEFtUTRqdzdqWjA/view?usp=sharing`

compared partitions to clustering results of the following clustering methods: k-means clustering based on vectors obtained by truncated SVD (retaining at least 80% of the information), hierarchical agglomerative clustering (HAC), complete and average linkage of the term-document matrix with Manhattan distance and cosine similarity, hierarchical agglomerative clustering (both linkage) of tf-idf matrix with Euclidean metric. In other words, we turned an unsupervised learning problem into the supervised one. The accuracy score for different clustering methods is represented in Figure 1. Curves correspond to the different partitions that have been identified by people.

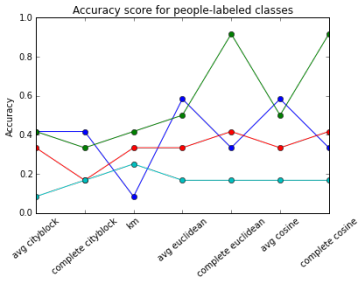


Fig. 1: Classification accuracy of clustering results and "true" clustering (example 1). Four lines are different news labeling made by people. The y-axis values for fixed x-value correspond to classification accuracy of a clustering method for each of the four labeling

As it was mentioned earlier, we obtain inconsistent "true" labeling. Thereby the accuracy of clustering differs from labeling made by evaluators. This approach doesn't allow to determine the best partition, because a partition itself is not natural for the given news set. For example, consider clusters obtained by HAC based on cosine similarity (trade-off between high accuracy and its low variation): 1-st cluster: 1,2,7,9; 2-nd cluster: 3,11,12; 3-rd cluster: 4,8; 4-th cluster: 5,6; 5-th cluster: 10.

Almost the same news 4, 8, 12 and 9, 10 are in the different clusters. News 10, 11 should be simultaneously in several clusters (1-st, 5-th and 2-nd,3-rd respectively).

4.2 Examples of pattern structures clustering

To construct hierarchy of overlapping clusters by the proposed methods, we use the following constraints: $\theta = 0,75$, $\mu_1 = 0,1$ and $\mu_2 = 0,9$. The value of θ limits the depth of the pattern structure (the maximal number of texts in a cluster), put differently, the higher θ, the closer should be the general intent of clusters. μ_1 and μ_2 determine the degree of dissimilarity of the clusters on different levels of the lattice (the clusters are prepared by adding a new document to the current one).

We consider the proposed clustering method on 2 examples. The first one was described above, it corresponds to the case of overlapping clusters, the second

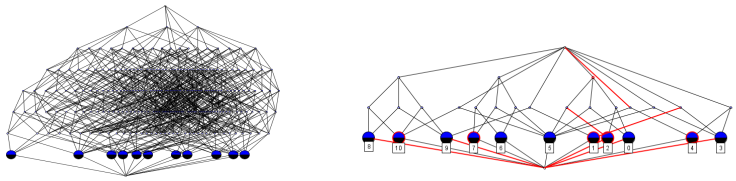
one is the case when clusters are non-overlapping and distinguishable. Texts of the second example are available by link [2]. Three clusters are naturally identified in this texts.

The cluster distribution depending on volume are shown in Table 1. We got 107 and 29 clusters for the first and the second example respectively.

Text number	1	2	3	4	5	6
Example 1	12	34	33	20	7	1
Example 2	11	15	3	0	0	0

Table 1: The clusters volume distribution for non-overlapping clusters (example 1) and overlapping clusters (example 2)

In fact, this method is an agglomerative hierarchical clustering with overlapping clusters. Hierarchical structure of clusters provides browsing of texts with similar content by layers. The cluster structure is represented on Figure 2. The top of the structure corresponds to meaningless clusters that consist of all texts. Upper layer consists of clusters with large volume.



(a) pattern structure without reduction

(b) reduced pattern structure

Fig. 2: The cluster structure (example 2). The node on the top corresponds to the "dummy" cluster, high level nodes correspond to the big clusters with quite general content, while the clusters at lower levels correspond to more specific news.

Clustering based on pattern structures provides well interpretable groups. The upper level of hierarchy (the most representative clusters for example 1) consists of the clusters presented in Table 2.

MaxScore	Cluster (extent)	MaxScore	Cluster (extent)	MaxScore	Cluster (extent)
7,8	{3, 11, 12}	3,8	{1, 2, 3, 7, 9}	3,2	{3, 9, 11}
4,1	{4, 8, 11}	3,3	{2, 4, 11}	2,8	{3, 10}
3,8	{1, 5, 11}	3,3	{2, 11}	2,4	{1, 2, 6, 9, 10}
3,8	{1, 11}	3,3	{5, 6}	2,3	{1, 5, 6}

Table 2: Scores of representative clusters

We also consider smaller clusters and select those for which adding of any object (text) dramatically reduces the $MaxScore$ $\{1,2,3,7,9\}$ and $\{5,6\}$. For

[2] `https://drive.google.com/file/d/0B7I9HM34b_62czFlZ29zZl9kblk/view?usp=sharing`

other nested clusters significant decrease of $MaxScore$ occurred exactly with the an expansion of single clusters.

For the second example we obtained 3 clusters that corresponds to "true" labeling.

Our experiments show that pattern structure clustering allows to identify easily interpretable groups of texts and significantly improves text browsing.

5 Conclusion

In this paper, we presented an approach that addressed the problem of short text clustering. Our study shows a failure of the traditional clustering methods, such as k-means and HAC. We propose to use parse thickets that retain the structure of sentences instead of the term-document matrix and to build the reduced pattern structures to obtain overlapping groups of texts. Experimental results demonstrate considerable improvement of browsing and navigation through a texts set for users. Introduced indices $Score$ and $ScoreLoss$ both improve computing efficiency and tackle the problem of redundant clusters.

An important direction for future work is to take into account synonymy and to compare the proposed method to similar approach that use key words instead of parse thickets.

Acknowledgments

The project is being developed by the "Methods of web corpus collection, analysis and visualisation" research and study group under guidance of prof. B.Mirkin (grant 15 - 05 - 0041 of Academic Fund Program).

References

1. Salton, G., Wong, A., Yang, C.S.: A vector space model for automatic indexing. Communications of the ACM **18** (1975) 613–620
2. Wong, S.M., Ziarko, W., Wong, P.C.: Generalized vector spaces model in information retrieval. In: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (1985) 18–25
3. Tsatsaronis, G., Panagiotopoulou, V.: A generalized vector space model for text retrieval based on semantic relatedness. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, Association for Computational Linguistics (2009) 70–78
4. Becker, J., Kuropka, D.: Topic-based vector space model. In: Proceedings of the 6th International Conference on Business Information Systems. (2003) 7–12
5. Polyvyanyy, A., Kuropka, D.: A quantitative evaluation of the enhanced topic-based vector space model. (2007)
6. Hammouda, K.M., Kamel, M.S.: Document similarity using a phrase indexing graph model. Knowledge and Information Systems **6** (2004) 710–727

7. Zamir, O., Etzioni, O.: Web document clustering: A feasibility demonstration. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval, ACM (1998) 46–54
8. Schenker, A., Bunke, H., Last, M., Kandel, A.: Clustering of web documents using graph representations. In: Applied Graph Theory in Computer Vision and Pattern Recognition. Springer (2007) 247–265
9. Galitsky, B.: Natural language question answering system: Technique of semantic headers. Advanced Knowledge International (2003)
10. Zamir, O., Etzioni, O.: Grouper: a dynamic clustering interface to web search results. Computer Networks **31** (1999) 1361–1374
11. Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: Learning to cluster web search results. In: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM (2004) 210–217
12. Galitsky, B., Ilvovsky, D., Kuznetsov, S., Strok, F.: Finding maximal common subparse thickets for multi-sentence search. In Croitoru, M., Rudolph, S., Woltran, S., Gonzales, C., eds.: Graph Structures for Knowledge Representation and Reasoning. Volume 8323 of Lecture Notes in Computer Science. Springer International Publishing (2014) 39–57
13. Cole, R., Eklund, P., Stumme, G.: Document retrieval for e-mail search and discovery using formal concept analysis. Applied artificial intelligence **17** (2003) 257–280
14. Koester, B.: Conceptual knowledge retrieval with fooca: Improving web search engine results with contexts and concept hierarchies. In: Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining. Springer (2006) 176–190
15. Messai, N., Devignes, M.D., Napoli, A., Smail-Tabbone, M.: Many-valued concept lattices for conceptual clustering and information retrieval. In: ECAI. Volume 178. (2008) 127–131
16. Carpineto, C., Romano, G.: A lattice conceptual clustering system and its application to browsing retrieval. Machine Learning **24** (1996) 95–122
17. Strok, F., Galitsky, B., Ilvovsky, D., Kuznetsov, S.: Pattern structure projections for learning discourse structures. In Agre, G., Hitzler, P., Krisnadhi, A., Kuznetsov, S., eds.: Artificial Intelligence: Methodology, Systems, and Applications. Volume 8722 of Lecture Notes in Computer Science. Springer International Publishing (2014) 254–260
18. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Conceptual Structures: Broadening the Base. Springer (2001) 129–142
19. Galitsky, B., Ilvovsky, D., Kuznetsov, S., Strok, F.: Matching sets of parse trees for answering multi-sentence questions. Proc. Recent Advances in Natural Language Processing (RANLP 2013), Bulgaria (2013)
20. Lee, H., Recasens, M., Chang, A., Surdeanu, M., Jurafsky, D.: Joint entity and event coreference resolution across documents. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics (2012) 489–500
21. Mann, W.C., Thompson, S.A.: Discourse description: Diverse linguistic analyses of a fund-raising text. Volume 16. John Benjamins Publishing (1992)
22. Searle, J.R.: Speech acts : an essay in the philosophy of language. Cambridge University Press (1969)
23. Galitsky, B., de la Rosa, J.L.: Concept-based learning of human behavior for customer relationship management. Information Sciences **181** (2011) 2016–2035

Lazy Classification with Interval Pattern Structures: Application to Credit Scoring

Alexey Masyutin, Yury Kashnitsky, and Sergei Kuznetsov

National Research University Higher School of Economics
Scientific-Educational Laboratory for Intelligent Systems and Structural Analysis
Moscow, Russia
alexey.masyutin@gmail.com,ykashnitsky@hse.ru,skuznetsov@hse.ru

Abstract. Pattern structures allow one to approach the knowledge extraction problem in case of arbitrary object descriptions. They provide the way to apply Formal Concept Analysis (FCA) techniques to non-binary contexts. However, in order to produce classification rules a concept lattice should be built. For non-binary contexts this procedure may take much time and resources. In order to tackle this problem, we introduce a modification of the lazy associative classification algorithm and apply it to credit scoring. The resulting quality of classification is compared to existing methods adopted in bank systems.

1 Introduction

Banks and credit institutions face classification problem each time they consider a loan application. In the most general case, a bank aims to have a tool to discriminate between solvent and potentially delinquent borrowers, i.e. the tool to predict whether the applicant is going to meet his or her obligations or not. Before 1950s such a decision was expert driven and involved no explicit statistical modeling. The decision whether to grant a loan or not was made upon an interview and after retrieving information about spouse and close relatives [4]. From the 1960s, banks have started to adopt statistical scoring systems that were trained on datasets of applicants, consisting of their socio-demographic factors and loan application features. As far as mathematical models are concerned, they were typically logistic regressions run on selected set of attributes. Apparently, a considerable amount of research was done in the field of alternative machine learning techniques seeking the goal to improve the results of the wide-spread scorecards [7,8,9,10,11].

All mentioned methods can be divided into two groups: the first one provides the result difficult for interpretation, so-called "black box" models, the second group provides interpretable results and clear model structure. The key feature of risk management practice is that, regardless of the model accuracy, it must not be the black box. That is why methods such as neural networks and SVM

classifiers did not earn much trust within the banking community [4]. The dividing hyperplane in an artificial high-dimensional space (dependent on the chosen kernel) cannot be easily interpreted in order to claim the reject reason for the client. As far as neural networks are concerned, they also do not provide the user with a set of reasons why a particular loan application has been approved or rejected. In other words, these algorithms do not provide the decision maker with knowledge. The predicted class is generated, but no knowledge is retrieved from data.

On the contrary, alternative methods such as association rules and decision trees provide the user with easily interpretable rules which can be applied to the loan application. FCA-based algorithms also belong to the second group since they use concepts in order to classify objects. The intent of the concept can be interpreted as a set of rules that is supported by the extent of the concept. However, for non-binary context the computation of the concepts and their relations can be very time-consuming. In case of credit scoring we deal with numerical context, as soon as categorical variables can be transformed into a set of dummy variables. Lazy classification [16] seems to be appropriate to use in this case since it provides the decision maker with the set of rules for the loan application and can be easily parallelized. In this paper, we modify the lazy classification framework and test it on credit scoring data of a top-10 Russian bank.

The paper is structured as follows: section 2 provides basic definitions. Section 3 argues why the original setting can be inconsistent in case of a large numerical context and describes the proposed modification and its parameters. Section 4 describes voting schemes that can be used to classify test objects. Section 5 describes the data in hand and some experiments with parameters of the algorithm. Finally, section 6 concludes the paper.

2 Main Definitions

First, we recall some standard definitions related to Formal Concept Analysis, see e.g. [1,2].

Let G be a set (of objects), let (D, $\sqcap$) be a meet-semi-lattice (of all possible object descriptions) and let δ: G $\to$ D be a mapping. Then (G, $\underline{D}$,δ), where $\underline{D}$ =(D, $\sqcap$), is called a *pattern structure* [1], provided that the set δ(G) := $\{\delta(g)|g \in G\}$ generates a complete subsemilattice (D_δ, $\sqcap$) of (D, $\sqcap$), i.e., every subset X of $\delta(G)$ has an infimum $\sqcap$X in (D, $\sqcap$). Elements of D are called *patterns* and are naturally ordered by *subsumption* relation $\sqsubseteq$:
given c, d $\in$ D one has c $\sqsubseteq$ d $\leftrightarrow$ c $\sqcap$ d = c. Operation $\sqcap$ is also called a *similarity operation*. A pattern structure (G, $\underline{D}$, δ) gives rise to the following *derivation operators* $(\cdot)^\diamond$:

$$A^\diamond = \prod_{g \in A} \delta(g) \qquad \text{for } A \in G,$$
$$d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D, \sqcap).$$

These operators form a Galois connection between the powerset of G and $(D, \sqcap)$. The pairs (A, d) satisfying A $\subseteq$ G, d $\in$ D, $A^\diamond$ = d, and A = $d^\diamond$ are called *pattern concepts* of (G,$\underline{D}$, δ), with *pattern extent* A and *pattern intent* d. Operator $(\cdot)^{\diamond\diamond}$ is an algebraical closure operator on patterns, since it is idempotent, extensive, and monotone [1].

The concept-based learning model for standard object-attribute representation (i.e., formal contexts) is naturally extended to pattern structures. Suppose we have a set of positive examples G_+ and a set of negative examples G_- w.r.t. a target attribute, $G_+ \cap G_- = \emptyset$, objects from
G_τ = G $\setminus (G_+ \cup G_-)$ are called undetermined examples. A pattern c $\in$ D is an α - weak positive premise (classifier) iff:

$$\frac{||c^\diamond \cap G_-||}{||G_-||} \leq \alpha \; and \; \exists A \subseteq G_+ : c \sqsubseteq A^\diamond$$

A pattern h $\in$ D is an α - weak positive hypothesis iff:

$$\frac{||h^\diamond \cap G_-||}{||G_-||} \leq \alpha \; and \; \exists A \subseteq G_+ : h = A^\diamond$$

In case of credit scoring we work with pattern structures on intervals as soon as a typical object-attribute data table is not binary, but has many-valued attributes. Instead of binarizing (scaling) data, one can directly work with many-valued attributes by applying interval pattern structure. For two intervals $[a_1, b_1]$ and $[a_2, b_2]$, with $a_1, b_1, a_2, b_2 \in \mathbb{R}$ the *meet operation* is defined as [15]:

$[a_1, b_1] \sqcap [a_2, b_2] = [min(a_1, a_2), max(b_1, b_2)]$.

The original setting for lazy classification with pattern structures can be found in [3].

3 Modification of lazy classification algorithm

In credit scoring the object-attribute context is typically numerical. Factors can have arbitrary distributions and take wide range of values. At the same time categorical variables and dummies can be present. With relatively large number of attributes (over 30-40) it produces high-dimensional space of continuous variables. That is when the result of the meet operator tends to be very specific, i.e. for almost every $g \in G$ only g and g_n have the descriptin $\delta(g_n) \sqcap \delta(g)$. This happens due to the fact that numerical variables, ratios especially, can have unique values for every object. This results in that for test object g_n the number of positive and negative premises is close to the number of observations in those context correspondingly. In other words, too specific descriptions are usually not falsified (i.e. there are no objects of opposite class with such description) and almost always form either positive or negative premises. Therefore, the idea of voting scheme for lazy classification in the case of high dimensional numerical context may turn out to be obscure. Thus, it seems reasonable to seek the concepts with larger extent and with not too specific intent. At the same we would

like to preserve the advantages of lazy classification, e.g. no need to compute a full concept lattice, easy parallelization etc. The way to increase the extent of the generated concepts is to consider intersection of the test object with more than one element from the positive (negative) context. What is the suitable number of objects to take for intersection? In our modification we consider this as a parameter subsample size and perform grid search. The parameter is expressed as percentage of the observations in the context. As subsample size grows, the resulting intersection $\delta(g_1) \sqcap \ldots \sqcap \delta(g_k) \sqcap \delta(g)$ becomes more generic and it is more frequently falsified by the objects from the opposite context. Strictly speaking, in order to replicate the lazy classification approach, one should consider all possible combinations of the chosen number of objects from the positive (negative) context. Apparently, this is not applicable in the case of large datasets. For example, having 10 000 objects in positive context and having subsample size equal to only two objects will produce almost 50 mln combinations for intersection with the test object. Therefore, we randomly take the chosen number of objects from positive (negative) context as candidates for intersection with the test object. The number of times (number of iterations) we randomly pick a subsample from the context is also tuned through grid search. Intuition says , the higher the value of the parameter the more premises are mined from the data. However, the obvious penalty for increasing the value of this parameter is time and resources required for computing intersections. As mentioned before, the greater the subsample size, the more it is likely that $(\delta(g_1) \sqcap \ldots \sqcap \delta(g_k) \sqcap \delta(g))^{\diamond}$ contains the object of the opposite class. In order to control this issue, we add a third parameter which is alpha-threshold. If the percentage of objects from the positive (negative) context that falsify the premise $\delta(g_1) \sqcap \ldots \sqcap \delta(g_k) \sqcap \delta(g)$ is greater than alpha-threshold of this context then the premise will be considered as falsified, otherwise the premise will be supported and used in the classification of the test object.

4 Voting schemes

The final classification of a test object is based on a voting scheme among premises. In most general case voting scheme F is a mapping:

$$F(g_{test}, h_1^+, ..., h_p^+, h_1^-, ..., h_n^-) \to [-1, 1, \emptyset]$$

where g_{test} is the test object with unknown class, h_i^+ is a positive premise $\forall i = \overline{1,p}$ and h_j^- is a negative premise $\forall j = \overline{1,n}$, -1 is a label for negative class, and 1 is a label for positive class (i.e. defaulters). In other words, F is an aggregating rule that takes premises as input and gives the classification label as an output. Note that we allow for an empty label. If the label is empty it is said that the voting rule abstains from classification. There may be different approaches to build up aggregating rules. The voting scheme is built upon weighting function $\omega(\cdot)$, aggregation operator $A(\cdot)$ and comparing operator $\otimes$.

$$\begin{aligned} &F(\omega(\cdot), A(\cdot), \otimes) = \\ &= (A_{i=1}^{p}[\omega(h_i^+)]) \otimes (A_{j=1}^{n}[\omega(h_j^-)]) \end{aligned}$$

In order to configure a new weighting scheme it is sufficient to define the operators and the weighting function. In this paper we use the number of positive versus negative premises. In this case the rule allows the test object to satisfy both positive and negative premises which decreases the rejection from classification. The weighting function, aggregation operator and comparing operator are defined as follows:

$$A(h) = \sum h$$

$$\omega(h) = \begin{cases} 1, & if\ \delta(g_{test}) \sqsubseteq h \\ 0, & otherwise \end{cases}$$

$$a \otimes b = \begin{cases} sign(b-a), & if\ a \neq b \\ \emptyset, & a = b \end{cases}$$

So the label for a test object g_n is defined by the following mapping:

$$F(g_{test}, h_1^+, ..., h_p^+, h_1^-, ..., h_n^-) =$$
$$= (\sum_{i=1}^{p} [\delta(g_{test}) \sqsubseteq h_i^+]) \otimes (\sum_{j=1}^{n} [\delta(g_{test}) \sqsubseteq h_j^-])$$

However, one can think of margin $b - a$ as a measure for discrimination between two classes and consider the decision boundary based on receiver operating characteristic analysis, for instance. This approach is good for decreasing the number of rejects from classification, but it does not account for the support of the premises. Naturally, one would give more weight to the premise with large image (with higher support). Also, if the number of positive and negative premises is equal the rule rejects from classification.

5 Experiments

The data we used for the computation represent the customers and their metrics assessed on the date of loan application. The applications were approved by the bank credit policy and the clients were granted the loans. After that the loans were observed for the fact of delinquency. The dataset is divided into two contexts positive and negative. The positive context is the set of loans where the target attribute is present. The target attribute in credit scoring is typically defined as more than 90 days of delinquency within the first 12 months after the loan origination. So, the positive context is the set of bad borrowers, and the negative context consists of good ones. Each context consists of 1000 objects in order that voting scheme concerned in the second section was applicable. The test dataset consists of 300 objects and is extracted from the same population as the positive and negative contexts. Attributes represent various metrics such as loan amount, term, rate, payment-to-income ratio, age of the borrower, undocumented-to-documented income, credit history metrics etc. The

set of attributes used for the lazy classification trials contained 28 numerical attributes. In order to evaluate the accuracy of the classification we calculate the Gini coefficient for every combination of parameters based on 300 predictions on the test set. Gini coefficient is calculated based on the margin between the number of objects within positive premises and negative ones. In fact, the margin is the analog for the score value in credit scorecards. Gini coefficient was chosen as performance metric because it is conventionally used to evaluate the quality of classification models in credit scoring [4]. When the subsample size is low, the intersections of the test object description and the members of positive (negative) context tend to be more specific. That is why, a relatively high number of premises are mined and used for the classification. As subsample size increases, the candidates for premises start being generic and it is likely that there exists certain amount of objects from the opposite context which also satisfy the description. If alpha-threshold is low, the frequency of rejects from classification is high. The dynamics of premise mining is demonstrated on the following graphs:

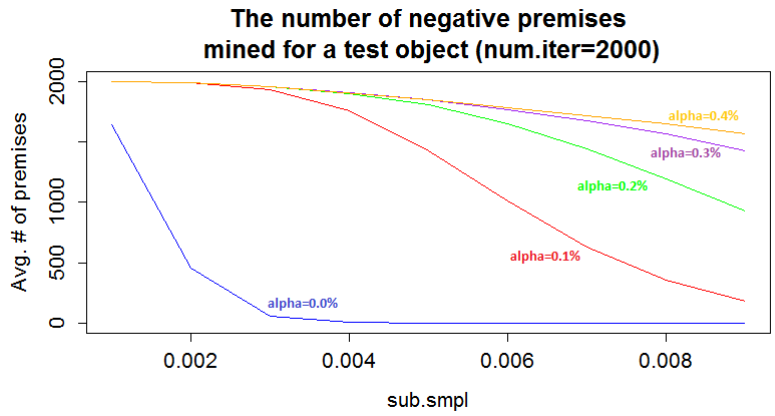


Fig. 1. The dynamics of negative α - weak premises mining

The average number of premises mined for a test object is dropping as expected with the increase in the subsample size and the drop is quicker for higher alpha-thresholds. This supports the idea, that if lazy classification is run in its original setting upon the numerical context (i.e. when subsample size consists of only one object) the number of premises generated is close to the number of objects in the context, so the premises can be considered as too specific. The descriptive graph above allows one to expect that the proposed parameters of the algorithm can be tuned (grid searched), so as to tackle the trade-off between the high number of premises used for classification and the size of their support. The average number of positive premises tends to fall slightly faster compared to negative premises. Below we present the classification accuracy obtained for different combinations of parameters (grid search).

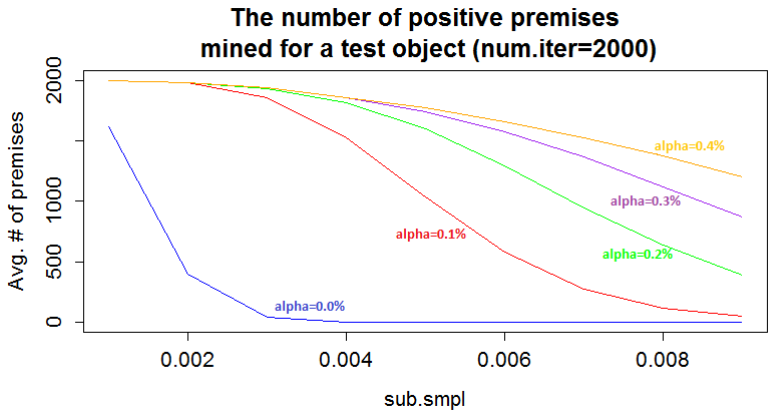


Fig. 2. The dynamics of α - weak positive premises mining

Table 1. Gini coefficients for the parameters grid search

		Subsample size								
Alpha-threshold	Number of iterations	0.1%	0.2%	0.3%	0.4%	0.5%	0.6%	0.7%	0.8%	0.9%
0.0%	100	40%	44%	39%	18%	1%	0%	0%	0%	0%
	150	35%	46%	35%	5%	0%	0%	0%	0%	0%
	200	42%	37%	36%	12%	5%	1%	0%	0%	0%
	500	39%	44%	44%	25%	6%	1%	0%	0%	0%
	1000	44%	47%	44%	41%	11%	3%	0%	0%	0%
	2000	44%	48%	46%	36%	17%	4%	0%	0%	0%
0.1%	100	33%	37%	40%	40%	44%	43%	34%	32%	34%
	150	41%	34%	33%	43%	41%	47%	41%	37%	37%
	200	40%	40%	34%	42%	51%	43%	44%	41%	36%
	500	37%	42%	47%	49%	51%	49%	43%	41%	34%
	1000	37%	42%	46%	48%	49%	48%	43%	43%	37%
	2000	39%	43%	45%	49%	51%	49%	46%	41%	38%
	5000	43%	40%	44%	49%	46%	50%	48%	38%	36%
0.2%	100	29%	38%	42%	32%	43%	37%	46%	43%	37%
	150	27%	42%	41%	41%	36%	47%	48%	45%	41%
	200	32%	40%	43%	42%	42%	49%	46%	47%	48%
	500	39%	46%	46%	48%	47%	48%	51%	48%	51%
	1000	41%	50%	48%	47%	49%	53%	52%	52%	47%
	2000	38%	48%	50%	48%	47%	53%	52%	53%	50%
0.3%	100	35%	38%	39%	42%	39%	45%	34%	45%	39%
	150	27%	43%	44%	42%	42%	39%	37%	40%	46%
	200	34%	46%	47%	45%	49%	47%	45%	45%	52%
	500	31%	45%	49%	50%	49%	46%	50%	51%	47%
	1000	37%	48%	49%	49%	49%	47%	52%	51%	51%
	2000	38%	46%	48%	51%	51%	50%	50%	52%	52%
	5000	40%	47%	46%	51%	52%	51%	49%	51%	53%
	10000	40%	44%	43%	46%	46%	48%	50%	52%	54%
	20000	40%	43%	42%	46%	47%	49%	50%	52%	53%
0.4%	100	28%	39%	44%	48%	43%	50%	53%	42%	49%
	150	34%	42%	43%	42%	43%	52%	50%	45%	47%
	200	33%	46%	43%	47%	51%	49%	49%	42%	45%
	500	37%	50%	50%	49%	49%	49%	51%	47%	48%
	1000	40%	48%	50%	50%	51%	52%	50%	48%	50%
	2000	37%	48%	49%	49%	49%	47%	52%	49%	51%
	5000	39%	42%	42%	43%	45%	47%	49%	52%	49%

We observe the area with zero Gini coefficients where the alpha-threshold is zero and the subsample size is relatively high. That is due to the fact that almost no premises were mined during the lazy classification run. It is quite intuitive because as the subsample size grows, the intersection of the subsample with a test object results in a generic description, which is very likely to be falsified at least by one object from the opposite context. In this case the rejection from classification takes place almost for all test objects. The first thing that is quite intuitive is that the more iterations are produced, the higher is the Gini on average:

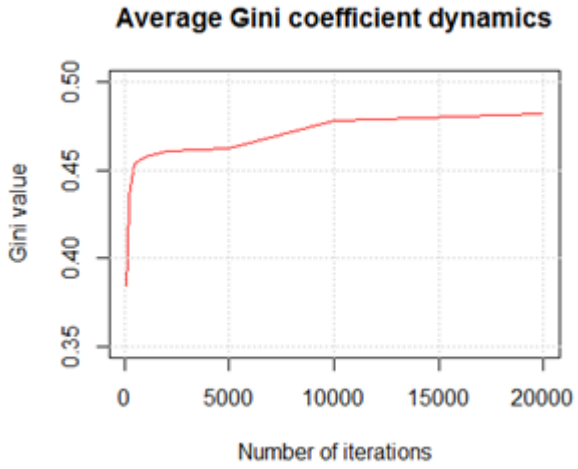


Fig. 3. Average Gini grouped by the different number of iterations (over all other parameter values)

The more times the subsamples are randomly extracted the more knowledge (in terms of premises) is generated. By increasing the number of premises used for classification according to voting scheme, we are likely to capture the structure of the data in more detail. However, the number of iterations is not the only driver of the classification accuracy in our case. We find a range with relatively high Gini in the area of mild alpha-threshold and relatively high subsample size. It also seems natural as soon as the support of a good predictive rule (i.e. premise) is expected to be higher than its support in the opposite context. We elaborate further and run additional grid search in range of parameters providing high Gini coefficient:

Table 2. Gini coefficients for the parameters grid search on specified area

		Subsample size					
Alpha-thresh-old	Number of iterations	1.0%	1.1%	1.2%	1.3%	1.4%	1.5%
0.3%	500	51%	49%	48%	43%	41%	38%
	1000	52%	51%	48%	45%	43%	39%
	2000	54%	53%	49%	47%	46%	38%
	5000	55%	52%	50%	47%	46%	40%
	10000	56%	53%	50%	47%	47%	40%
	20000	55%	53%	51%	46%	48%	41%

According to performed grid search the range with the highest Gini (55%-56%) on the test sample is in range with following parameter values: alpha-threshold = 0,3%, number of iterations = 10000, subsample size = 1,0%. The result was compared to three benchmarks that are traditionally used in the credit scoring within the bank system: logistic regression, scorecard and decision tree. It should be cleared what is implied by the scorecard classifier. Mathematical architecture of the scorecard is based on logistic regression which takes the transformed variables as input. The transformation of the initial variables which is typically used is weight of evidence transformation (WOE-transformation [13]). It is wide-spreaded in credit scoring to apply such a transformation to the input variables as soon as it accounts for non-linear dependencies and it also provides certain robustness coping with potential outliers. The aim of the transformation is to divide each variable into no more than k categories. The thresholds are derived so as to maximize the information value of a variable [13]. Having each variable binned into categories, the log-odds ratio is calculated for each category. Finally, instead of initial variables the discrete valued variables are considered as input in logistic regression. The properties of the decision tree were as follows: we ran CART with two possible child nodes from each parent node. The criterion for optimal threshold calculation was the greatest entropy reduction. The number of terminal nodes was not explicitly restricted; however, the minimum size of the terminal node was set to 50. As far as logistic regression is concerned, the variable selection was performed based on stepwise approach [14]. As for scorecard, the variables were initially selected based on their information value after the WOE-transformation. The comparison of the classifiers performance based on test sample of 300 objects is given in Table 3.

Table 3. Modified lazy classification algorithm versus models adopted in the bank

	Gini on test sample
Logistic regression	47.38%
Scorecard (Logistic based on WOE-transformation)	51.89%
CART (minsize= 50)	54.75%
MLCA (s = 1%, a=0.3%, n=10000)	56.30%

6 Conclusion

When dealing with large numerical datasets, lazy classification may be preferable to classification based on explicitly generated classifiers, since it requires less time and memory resources [3]. However, the original lazy classification setting in case of high dimensional numerical feature space meets certain limitation. The limitation is that, when intersecting descriptions of a test object and every object from the context, one is likely to acquire premises with image consisting only of those two objects. In other words, the premises tend to be very specific for the context and, therefore, the number of positive and negative premises is likely to be equal to the number of the objects in the contexts. The weighting cannot be considered helpful in this case as soon as the premises will have very similar low support. In this paper, we modified the original lazy classification setting by making it, in fact, a stochastic procedure with three parameters: subsample size, number of iterations and alpha-threshold. In effect, the modified algorithm mines the premises with relatively high support that will be used for the classification of the test object. The classification is then carried out upon the predefined voting scheme. We applied the introduced procedure to the retail loan classification problem. The data we used for was provided during the pilot project with one of the top-10 banks in Russia, the details are not provided due to non-disclosure agreement. The positive and negative contexts both had 1000 objects with 28 numerical attributes. The accuracy of the algorithm was evaluated on the test dataset consisting of 300 objects. Gini coefficient was chosen as accuracy metric. We performed the basic grid search by running the modified lazy classification algorithm with different parameter values. The classification accuracy of the algorithm was compared to the conventionally adopted models used in the bank. The benchmark models were logistic regression, scorecard and decision tree. The proposed algorithm outperforms the logistic regression the scorecard with the subsample size parameter around 1%, alpha-threshold equal to 0,3% and with number of iterations over 5000. The performance of the decision tree is at the comparable level with the proposed algorithm, however, the modified lazy classification is slightly better in terms of Gini coefficient. As an area for further research, one can consider and compare accuracy when other voting schemes are used. It is expected that taking into account premises' specificity

one can improve overall accuracy of the classification algorithm or, alternatively, one will reach the same accuracy given less number of iterations, which can save the time resources required for the calculations.

References

1. Bernhard Ganter and Sergei Kuznetsov, "Pattern structures and their projections," in Conceptual Structures: Broadening the Base, Harry Delugach and Gerd Stumme, Eds., vol. 2120 of Lecture Notes in Computer Science, pp. 129–142. Springer, Berlin/Heidelberg, 2001.
2. Ganter, B., Wille, R.: Formal concept analysis: Mathematical foundations. Springer, Berlin, 1999.
3. Sergei O. Kuznetsov, "Scalable knowledge discovery in complex data with pattern structures.," in PReMI, Pradipta Maji, Ashish Ghosh, M. Narasimha Murty, Kuntal Ghosh, and Sankar K. Pal, Eds. 2013, vol. 8251 of Lecture Notes in Computer Science, pp. 30–39, Springer.
4. Thomas L., Edelman D., Crook J. (2002) Credit Scoring and Its Applications, Monographs on Mathematical Modeling and Computation, SIAM: Pliladelphia, pp. 107–117
5. Bigss, D., Ville, B., and Suen, E. (1991). A Method of Choosing Multiway Partitions for Classification and Decision Trees. Journal of Applied Statistics, 18, 1, 49-62.
6. Naeem Siddiqi, Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring, WILEY,ISBN: 978-0-471-75451-0, 2005
7. B Baesens, T Van Gestel, S Viaene, M Stepanova, J Suykens, Benchmarking state-of-the-art classification algorithms for credit scoring, Journal of the Operational Research Society 54 (6), 627-635, 2003
8. Ghodselahi A., A Hybrid Support Vector Machine Ensemble Model for Credit Scoring, International Journal of Computer Applications (0975 – 8887), Volume 17–No.5, March 2011
9. Yu, L., Wang, S. and Lai, K. K. 2009. An intelligent agent-based fuzzy group decision making model for financial multicriteria decision support: the case of credit scoring. European journal of operational research. vol. 195. pp.942-959.
10. Gestel, T. V., Baesens, B., Suykens, J. A., Van den Poel, D., Baestaens, D.-E. and Willekens, B. 2006. Bayesian kernel based classification for financial distress detection. European journal of operational research. vol. 172. pp. 979-1003.
11. P. Ravi Kumar and V. Ravi, "Bankruptcy Prediction in Banks and Firms via Statistical and Intelligent Techniques-A Review," European Journal of Operational Research, Vol. 180, No. 1, 2007, pp. 1-28.
12. Sergei O. Kuznetsov and Mikhail V. Samokhin, "Learning closed sets of labeled graphs for chemical applications.," in ILP, Stefan Kramer and Bernhard Pfahringer, Eds. 2005, vol. 3625 of Lecture Notes in Computer Science, pp. 190– 208, Springer
13. SAS Institute Inc. (2012), Developing Credit Scorecards Using Credit Scoring for SAS® Enterprise Miner™ 12.1, Cary, NC: SAS Institute Inc.
14. Hocking, R. R. (1976) "The Analysis and Selection of Variables in Linear Regression," Biometrics, 32.
15. Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli, and Sebastien Duplessis, "Mining gene expression data with pattern structures in formal concept analysis," Information Sciences, vol. 181, no. 10, pp. 1989–2001, 2011.
16. Veloso, A. & Jr., W. M. (2011), Demand-Driven Associative Classification., Springer.

Algorithm 1 Lazy Classification by Sub-Samples in Numeric Context

Input: $\{Pos_{data}, Neg_{data}\}$ – positive and negative numerical contexts.
N^+, N^- – number of objects in the contexts. It is preferable that the positive and negative contexts are of the same size.
M – number of attributes.
sub.smpl – percentage of the context randomly used for intersection with the test object (parameter).
num.iter – number of iterations (resamplings) during the premise mining (parameter).
alpha.threshold is the maximum allowable percentage of the opposite context for that the premise is not falsified (parameter).
t – test object.

Output: $margin_t$ – measure that is produced by the voting rule.
y_t – class labels predicted for the test object.

for *iter* from 1 to *num.iter* **do**
S=random.sample(Pos_{data},size=$sub.smpl \cdot N^+$) — mine positive α - weak premises
$descr = \delta(g_1) \sqcap ... \sqcap \delta(g_s) \sqcap \delta(t)$
$Neg_{image} = \{x \in descr^{\diamond} | x \in Neg_{data}\}$
if $||Neg_{image}|| < alpha.threshold \cdot N^-$ **then**
Add *descr* to positive α - weak premises set
else
Do nothing
end if
S=random.sample(Neg_{data},size=$sub.smpl \cdot N^-$) — mine α - weak negative premises
$descr = \delta(g_1) \sqcap ... \sqcap \delta(g_s) \sqcap \delta(t)$
$Pos_{image} = \{x \in descr^{\diamond} | x \in Pos_{data}\}$
if $||Pos_{image}|| < alpha.threshold \cdot N^+$ **then**
Add *descr* to negative α - weak premises set
else
Do nothing
end if
end for
$p = dim$(set of positive α - weak premises)
$n = dim$(set of negative α - weak premises)
Choose voting scheme: $A(\cdot), w(\cdot), \otimes$
$pos.power = A_i^p(w(h_i^+))$
$neg.power = A_j^n(w(h_j^-))$
$margin = pos.power - neg.power$
$y_t = pos.power \otimes neg.power$

Reduction in Triadic Data Sets

Sebastian Rudolph[1], Christian Săcărea[2], and Diana Troancă[2]

[1]Technische Universität Dresden
[2]Babeş-Bolyai University Cluj Napoca
sebastian.rudolph@tu.dresden.de, csacarea@cs.ubbcluj.ro,
dianat@cs.ubbcluj.ro

Abstract. Even if not explicitly stated, data can be often interpreted in a triadic setting in numerous scenarios of data analysis and processing. Formal Concept Analysis, as the underlying mathematical theory of Conceptual Knowledge Processing gives the possibility to explore the structure of data and to understand its structure. Representing knowledge as conceptual hierarchies becomes increasingly popular as a basis for further communication of knowledge. While in the dyadic setting there are well-known methods to reduce the complexity of data without affecting its underlying structure, these methods are missing in the triadic case. Driven by practical requirements, we discuss an extension of the classical reduction methods to the triadic case and apply them to a medium-sized oncological data set.

1 Introduction

Formal Concept Analysis has constantly developed in the last 30 years, one important point in its evolution being, the extension to Triadic Formal Concept Analysis (3FCA) proposed by Lehmann and Wille in [7]. Wille introduces Conceptual Knowledge Processing as an approach to knowledge management which is based on Formal Concept Analysis as its underlying mathematical theory [12, 14]. Dealing with three-dimensional data-sets, 3FCA is used to build triadic landscapes of knowledge [13]. The present paper is part of a broader discussion on a navigation paradigm in triadic conceptual landscapes.

Triadic FCA has been successfully used in inherently triadic scenarios such as collaborative tagging [6], triadic factor analysis [4], or investigation of oncological databases [10]. Despite the fact that 3FCA is just an extension of FCA, the graphical representation for the dyadic case does not have an intuitive extension to the triadic case. An initial investigation based on locally displaying smaller parts of the space of triconcepts, using *perspectives* for navigation has been done in [9].

For dyadic contexts, reducible objects and attributes can be deleted, without affecting the underlying conceptual structure. Clarifying and reducing is thus a preprocessing stage, in order to simplify the structure of the context for further analysis. For triadic data sets, these notions have not been defined until now. This paper is devoted to reduction procedures in triadic contexts and an analysis

of the effects of reducing in a medical data set is provided in the applications section. The paper concludes with a discussion about how an efficient navigation environment for different types of conceptual structures could combine existing tools (see Applications section) with newly developed navigation paradigms for triadic concept sets, starting from the same underlying data set (which does not have to be necessarily a typical triadic set).

2 Preliminaries

This section is devoted to some basic notions of triadic formal concept analysis as they have been introduced in [7, 11]. For further information about the dyadic case or more specific results about 3FCA we refer the interested reader to the standard literature [3].

Definition 1. *A* triadic context *(also:* tricontext*) is a quadruple* (K_1, K_2, K_3, Y)*, where* K_1, K_2 *and* K_3 *are sets and* $Y \subseteq K_1 \times K_2 \times K_3$ *is a ternary relation between them. The elements of* K_1, K_2, K_3 *are called (formal) objects, attributes and conditions, respectively. An element* $(g, m, b) \in Y$ *is read* object g has attribute m under condition b*.*

The following definition shows how dyadic contexts can be obtained from a triadic one in a natural way.

Definition 2 (Derived contexts). *Every triadic context* (K_1, K_2, K_3, Y) *gives rise to the following projected dyadic contexts:*

$\mathbb{K}^{(1)} := (K_1, K_2 \times K_3, Y^{(1)})$ *with* $gY^{(1)}(m, b) :\Leftrightarrow (g, m, b) \in Y$*,*
$\mathbb{K}^{(2)} := (K_2, K_1 \times K_3, Y^{(2)})$ *with* $mY^{(2)}(g, b) :\Leftrightarrow (g, m, b) \in Y$*,*
$\mathbb{K}^{(3)} := (K_3, K_1 \times K_2, Y^{(3)})$ *with* $bY^{(3)}(g, m) :\Leftrightarrow (g, m, b) \in Y$*.*

For $\{i, j, k\} = \{1, 2, 3\}$ *and* $A_k \subseteq K_k$*, we define* $\mathbb{K}^{(ij)}_{A_k} := (K_i, K_j, Y^{(ij)}_{A_k})$*, where* $(a_i, a_j) \in Y^{(ij)}_{A_k}$ *if and only if* $(a_i, a_j, a_k) \in Y$ *for all* $a_k \in A_k$*.*

Intuitively, the contexts $\mathbb{K}^{(i)}$ represent "flattened" versions of the triadic context, obtained by putting the "slices" of (K_1, K_2, K_3, Y) side by side. Moreover, $\mathbb{K}^{(ij)}_{A_k}$ corresponds to the intersection of all those slices that correspond to elements of A_k.

The derivation operators in the triadic case are defined using the dyadic derivation operators in the projected formal dyadic contexts.

Definition 3 ((i)-derivation operators). *For* $\{i, j, k\} = \{1, 2, 3\}$ *with* $j < k$ *and for* $X \subseteq K_i$ *and* $Z \subseteq K_j \times K_k$ *the* (i)*-derivation operators are defined by:*

$X \mapsto X^{(i)} := \{(a_j, a_k) \in K_j \times K_k \mid (a_i, a_j, a_k) \in Y \text{ for all } a_i \in X\}$.
$Z \mapsto Z^{(i)} := \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_j, a_k) \in Z\}$.

Obviously, these derivation operators correspond to the derivation operators of the dyadic contexts $\mathbb{K}^{(i)}, i \in \{1, 2, 3\}$.

Definition 4 ((i,j,X_k)-derivation operators). *For $\{i,j,k\} = \{1,2,3\}$ and $X_i \subseteq K_i, X_j \subseteq K_j, X_k \subseteq K_k$, the (i,j,X_k)-derivation operators are defined by*

$$X_i \mapsto X_i^{(i,j,X_k)} := \{a_j \in K_j \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_i, a_k) \in X_i \times X_k\}$$
$$X_j \mapsto X_j^{(i,j,X_k)} := \{a_i \in K_i \mid (a_i, a_j, a_k) \in Y \text{ for all } (a_j, a_k) \in X_i \times X_k\}.$$

The (i,j,X_k)-derivation operators correspond to those of the dyadic contexts $(K_i, K_j, Y_{X_k}^{(ij)})$.

Triadic concepts are defined using the above derivation operators and are maximal cuboids of incidences.

Definition 5. *A* triadic concept *(short:* triconcept*) of $\mathbb{K} := (K_1, K_2, K_3, Y)$ is a triple (A_1, A_2, A_3) with $A_i \subseteq K_i$ for $i \in \{1,2,3\}$ and $A_i = (A_j \times A_k)^{(i)}$ for every $\{i,j,k\} = \{1,2,3\}$ with $j < k$. The sets A_1, A_2, and A_3 are called* extent, intent, *and* modus *of the triadic concept, respectively. We let $\mathfrak{T}(\mathbb{K})$ denote the set of all triadic concepts of $\mathbb{K}$.*

A *complete trilattice* is a triordered set $(L, \lesssim_1, \lesssim_2, \lesssim_3)$ in which the ik-joins exist for all $i \neq k$ in $\{1,2,3\}$ and all pairs of subsets of L. We denote the set of all *order filters* of the complete trilattice L with respect to the preorder $\lesssim_i$ by $\mathcal{F}_i(L)$. A *principal filter* is denoted by $[x) := \{y \in L \mid x \lesssim_i y\}$. A subset $\mathcal{X}$ of L is said to be $i-$*dense* with respect to L if each principal filter of $(L, \lesssim_i)$ is the intersection of some order filters from $\mathcal{X}$.

Theorem 1 (The basic theorem of triadic concept analysis). *Let $\mathbb{K} := (K_1, K_2, K_3, Y)$ be a triadic context. Then $\mathfrak{T}(\mathbb{K})$ is a complete trilattice of $\mathbb{K}$ for which the ik-joins can be described as follows*

$$\nabla_{ik}(\mathcal{X}_i, \mathcal{X}_k) := \mathfrak{b}_{ik}\left(\bigcup\{A_i \mid (A_1, A_2, A_3) \in \mathcal{X}_i\}, \bigcup\{A_k \mid (A_1, A_2, A_3) \in \mathcal{X}_k\}\right).$$

In general, a complete trilattice $(L \lesssim_1, \lesssim_2, \lesssim_3)$ is isomorphic to $\mathfrak{T}(\mathbb{K})$ if and only if there exist mappings $\tilde{\kappa_i}: K_i \to \mathcal{F}_i(L) (i = 1,2,3)$ such that $\tilde{\kappa_i}(K_i)$ is i-dense with respect to L and $A_1 \times A_2 \times A_3 \subseteq Y \Leftrightarrow \cap_{i=1}^3 \cap_{a_i \in A_i} \tilde{\kappa_i}(a_i) \neq \emptyset$ for all $A_1 \subseteq K_1$, $A_2 \subseteq A_2$, $A_3 \subseteq K_3$. In particular, $L \cong \mathfrak{T}(L, L, L, Y_L)$ with $Y_L := \{(x_1, x_2, x_3) \in L^3 \mid (x_1, x_2, x_3)$ is joined$\}$.

3 Reduced tricontexts

In the dyadic case, a context is called *clarified* if there are no identical rows and columns, more precisely,

Definition 6. *A dyadic context (G, M, I) is clarified if for any objects $g, h \in G$, from $g' = h'$ follows $g = h$, and for all attributes $m, n \in M$, $m' = n'$ implies $m = n$.*

In the triadic case, we can make use of the same idea applied on the "flattened" projection of the tricontext. Since a triconcept (A_1, A_2, A_3) is a maximal triple of triadic incidences, removing identical "rows" in the tricontext does not alter the structure of triconcepts.

Definition 7. *A triadic context* (K_1, K_2, K_3, Y) *is* clarified *if for every* $i \in \{1,2,3\}$ *and every* $u, v \in K_i$*, from* $u^{(i)} = v^{(i)}$ *follows* $u = v$*.*

Context reduction is one of the most important operations performed in the dyadic case, with no effect on the conceptual structure. This consists in the removal of reducible objects and attributes. Reducible objects and attributes are precisely those objects and attributes which can be written as combinations of other objects and attributes, respectively. Formally,

Definition 8. *A clarified context* (G, M, I) *is called* row reduced *if every object concept is* $\vee$*-irreducible and* column reduced *if every attribute concept is* $\wedge$*-irreducible.*

Remark 1. Due to the symmetry of the context, if we switch the role of the objects with that of the attributes and look at the context (M, G, I^{-1}), then the context is row reduced if every object concept (attribute concept in the former context) is $\vee$-irreducible. So we can consider only $\vee$-irreducible concepts by "switching the perspective".

Similar to the dyadic case, objects, attributes, and conditions which can be written as combinations of others have no influence on the structure of the trilattice of $\mathbb{K}$, hence they can be reduced.

Definition 9. *A clarified tricontext* (K_1, K_2, K_3, Y) *is called* object reduced *if every object concept from the context* $(K_1, K_2 \times K_3, Y^{(1)})$ *is* $\vee$*-irreducible,* attribute reduced *if every object concept from the context* $(K_2, K_3 \times K_1, Y^{(2)})$ *is* $\vee$*-irreducible, and* condition reduced *if every object concept from the context* $(K_3, K_1 \times K_2, Y^{(3)})$ *is* $\vee$*-irreducible.*

Proposition 1. *Let* $g \in K_1$ *be an object and* $X \subseteq K_1$ *with* $g \notin X$ *but* $g^{(1)} = X^{(1)}$ *in* $\mathbb{K}^{(1)} = (K_1, K_2 \times K_3, Y^{(1)})$*, i.e.* g *is* $\vee$*-reducible in* $\mathbb{K}^{(1)}$*. Then*

$$\mathfrak{T}(K_1, K_2, K_3, Y) \cong \mathfrak{T}(K_1 \setminus \{g\}, K_2, K_3, Y \cap ((K_1 \setminus \{g\}) \times K_2 \times K_3)).$$

Proof. By Theorem 1, it suffices to define a map $\tilde{\kappa}_1 \colon K_1 \to \mathcal{F}_1(\mathfrak{T}(K_1 \setminus \{g\}, K_2, K_3, Y \cap (K_1 \setminus \{g\} \times K_2 \times K_3))$ such that $\tilde{\kappa}_1(K_1)$ is 1-dense in $\mathcal{F}_1(\mathfrak{T}(K_1 \setminus \{g\}, K_2, K_3, Y \cap (K_1 \setminus \{g\} \times K_2 \times K_3))$. This can be done by $\tilde{\kappa}_1(h) := \kappa(h)$ if $h \neq g$ and $\tilde{\kappa}_1(g) := \cap_{x \in X} \kappa_1(x)$ elsewhere.

Let $(A_1, A_2, A_3) \in \mathfrak{T}(\mathbb{K})$ with $g \in A_1$. Since $A_1 = (A_2 \times A_3)^{(1)}$, we have $g \in (A_2 \times A_3)^{(1)}$, wherefrom follows that $(A_2 \times A_3)^{(3)(3)} \subseteq g^{(1)} = X^{(1)}$. Then $X^{(1)(1)} \subseteq (A_2 \times A_3)^{(1)} = A_1$, hence $X \subseteq A_1$. We have that $\kappa_1(g) \subseteq \cap_{x \in X} \kappa_1(x)$. By a similar argument, we can prove the converse inclusion, hence the equality.

This proves that $\tilde{\kappa}_1(K_1)$ is 1-dense, i.e., the two trilattices are isomorphic. □

Example 1. The following example shows how reduction works:

b_1	m_1	m_2	m_3
g_1	×		
g_2			×
g_3			

b_2	m_1	m_2	m_3
g_1	×		×
g_2	×		
g_3	×		

b_3	m_1	m_2	m_3
g_1	×		
g_2	×	×	
g_3	×		

The non-trivial triconcepts of this context are: $(\{g_1\}, \{m_1\}, \{b_1, b_2, b_3\})$, $(\{g_2\}, \{m_3\}, \{b_1\})$, $(\{g_1, g_2, g_3\}, \{m_1\}, \{b_2, b_3\})$, $(\{g_1\}, \{m_1, m_3\}, \{b_2\})$, $(\{g_2\}, \{m_1, m_2\}, \{b_3\})$. We can observe that by reducing g_3, the number of triconcepts remains unchanged and the trilattice will be the same.

We obtain the following characterization for reducible elements.

Proposition 2. *Let $\mathbb{K} = (K_1, K_2, K_3, Y)$ be a tricontext and $a_i \in K_i$, $i = 1, 2, 3$. Then the element a_i is reducible if and only if there exist a subset $X \subseteq K_i$ with $Y_X^{(jk)} = Y_{a_i}^{(jk)}$, where $Y_X^{(jk)} := \{(b_j, b_k) \in K_j \times K_k \mid \forall b_i \in X.\ (b_i, b_j, b_k) \in Y\}$, for $\{i, j, k\} = \{1, 2, 3\}$.*

Proof. The element $a_i \in K_i$ is reducible if and only if there exists a subset $X \subseteq K_i$, such that they have the same derivative, i.e., $a_i^{(i)} = X^{(i)}$ in $\mathbb{K}^{(i)}$. Now $(b_j, b_k) \in Y_{a_i}^{(jk)}$ if and only if $(a, b_j, b_k) \in Y$ which is equivalent to $(b_j, b_k) \in a_i^{(i)} = X^{(i)}$. □

Remark 2. Remember that finite tricontexts can be represented as slices consisting of dyadic contexts. Moreover, this representation has a sixfold symmetry. In order to represent the triadic context in a plane, we just put these slices one next to the other (see previous example). This proposition states that a_i is reducible if and only if the slice of a_i is the intersection of some slices corresponding to the elements of a certain subset $X \subseteq K_i$. This has a striking similarity to the dyadic case, where, for example, an object is reducible, if its row is the intersection of the rows from a certain subset X of objects. This also gives us an algorithmic approach to the problem of finding all reducible elements in a tricontext.

Similar to the dyadic case, where double arrow have been introduced in order to identify those rows and columns which are not reducible (remember that a row or a column is not reducible, if it contains a double arrow), we can define a similar notion for tricontexts, where the role of the double arrow will be played by the symbol $\divideontimes$.

Definition 10. *Let $\mathbb{K} := (K_1, K_2, K_3, Y)$ be a tricontext. For $g \in K_1, m \in K_2, b \in K_3$ we define the following relations, where $\swarrow$ is the arrow relation from dyadic FCA:*

- $(g, m, b) \in \triangleleft \Leftrightarrow g \swarrow (m, b)$
- $(g, m, b) \in \triangle \Leftrightarrow m \swarrow (g, b)$
- $(g, m, b) \in \triangleright \Leftrightarrow b \swarrow (g, m)$
- $(g, m, b) \in \divideontimes \Leftrightarrow (g, m, b) \in \triangleleft$ *and* $(g, m, b) \in \triangle$, *and* $(g, m, b) \in \triangleright$

Remark 3. An element $a_i \in K_i$ will be reducible if and only if its corresponding slice, i.e., $(K_j, K_k, Y_{a_i}^{(jk)})$ does not contain the triadic arrow $\divideontimes$.

In the dyadic case, object and attribute concepts are playing an important role, see for instance the Basic Theorem on Concept Lattices. We might ask if there is a similar notion in the triadic case. Due to the structure of triconcepts, it proves that an object concept, for instance, should be defined as a set of triconcepts.

Definition 11. *Let $\mathbb{K} := (K_1, K_2, K_3, Y)$ be a tricontext, $g \in K_1$, $m \in K_2$, and $b \in K_3$ be objects, attributes, and conditions, respectively. The* object concept *of g is defined as $\gamma^{\Delta}(g) := \{(A_1, A_2, A_3) \in \mathfrak{T}(\mathbb{K}) \mid A_1 = g^{(1)(1)}\}$, where $(\cdot)^{(i)}$ is the derivation operator g in $\mathbb{K}^{(i)}$, $i \in \{1, 2, 3\}$. Similar, the* attribute concept *of m is defined as $\mu^{\Delta}(m) := \{(A_1, A_2, A_3) \in \mathfrak{T}(\mathbb{K}) \mid A_2 = m^{(2)(2)}\}$, while the* condition concept *of b is defined as $\beta^{\Delta}(b) := \{(A_1, A_2, A_3) \in \mathfrak{T}(\mathbb{K}) \mid A_2 = b^{(3)(3)}\}$.*

Lemma 1. *Let (K_1, K_2, K_3, Y) be a tricontext, $a_i \in K_i$, $i \in \{1, 2, 3\}$. Let $\Gamma_1(a_1) := [\gamma_1^{\Delta}(a_1))$ be the filter generated by the triadic object concept $\gamma_1^{\Delta}(a_1)$ in $(\mathfrak{T}(\mathbb{K}), \lesssim_1)$ (and similar $\Gamma_2(a_2)$, and $\Gamma_3(a_3)$ for attribute and conditions triconcepts, respectively). Then $\Gamma_i(K_i) := \{\Gamma_i(a_i) \mid a_i \in K_i\}$ is i-dense in $(\mathfrak{T}(\mathbb{K}), \lesssim_1, \lesssim_2, \lesssim_3)$.*

Proof. Following the construction used in the proof of Theorem 1, the principal filter of the triadic concept (A_1, A_2, A_3) in $(\mathfrak{T}(\mathbb{K}), \lesssim_i)$ is $\bigcap_{a_i \in A_i} \{(B_1, B_2, B_3) \in \mathfrak{T}(\mathbb{K}) \mid a_i \in B_i\} \in \mathcal{F}_i(\mathfrak{T}(\mathbb{K}))$. Combining this with the fact that for $(B_1, B_2, B_3) \in \mathfrak{T}(\mathbb{K})$, $a_i \in B_i$ iff $a_i^{(i)(i)} \subseteq B_i$, we obtain an i-dense set of order filters $\Gamma_i(K_i)$ and $\Gamma_i(a_i) = \{(B_1, B_2, B_3) \in \mathfrak{T}(\mathbb{K}) \mid a_i \in B_i\}$ for $a_i \in K_i$ and $i = 1, 2, 3$. □

4 Applications

In this section we discuss some applications of the previous results on a cancer registry database comprising information about several thousand patients. Even if the original data set does not have an inherently triadic format, one can select triadic subsets herefrom which are then suitable for further analysis. This proves that even many-valued dyadic contexts can be interpreted and studied from a triadic point of view. For more about this interpretation mechanism we refer to [10]). In order to prepare the data for a triadic interpretation, the knowledge management suite ToscanaJ ([1]) and Toscana2Trias, a triadic extension developed at Babes-Bolyai University Cluj-Napoca have been used. Toscana2Trias uses the TRIAS algorithm developed by R. Jaeschke et al. [5]. It connects to a database and displays the table names (or attribute names). The user may define, according to his own view, which are the objects, the attributes and the conditions. The ternary incidence relation is then read from the database. Moreover, if a conceptual schema has been built upon the data set, i.e., the data has been preprocessed for ToscanaJ, then the user has even more control over the selection of objects, attributes and conditions. From the conceptual schema, a part of the scaled attributes can be considered as conditions, the rest being considered as attributes in the tricontext. Triadic concepts are then computed, using the Trias algorithm and displayed in a variety of formats. If the data set is larger, the visualization becomes easily obscure because of the number of triconcepts. In this case, one can make use of the navigation paradigm discussed in [9].

The cancer registry database, in its original form, contains 25 attributes for each patient, including an identification number, for example *Tumor sequence, Topography, Morphology, Behavior, Basis of diagnosis, Differentiation*

degree, Surgery, Radiotherapy, Hormonal Therapy, Curative Surgery, Curative Chemotherapy etc. These attributes are all interpreted as conceptual scales and represented as conceptual landscapes for an enhanced knowledge retrieval.

The triadic approach makes possible to investigate these data from a totally different point of view. While a typical usage of ToscanaJ implies the combination of several scales into a so-called *b*rowsing scenario, 3FCA gives a certain depth to the scale-based navigation of the conceptual landscapes.

For the first example, we have selected a number of 4686 objects, 11 attributes (all 8 degrees of certainty in the oncological decision process, in-situs carcinoma and tumor sequence 1, i.e., just one tumor) and three conditions ($Gender = Male$, $age < 59$, and $survival > 30\ months$). This selection generated a relation with 44545 tuples (crosses in the tricontext) and 63 triconcepts and a clarified tricontext with 61 objects. Herefrom, 38 objects could be reduced as well as 7 attributes (all of them being certainty-related, due to the specific selection we have made), resulting in a relation with 77 tuples.

For the next example, the selection was restricted to types of tumors (as attributes) versus stage (as conditions). A clarified tricontext resulted, with 13 objects, 5 attributes and 8 conditions, and 23 triconcepts. Three more objects, one attribute and one condition could be further reduced.

5 Conclusions and Future Work

In this paper we have defined the notion of reduction for triadic FCA and the notion of triadic object, attribute, and condition concept, showing that these triconcepts are playing for the basic theorem of 3FCA the same role to that played by object and attribute concepts in the dyadic case.

In the applications section, we have shown how reducing a tricontext eliminates redundant information, hence increasing the efficiency in determining its underlying conceptual structure. Moreover, due to the selection procedure specific to the Toscana2Trias extension, reducible objects (or attributes, conditions) may give important clues about the structure of the data subset.

This contribution is a natural development of the navigation paradigm discussed in [9], which will include reduction as a preprocessing stage. The ToscanaJ knowledge management suite and its triadic extension Toscana2Trias makes possible to generate triadic data sets in a natural way, even if the underlying data does not have a natural triadic structure (as, for instance, folksonomies have). A navigation tool for triadic conceptual landscapes is imperatively necessary, and the local navigation approach described in [9] makes use of a similar approach to that of combining scales in ToscanaJ, hence restricting only to a local view. A selection of the starting points for navigation could be performed by user defined constraints. More specifically, the user defines two lists: one containing required and one forbidden objects, attributes and conditions. This selection will focus on a subset of triconcepts, wherefrom navigation can start. For a detailed discussion of user defined constraints for FCA, including complexity results, we refer to [8].

References

1. Becker, P., Correia, J.H.: The toscanaj suite for implementing conceptual information systems. In: Ganter et al. [2], pp. 324–348
2. Ganter, B., Stumme, G., Wille, R. (eds.): Formal Concept Analysis, Foundations and Applications, Lecture Notes in Computer Science, vol. 3626. Springer (2005)
3. Ganter, B., Wille, R.: Formal concept analysis - mathematical foundations. Springer (1999)
4. Glodeanu, C.: Triadic factor analysis. In: Kryszkiewicz, M., Obiedkov, S.A. (eds.) Proceedings of the 7th International Conference on Concept Lattices and Their Applications, Sevilla, Spain, October 19-21, 2010. CEUR Workshop Proceedings, vol. 672, pp. 127–138. CEUR-WS.org (2010), http://ceur-ws.org/Vol-672/paper12.pdf
5. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: TRIAS - an algorithm for mining iceberg tri-lattices. In: Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China. pp. 907–911. IEEE Computer Society (2006), http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4053012
6. Jäschke, R., Hotho, A., Schmitz, C., Ganter, B., Stumme, G.: Discovering shared conceptualizations in folksonomies. Journal of Web Semantics 6(1), 38–53 (2008)
7. Lehmann, F., Wille, R.: A triadic approach to formal concept analysis. In: Ellis, G., Levinson, R., Rich, W., Sowa, J.F. (eds.) Proceedings of the Third International Conference on Conceptual Structures, ICCS '95. LNCS, vol. 954, pp. 32–43. Springer (1995)
8. Rudolph, S., Săcărea, C., Troancă, D.: Membership constraints in formal concept analysis. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI) (2015), to appear
9. Rudolph, S., Săcărea, C., Troancă, D.: Towards a navigation paradigm for triadic concepts. In: Baixeries, J., Sacarea, C., Ojeda-Aciego, M. (eds.) Proceedings of the 13th International Conference on Formal Concept Analysis (ICFCA 2015). LNCS, vol. 9113, pp. 232–248. Springer (2015)
10. Săcărea, C.: Investigating oncological databases using conceptual landscapes. In: Hernandez, N., Jäschke, R., Croitoru, M. (eds.) Graph-Based Representation and Reasoning - 21st International Conference on Conceptual Structures, ICCS 2014, Iaşi, Romania, July 27-30, 2014, Proceedings. Lecture Notes in Computer Science, vol. 8577, pp. 299–304. Springer (2014)
11. Wille, R.: The basic theorem of triadic concept analysis. Order 12(2), 149–158 (1995)
12. Wille, R.: Begriffliche Wissensverarbeitung: Theorie und Praxis. Informatik Spektrum (23), 357–369 (2000)
13. Wille, R.: Formal concept analysis as mathematical theory of concepts and concept hierarchies. In: Ganter et al. [2], pp. 1–33
14. Wille, R.: Methods of conceptual knowledge processing. In: Missaoui, R., Schmid, J. (eds.) Formal Concept Analysis, 4th International Conference, ICFCA 2006, Dresden, Germany, February 13-17, 2006, Proceedings. Lecture Notes in Computer Science, vol. 3874, pp. 1–29. Springer (2006)

Lazy associative graph classification

Yury Kashnitsky, and Sergei O. Kuznetsov

National Research University Higher School of Economics
Moscow, Russia
{ykashnitsky, skuznetsov}@hse.ru

Abstract. In this paper, we introduce a modification of the lazy associative classification which addresses the graph classification problem. To deal with intersections of large graphs, graph intersections are approximated with all common subgraphs up to a fixed size similarly to what is done with graphlet kernels. We illustrate the algorithm with a toy example and describe our experiments with a predictive toxicology dataset.

Keywords: graph classification, graphlets, formal concept analysis, pattern structures, lazy associative classification

1 Introduction

Classification methods for data given by graphs usually reduce initial graphs to numeric representation and then use standard classification approaches, like SVM [1] and Nearest neighbors with graph kernels [2], graph boosting [3], etc. By doing so, one usually constructs numeric attributes corresponding to subgraphs of initial graphs or computes graph kernels, which usually are also based on the number of common subgraphs of special type. In this paper, we suggest an approach based on weak classifiers in the form of association rules [4] applied in a "lazy" way: not all of the association rules are computed to avoid exponential explosion, but only those that are relevant to objects to be classified. Lazy classification is well studied experimentally [5], here we extend the approach to graphs and propose a uniform theoretical framework (based on pattern structures [6]) which can be applied to arbitrary kinds of descriptions. We show in a series of experiments with data from the Predictive Toxicology Challenge (PTC [7]) that our approach outperforms learning models based on SVM with graphlet kernel [8] and kNN with graphlet-based distance.

The rest of the paper is organized as follows. In Section 2, we give main definitions on labeled graphs, pattern structures, and lazy associative classification. In Section 3, we consider an example. In Section 4, we discuss the results of computational experiments on PTC dataset. In Section 5, we give the conclusion and discuss directions of further research.

2 Main definitions

In this section, we give the definitions of the main concepts used in the paper.

2.1 Labeled graphs and isomorphism

First, we recall some standard definitions related to labeled graphs, see e.g. [9,10,11].

Undirected graph is a pair $G = (V, E)$. Set V is referred to as a set of *nodes* of a graph. Set $E = \{\{v, u\} \mid v, u \in V\} \cup E_0$, a set of unordered elements of V, is called a set of *edges*, and $E_0 \subseteq V$ — is a set of *loops*. If $E_0 = \emptyset$, then G is called a *graph without loops*.

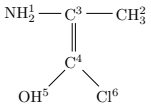
Graph $H = (V_H, E_H)$ is called a *subgraph* of graph $G = (V_G, E_G)$, if all nodes and edges of H are at the same time nodes and edges of G correspondingly, i.e. $V_H \subseteq V_G$ and $E_H \subseteq E_G$.

Graph $H = (V_H, E_H)$ is called an *induced subgraph* of graph $G = (V_G, E_G)$, if H is a subgraph of G, and edges of H are comprised of all edges of G with both nodes belonging to H.

Given sets of nodes V, node labels L_V, edges E, and edge labels L_E, a *labeled* graph is defined by a quadruple $G = ((V, lv), (E, le))$ such that

- $lv \subseteq V \times L_V$ is the relation that associates nodes with labels, i.e., lv is a set of pairs (v_i, l_i) such that node v_i has label l_i,
- $le \subseteq V \times V \times L_E$ is the relation that associates edges with labels, i.e., le is a set of triples (v_i, v_j, l_{ij}) such that edge (v_i, v_j) has label l_{ij}.

Example 1. A molecule structure can be represented by a labeled graph.

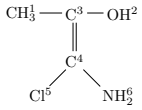
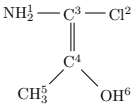


Here $V = \{1, 2, 3, 4, 5, 6\}$, $E = \{(1, 3), (2, 3), (3, 4), (4, 5), (4, 6)\}$,
$lv = \{(1, NH_2), (2, CH_3), (3, C), (4, C), (5, OH), (6, Cl)\}$,
$le = \{(1, 3, 1), (2, 3, 1), (3, 4, 2), (4, 5, 1), (4, 6, 1)\}$, and edge type 1 corresponds to a single bond (ex. HN_2—C) while edge type 2 – to a double bond (ex. $C = C$).

A labeled graph $G_1 = ((V_1, lv_1), (E_1, le_1))$ *dominates* a labeled graph $G_2 = ((V_2, lv_2), (E_2, le_2))$ with given order $\leq$ (e.g. natural, lexicographic) on vertex and edge labels, or $G_2 \leq G_1$ (or G_2 is a *subgraph* of G_1), if there exists an injection $\varphi\colon V_2 \to V_1$ such that it:

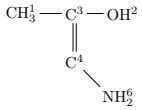
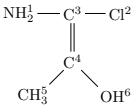
- respects edges: $(v, w) \in E_2 \Rightarrow (\varphi(v), \varphi(w)) \in E_1$,
- fits under labels: $lv_2(v) \leq lv_1(\varphi(v)), (v, w) \in E_2 \Rightarrow le_2(v, w) \leq le_1(\varphi(v), \varphi(w))$.

Two labeled graphs G_1 and G_2 are called *isomorphic* ($G_1 \simeq G_2$) if $G_1 \leq G_2$ and $G_2 \leq G_1$.

Example 2. G_1 :  G_2 : 

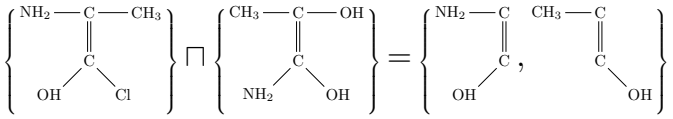
$G_1 \simeq G_2$ as $\exists\varphi : V_2 = \{1,2,3,4,5,6\} \to V_1 = \{1,2,3,4,5,6\} = (6,5,4,3,1,2)$, satisfying the definitions of graph dominance and isomorphism.

An injective function $f : V \to V'$ is called a *subgraph isomorphism* from G to G', if there exists a subgraph of G': $S \leq G'$, such that f is a graph isomorphism from G to S, or $G \simeq S$.

Example 3. G_1 :  G_2 : 

G_1 is subgraph-isomorphic to G_2.

Given labeled graphs G_1 and G_2, a set $G_1 \sqcap G_2 =$
$\{G \mid G \leq G_1, G_2, \quad \forall G_* \leq G_1, G_2 \quad G_* \not\geq G\}$ is called a set of *maximal common subgraphs* of graphs G_1 and G_2. We also refer to $G_1 \sqcap G_2$ as to *intersection* of graphs G_1 and G_2, and to $\sqcap$ – as to *similarity operator* defined on graphs.

Example 4. 

For sets of graphs $\mathcal{G} = \{G_1, \ldots, G_k\}$ and $\mathcal{H} = \{H_1, \ldots, H_n\}$ the similarity operator is defined in the following way:

$\mathcal{G} \sqcap \mathcal{H} = \mathrm{MAX}_{\leq}\{G_i \sqcap H_i \mid G_i \in \mathcal{G}, H_j \in \mathcal{H}\}$

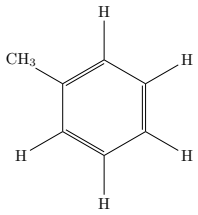
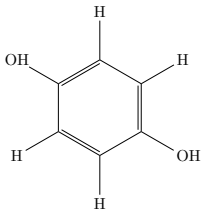
Given sets of labeled graphs $\mathcal{G}_1$ and $\mathcal{G}_2$, we say that a set of graphs $\mathcal{G}_1$ is *subsumed* by a set of graphs $\mathcal{G}_2$, or $\mathcal{G}_1 \sqsubseteq \mathcal{G}_2$, if $\mathcal{G}_1 \sqcap \mathcal{G}_2 = \mathcal{G}_1$.

2.2 Graphlets

Definition 1. *A labeled graph g is called a k-**graphlet** of a labeled graph G if g is a connected induced subgraph of graph G with k nodes [12].*

Definition 2. *A set of labeled graphs $\mathcal{G}^k$ is called a k-**graphlet representation** of a labeled graph G if any $g \in \mathcal{G}$ is a unique (up to subgraph isomorphism) k-graphlet of graph G, i.e*
$\forall g \in \mathcal{G}^k$ graph g is a k-graphlet of G, $\forall g_1, g_2 \in \mathcal{G}$ one does not have $g_1 \leq g_2$.

Definition 3. *k-**graphlet distribution** of a labeled graph G is the set $\{(g_i, n_i)\}$, where g_i is a k-graphlet of G and n_i is the number of k-graphlets in G isomorphic to g_i.*

Example 5. G_1 :  G_2 : 

$\mathcal{G}_1 = \{C-C=C,\ C-C-H,\ C=C-H, C-C-C\}$,
$\mathcal{G}_2 = \{C-C=C,\ C-C-H, C=C-H,\ C-C-O,\ C=C-O,\ C-O-H\}$ – are 3-graphlet representations of graphs G_1 and G_2 correspondingly (with benzene rings comprised of carbon molecules C). 3-graphlet distributions of graphs G_1 and G_2 are given in Table 1.

Table 1. 3-graphlet distributions of graphs G_1 and G_2 (benzene rings are comprised of carbon molecules C).

	CC=C	CCH	C=CH	CCO	C=CO	COH	CCC
G_1	7	8	5	0	0	0	1
G_2	6	4	4	2	2	2	0

Graphlets were introduced in biomedicine and are used to compare real cellular networks with their models. It is easy to demonstrate that two networks are different by simply showing a short list of properties in which they differ. It is much harder to show that two networks are similar, as it requires demonstrating their similarity in all of their exponentially many properties [12].

Graphlet distribution serves as a measure of network local structure agreement and was shown to express more structural information than other metrics such as centrality, local clustering coefficient, degree distribution etc. In [12], they considered all 30 combinations[1] of graphlets with 2, 3, 4 and 5 nodes.

2.3 Pattern structures

Pattern structures are natural extension of ideas proposed in Formal Concept Analysis [13], [6].

Definition 4. *Let G be a set (of objects), let $(D,\ \sqcap)$ be a meet-semi-lattice (of all possible object descriptions) and let $\delta : G \to D$ be a mapping between objects and descriptions. Set $\delta(G) := \{\delta(g)|g \in G\}$ generates a complete subsemilattice $(D_\delta,\ \sqcap)$ of $(D,\ \sqcap)$, if every subset X of $\delta(G)$ has infimum $\sqcap X$ in $(D,\ \sqcap)$.* ***Pattern structure*** *is a triple $(G,\ \underline{D},\ \delta)$, where $\underline{D} = (D,\ \sqcap)$, provided that the set $\delta(G) := \{\delta(g) \mid g \in G\}$ generates a complete subsemilattice $(D_\delta,\ \sqcap)$ [6,11].*

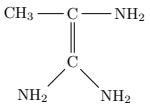
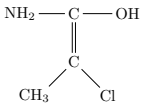
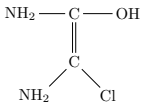
[1] `https://parasol.tamu.edu/dreu2013/OLeary`

Definition 5. ***Patterns*** *are elements of* D*. Patterns are naturally ordered by subsumption relation* $\sqsubseteq$*: given* $c,\ d \in D$ *one has* $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$*. Operation* $\sqcap$ *is also called a* ***similarity*** *operation. A pattern structure* $(G,\ \underline{D},\ \delta)$ *gives rise to the following* ***derivation operators*** $(\cdot)^\diamond$*:*

$$A^\diamond = \prod_{g \in A} \delta(g) \qquad \text{for } A \in G,$$
$$d^\diamond = \{g \in G \mid d \sqsubseteq \delta(g)\} \qquad \text{for } d \in (D,\ \sqcap).$$

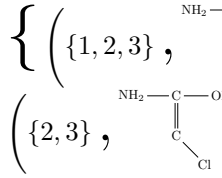
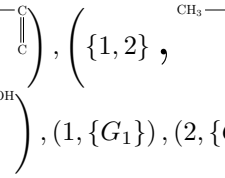
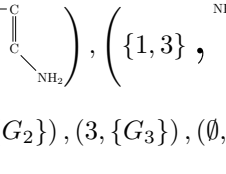
Pairs $(A,\ d)$ *satisfying* $A \subseteq G,\ d \in \underline{D},\ A^\diamond = d$*, and* $A = d^\diamond$ *are called* ***pattern concepts*** *of* $(G,\ \underline{D},\ \delta)$*.*

Example 6. Let $\{1, 2, 3\}$ be a set of objects, $\{G_1, G_2, G_3\}$ – be a set of their descriptions (i.e., graph representations):

G_1 :  G_2 :  G_3 : 

D is the set of all sets of labeled graphs, $\sqcap$ is a graph intersection operator, $\underline{D} = (D,\ \sqcap)$. A set of objects (graphs) $\{1, 2, 3\}$, their "descriptions" (i.e. graphs themselves) $D = \{G_1, G_2, G_3\}$ $(\delta(i) = G_i, i = 1, \ldots, 3)$, and similarity operator $\sqcap$ comprises a pattern structure $(\{1, 2, 3\}, \underline{D}, \delta)$.
$\{1, 2, 3\}^\diamond = \{NH_2 - C = C\}$, because $\{NH_2 - C = C\}$ is the only graph, subgraph-isomorphic to all three graphs $1, 2$, and 3. Likewise,
$\{NH_2 - C = C\}^\diamond = \{1, 2, 3\}$, because graphs $1, 2$, and 3 subsume graph $\{NH_2 - C = C\}$.
$\{1, 2\}^\diamond = \{CH_3 - C = C - NH_2\}$, because $\{CH_3 - C = C - NH_2\}$ is a graph, subgraph-isomorphic to 1, and 2, but not to graph 3. Likewise,
$\{CH_3 - C = C - NH_2\}^\diamond = \{1, 2\}$, because only graphs 1, and 2 subsume graph $\{CH_3 - C = C - NH_2\}$, but graph 3 does not.

Here is the set of all pattern concepts for this pattern structure:

$\Big\{$  ,  ,  , $, (1, \{G_1\}), (2, \{G_2\}), (3, \{G_3\}), (\emptyset, \{G_1, G_2, G_3\}) \Big\}$.

For some pattern structures (e.g., for the pattern structures on sets of graphs with labeled nodes) even computing subsumption of patterns may be NP-hard. Hence, for practical situations one needs approximation tools, which would replace the patterns with simpler ones, even if that results in some loss of information. To this end, we use a contractive monotone and idempotent mapping $\psi : D \to D$ that replaces each pattern $d \in D$ by $\psi(d)$ such that the pattern

structure $(G,\ \underline{D},\ \delta)$ is replaced by $(G,\ \underline{D},\ \psi \circ \delta)$. Under some natural algebraic requirements that hold for all natural projections in particular pattern structures we studied in applications, see [11], the meet operation $\sqcap$ is preserved: $\psi(X \sqcap Y) = \psi(X) \sqcap \psi(Y)$. This property of a projection allows one to relate premises in the original representation with those approximated by a projection. In this paper, we utilize projections to introduce graphlet-based classification rules.

2.4 Lazy associative classification

Consider a binary classification problem with a set of positive examples G_+, negative examples G_-, test examples G_{test}, and a pattern structure $(G_+ \cup\ G_-, \underline{D}, \delta)$ defined on the training set.

Definition 6. *A pattern $h \in D$ is a **positive premise** iff [11]*

$$h^\diamond \cap G_- = \emptyset \text{ and } h^\diamond \cap G_+ \neq \emptyset$$

A positive premise is a subset of the *least general generalization* of descriptions of positive examples, which is not contained in (does not cover) any negative example. A *negative premise* is defined similarly. Various classification schemes using premises are possible, as an example consider the following simplest scheme from [6]: if the description $\delta(g)$ of an undetermined example g contains a positive premise h, i.e., $h \sqsubseteq \delta(g)$, then g is *classified positively*. Negative classifications are defined similarly. If $\delta(g)$ contains premises of both signs, or if $\delta(g)$ contains no premise at all, then the classification is contradictory or undetermined, respectively, and some probabilistic techniques allowing for a certain tolerance should be applied.

Definition 7. Class association rule *(CAR) [5] for a binary classification problem is an association rule in a form $h \to \{+, -\}$, where h is a positive or negative premise, respectively.*

The definition means that for a binary graph classification problem, for instance, we can mine classification association rules in a form $\{g_i\} \to \{+, -\}$, i.e. if a test graph subsumes a subgraph g_i, that is common only to positive (negative) training examples, it is therefore classified as positive (negative). We elaborate this idea in the next subsection. As there might be lots of such CARs, we might come up with a single classification rule taking into account these CARs. For instance, we can count all positive and negative CARs for each test object and classify it with a majority voting procedure. Of course, the idea is easily generalized to multi-label classification problem. The described classification schemes are explored in [5].

Another advantage of the lazy classification framework is its obvious parallelization. Suppose there are K processors. If we consider classification of an unlabeled object we can divide the training set into K separate subsets. Then, for each subset we perform intersections between the labeled objects with the unlabeled one to be classified. After all unfalsified intersections are found we can go on to the classification phase which involves voting based on those intersections.

2.5 Graphlet-based lazy associative classification

In this subsection, we combine the ideas of pattern structures and their projections, graphlets, and lazy associative classification, and introduce our algorithm. First, we recall the definition of k-projection producing all graphs with less than or equal to k nodes.

Definition 8. *Given a graph pattern structure $(G, \underline{D}, \delta)$, we call $\psi_k(G) = \{H_i = ((V_i, lv_i), (E_i, le_i)) \mid H_i \leq G, H_i$ is connected, $|V_i| \leq k\}$ a k-**projection**, defined for graph descriptions G.*

Obviously, this operator is a projection, i.e. contractive, monotone, and idempotent function.

Definition 9. *Given a graph pattern structure $(G, \underline{D}, \delta)$, k-**graphlet derivation operator** $\delta_k = \bigcup_{1 \leq l \leq k} \psi_l \circ \delta$ takes an object g described by graph G and produces all l-graphlets of G for $l = 1, \ldots k$.*

Example 7. For object 1 with "graph description" G_1 from example 5 $\delta_3(1)$ is the set of all 1-,2-, and 3-graphlets of graph 1:
$\delta_3(1) = \{C,\ H,\ C-C,\ C=C,\ C-H,\ C-C=C,\ C-C-H,\ C=C-H,\ C-C-C\}$. To clarify, here $\delta(1) = \{G_1\}, \delta_3(1) = \psi_3(\delta(1)) = \psi_3(G_1) = \{H_i = ((V_i, lv_i), (E_i, le_i)) \mid H_i \leq G_1, |V_i| \leq 3\}$.

Definition 10. *Given k-graphlet representations $\mathcal{G}_1^k$ and $\mathcal{G}_2^k$ of labeled graphs G_1 and G_2, the intersection $\mathcal{G}_1^k \sqcap_k \mathcal{G}_2^k$ is called k-**graphlet intersection** of G_1 and G_2. The $\sqcap_k$ operator is further called k-**graphlet similarity operator**.*

Example 8. For graphs 1 and 2 with "graph descriptions" G_1 and G_2 from example 5 $G_1 \sqcap_3 G_2 = \{C,\ H, C-C, C=C, C-H, C-C=C, C-C-H, C=C-H\}$ is the set of all common 1-, 2-, and 3-graphlets of graphs 1 and 2.

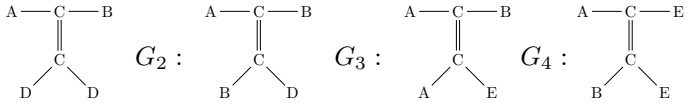
Here are the main steps of our algorithm:

1. All k-graphlet intersections of test examples and positive training examples are computed: $h_+ = G_{tr} \sqcap_k G_+$;
2. Each intersection h_+ is tested on subsumption by negative training examples. If some of them subsumes h_+, then this intersection is *falsified*. Otherwise, h_+ gives a vote for positive classification of the test example G_{tr};
3. The same procedure is done for each intersection of G_{tr} with negative examples;
4. Test example G_{tr} is classified according to the weighted majority rule where each unfalsified intersection is given a weight equal to its cardinality (the cardinality of the corresponding set of graphs).

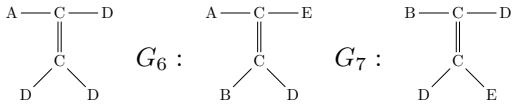
3 A toy example

We illustrate the principle of our method with a toy example. Let us consider the following training and test sets comprised of molecular descriptions of toxic (G_1 – G_4) and non-toxic (G_5 – G_7) chemical compounds. The task is to build a discriminative classifier able to determine whether the objects from the test set ($G_8 - G_{11}$) are toxic or not. The main steps of the algorithm, described in the previous section, are briefly illustrated with Tables 2 and 3. First, we build 3-graphlet intersections of test and training examples (we use only graphlets with 3 nodes for the purpose of illustration). Then, a "+" or "—" sign with cardinality of intersection is put in Table 3 if this intersection is not subsumed by any example of the opposite class. Otherwise, the counter-example subsuming this intersection is given.

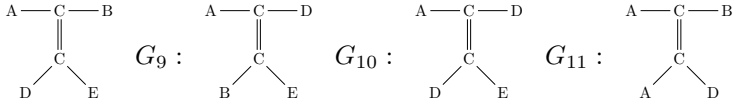
Positive examples:



Negative examples:



Test examples:



3-graphlet intersections of training and test examples are given in Table 2. For instance, graphs G_1 and G_8 have 4 common 3-graphlets: A–C–B, A–C=C, B–C=C, and C=C–D. In this simple case, we do not differentiate between a single and a double bond (e.g., ACC here stands for A–C=C without ambiguity).

Further, Table 3 summarizes the procedure. For instance, a '+4' sign for graphs G_1 and G_8 means that all common 3-graphlets of G_1 and G_8 (i.e., A–C–B, A–C=C, B–C=C, and C=C–D) are not subgraph-isomorphic to any of the negative examples G_5 – G_7 altogether at the same time. Thus, this intersection "gives a vote" of weight 4 (the cardinality of the mentioned set of graphlets) for positive classification of G_8. On the contrary, all common 3-graphlets of G_4 and G_8 (A–C=C, B–C=C, and C=C–E) are altogether subgraph-isomorphic to negative example G_6, therefore, the intersection of G_4 and G_8 doesn't "give a vote" for positive classification of G_8.

Thus, molecules G_8 and G_{11} are classified as toxic, G_9, G_{10} are classified as non-toxic.

4 Experiments

The proposed algorithm was tested with the 2001 Predictive Toxicology Challenge dataset in comparison with SVM with graphlet kernel and k-Nearest-

Table 2. All common 3-graphlets of test ($G_8 - G_{11}$) and training examples.

	G_8	G_9	G_{10}	G_{11}
G_1	ACB, ACC, BCC, CCD	ACC, BCC, CCD	ACC, CCD	ACB, ACC, BCC, CCD
G_2	ACB, ACC, BCC, CCD	ACC, BCC, CCD	ACC, CCD	ACB, ACC, BCC, CCD
G_3	ACB, ACC, BCC, CCE	ACC, BCC, CCE	ACC, CCE	ACB, ACC, BCC
G_4	ACC, BCC, CCE	ACC, BCC, BCE, CCE	ACC, CCE	ACC, BCC
G_5	ACC, CCD	ACC, ACD, CCD	ACC, ACD, CCD	ACC, ACD, CCD
G_6	ACC, BCC, CCD, CCE	ACC, BCC, CCD, CCE	ACC, CCD, CCE	ACC, BCC, CCD
G_7	BCC, CCD, CCE, DCE	BCC, CCD, CCE	CCD, CCE, CDE	BCC, CCD

Table 3. Lazy classification table

	G_1	G_2	G_3	G_4	G_5	G_6	G_7	Score	Class
G_8	+4	+4	+4	G_6	G_1	–4	–4	4:0	+
G_9	G_6	G_6	G_6	+4	–3	–4	–3	0:6	—
G_{10}	G_5	G_5	G_6	G_6	–3	–3	–3	0:9	—
G_{11}	+4	+4	+3	G_6	–3	G_1	G_1	8:0	+

Neighbor with graphlet-based Hamming distance. SVM classifiers are considered to be good benchmarks for graph classification problem [8]. We implemented a Scikit-learn [14] version of Support Vector Classifier with graphlet kernel and graphlets having up to 5 nodes. We also adopted a k-Nearest-Neighbor for graph classification problem by defining a Hamming distance between two graphs (0 if two objects have a certain graphlet in common, 1 otherwise). For instance, for two graphs from example 5 in case of graphlets with up to 3 nodes this distance is equal to 7 (G_1 subsumes graphlet $C-C-C$ not subsumed by G_2, while G_2 subsumes graphlets $\{O,\ C-O,\ O-H,\ C-C-O,\ C=C-O,\ C-O-H\}$ not subsumed by G_1).

The training set is comprised of 417 molecular graphs of chemical compounds with indication of whether a compound is toxic or not for a particular sex and species group out of four possible groups: {mice, rats} × {male, female}. Thus, 4 separate sets were built for male rats (MR, 274 examples, 117 are toxic for male rats, 157 are non-toxic), male mice (MM, 266 examples, 94 are positive, 172 are negative), female rats (FR, 281 examples, 86 are positive, 195 are negative) and female mice (FM, 279 examples, 108 are positive, 171 are negative).

We run 5-fold cross-validation for each group (MR, MM, FR, FM) and compared average classification metrics for each fold. The results for male rats are presented in Table 4 (we got similar results for other groups).

The parameters for SVM and kNN classifiers were tuned through the process of GridSearch cross-validation[2]. The 'K nodes' parameter determines the maximum number of nodes in graphlet representation of graphs, i.e. when it is equal to 4, all graph are approximated with their 4-graphlet representation, or all unique (in the sense of isomorphism) graphlets with up to 4 nodes.

As we can observe, graphlet-based lazy associative classification is reasonable with at least 3-graphlet descriptions. In case of 2-graphlet descriptions the

[2] `http://scikit-learn.org/stable/modules/grid\_search.html`

Table 4. Experimental results for the male rats group. "GLAC" stands for "Graphlet-based lazy associative classification", "SVM" here denotes "Support Vector Machine with graphlet kernel" "kNN" here stands for a k-Nearest-Neighbor classifier with Hamming distance.

	K nodes	Accuracy	Precision	Recall	F-score	Time (sec.)
GLAC	2	0.36	0.32	0.33	0.32	5.78
	3	0.68	0.83	0.68	0.75	17.40
	4	0.59	0.57	0.62	0.59	65.72
	5	0.55	0.7	0.62	0.66	196.03
SVM	2	0.45	0.15	0.33	0.21	1.54
	3	0.52	0.35	0.35	0.35	9.03
	4	0.41	0.27	0.28	0.28	61.31
	5	0.36	0.24	0.25	0.24	295.89
kNN	2	0.45	0.15	0.33	0.21	3.35
	3	0.34	0.21	0.23	0.22	15.75
	4	0.48	0.31	0.32	0.31	73.38
	5	0.45	0.30	0.31	0.30	211.58

algorithm often refuses to classify test objects, because 2-graphlet intersections of positive and test objects are falsified by negative objects and vice versa. But 3-graphlet descriptions are optimal for this method as the model is probably overfitted in case of 4- and 5-graphlet descriptions.

5 Conclusion

In this paper, we have proposed an approach to graph classification based on the combination of graphlets, pattern structures and lazy classification. The key principle of lazy classification is that one does not have to produce the whole set of classification rules whatever they are. Instead, one generates those rules that allow one to classify the current test object. The framework favors the complex structure of objects as soon as the algorithm does not require a training phase.

We have carried out a number of experiments in molecule classification within the proposed lazy classification framework. We compared classification performance of our method and SVM with graphlet kernel and KNN with graphlet-based distance. The reason for such a choice is that SVM classifiers are considered to be good benchmarks for graph classification problem, while kNN is a famous lazy classification method.

In our experiments graphlet-based lazy classification - following the same learning curve as the other methods - shows better classification performance compared to the classical methods in case of molecule toxicology prediction problem. Further, we plan to investigate the overfitting problem for our algorithm, in particular, the dependency of classification metrics on the number of considered nodes in graphlets. Other types of descriptions and a parallel version of our algorithm are also promising directions of study.

References

1. Corinna Cortes and Vladimir Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sept. 1995.
2. S. V. N. Vishwanathan, Nicol N. Schraudolph, Risi Kondor, and Karsten M. Borgwardt, "Graph Kernels," *J. Mach. Learn. Res.*, vol. 11, pp. 1201–1242, Aug. 2010.
3. Hiroto Saigo, Sebastian Nowozin, Tadashi Kadowaki, Taku Kudo, and Koji Tsuda, "GBoost: a mathematical programming approach to graph classification and regression," *Machine Learning*, vol. 75, no. 1, pp. 69–89, 2009.
4. Rakesh Agrawal and Ramakrishnan Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, San Francisco, CA, USA, 1994, VLDB '94, pp. 487–499, Morgan Kaufmann Publishers Inc.
5. Adriano Veloso, Wagner Meira Jr., and Mohammed J. Zaki, "Lazy Associative Classification," in *Proceedings of the Sixth International Conference on Data Mining*, Washington, DC, USA, 2006, ICDM '06, pp. 645–654, IEEE Computer Society.
6. Bernhard Ganter and Sergei Kuznetsov, "Pattern Structures and Their Projections," in *Conceptual Structures: Broadening the Base*, Harry Delugach and Gerd Stumme, Eds., vol. 2120 of *Lecture Notes in Computer Science*, pp. 129–142. Springer, Berlin/Heidelberg, 2001.
7. Christoph Helma and Stefan Kramer, "A Survey of the Predictive Toxicology Challenge 2000-2001," *Bioinformatics*, vol. 19, no. 10, pp. 1179–1182, 2003.
8. Nino Shervashidze, S. V. N. Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten M. Borgwardt, "Efficient graphlet kernels for large graph comparison," *Journal of Machine Learning Research - Proceedings Track*, vol. 5, pp. 488–495, 2009.
9. Reinhard Diestel, *Graph Theory (Graduate Texts in Mathematics)*, Springer, August 2005.
10. Horst Bunke and Kim Shearer, "A Graph Distance Metric Based on the Maximal Common Subgraph," *Pattern Recogn. Lett.*, vol. 19, no. 3-4, pp. 255–259, Mar. 1998.
11. Sergei O. Kuznetsov, "Scalable Knowledge Discovery in Complex Data with Pattern Structures," in *PReMI*, Pradipta Maji, Ashish Ghosh, M. Narasimha Murty, Kuntal Ghosh, and Sankar K. Pal, Eds. 2013, vol. 8251 of *Lecture Notes in Computer Science*, pp. 30–39, Springer.
12. Natasa Przulj, "Biological network comparison using graphlet degree distribution," *Bioinformatics*, vol. 23, 2003.
13. Bernhard Ganter and Rudolf Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1997.
14. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Machine-assisted Cyber Threat Analysis using Conceptual Knowledge Discovery

– Position Paper –

Martín Barrère*[1,5], Gustavo Betarte[1], Victor Codocedo[2], Marcelo Rodríguez[1], Hernán Astudillo[3], Marcelo Aliquintuy[3], Javier Baliosian[1], Rémi Badonnel[2], Olivier Festor[2], Carlos Raniery Paula dos Santos[4], Jéferson Campos Nobre[4], Lisandro Zambenedetti Granville[4], and Amedeo Napoli[2]

[1] InCo, Facultad de Ingeniería, Universidad de la República, Uruguay
[2] LORIA/INRIA/CNRS - Nancy, France
[3] Universidad Técnica Federico Santa María, Valparaíso, Chile
[4] Institute of Informatics, Federal University of Rio Grande do Sul, Brazil
[5] Imperial College London, UK

Abstract. Over the last years, computer networks have evolved into highly dynamic and interconnected environments, involving multiple heterogeneous devices and providing a myriad of services on top of them. This complex landscape has made it extremely difficult for security administrators to keep accurate and be effective in protecting their systems against cyber threats. In this paper, we describe our vision and scientific posture on how artificial intelligence techniques and a smart use of security knowledge may assist system administrators in better defending their networks. To that end, we put forward a research roadmap involving three complimentary axes, namely, (I) the use of FCA-based mechanisms for managing configuration vulnerabilities, (II) the exploitation of knowledge representation techniques for automated security reasoning, and (III) the design of a cyber threat intelligence mechanism as a CKDD process. Then, we describe a machine-assisted process for cyber threat analysis which provides a holistic perspective of how these three research axes are integrated together.

1 Introduction

The goal of this paper is to introduce some novel applications of formal concept analysis [13], knowledge discovery in databases and, in a broader sense, artificial intelligence techniques to support security analysis of computer networks and systems. Computer networks are very dynamic environments composed by diverse entities which, on a daily basis, hold thousands of virtual activities. Additionally, they often require configuration changes to satisfy existing or new operational requirements (e.g. new services, upgrading existing versions, replacing faulty hardware). Such dynamicity highly increases the complexity of security management. Even if automated tools help to simplify security tasks there is a

* mbarrere@fing.edu.uy, m.barrere@imperial.ac.uk

need for advanced and flexible solutions able to assist security analysts in better understanding what is happening inside their networks.

The research work we put forward is being developed in the context of the AKD (Autonomic Knowledge Discovery) project [7], a research collaboration effort involving five teams with different expertises. We have identified several key aspects in which the use of artificial intelligence techniques, and particularly formal concept analysis (FCA), can quickly improve on the current state of affairs for processes and tasks in the field of computer and network security. We describe how we envision an adaptation of the conceptual knowledge discovery on databases (CKDD) machinery to provide support in developing scientifically grounded techniques for the domain of cyber threat intelligence. In particular, we are concerned with vulnerability management and cyber threat analysis. We also motivate the benefits of using ontology engineering methods and tools to improve the state of the art of security-oriented automated reasoning.

The remainder of this paper is organized as follows: Section 2 points out the scientific challenges of the research that is being developed in the context of the AKD project. Section 3 motivates three different research fields in which artificial intelligence techniques can be used to provide machine-assisted support to the domain of cyber security. Section 4 describes a cyber threat analysis process aimed at detecting and recognizing security threats within computer systems and points out how and where the techniques previously discussed apply. Finally, Section 5 concludes and summarizes research perspectives.

2 Scientific challenges

Vulnerabilities, understood as program flaws or configurations errors, are used by attackers to bypass the security policies of computer systems. Therefore, vulnerability management mechanisms constitute an essential component of any system intended to be protected. During the last decades, strong research efforts as well as dozens of security tools have been proposed for dealing with security vulnerabilities [5]. However, current security solutions still seem to work under certain boundaries that prevent them to act intelligently and flexibly, i.e. strictly sticked to the available security information in order to analyze, report and eventually remediate found problems.

In addition to this inflexibility, remediating vulnerabilities is already a complex problem and despite the great advances made in this area, remediation tasks are reactive by nature and they can be hard to perform due to costly activities and performance degradation issues. They may also generate consistency conflicts with other system policies. Therefore, our scientific posture in this context is that instead of detecting vulnerable states and then applying several corrective actions, it would be better to anticipate and avoid these vulnerable states in the first place. This objective constitutes a challenging problem. Firstly, mechanisms for understanding the behavior and dynamics of the system are needed. Secondly, sometimes vulnerabilities are not known, so techniques for analyzing

the available knowledge and extracting measures that might allow the system to make decisions are essential.

The aforementioned security challenge gets more complex when considered in dynamic networked scenarios. The accelerated growth of highly heterogeneous and interconnected computer networks has severely increased the complexity of network management. This phenomenon has naturally affected network security where traditional solutions seem unable to cope with this evolving and changing landscape. The main problem is that even when current security techniques may enable high levels of automation, they might fail to achieve their purpose when certain aspects of a managed environment slightly change. We need to provide systems with mechanisms to understand, reason about, and anticipate the surrounding environment. In light of this, we firmly believe that an advanced, flexible, and clever management of security knowledge constitutes one of the key factors to take security solutions to the next level. Our vision is that, independently of the nature of an automated solution (automatically assisting an administrator or automatically making security decisions), the ability to intelligently manage knowledge is essential.

In the broad sense of knowledge management, several scientific areas within the artificial intelligence domain can contribute to achieve our vision. In this work, we identify domains such as formal concept analysis (FCA), ontological engineering, information retrieval (IR), case-based reasoning (CBR), and conceptual knowledge discovery on databases (CKDD), as sound scientific areas that may support a new level of smart cyber security solutions. Fig. 1 illustrates our research strategy for the short, medium and long term.

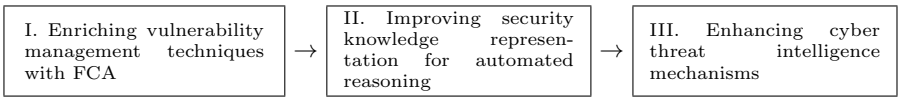


Fig. 1: Research strategy for the short, medium and long term

In the short term (I), our objective is to understand to what extent FCA can enrich and advance the state of the art of vulnerability management techniques. Vulnerability management can be usually seen as the cyclical process of assessing and remediating vulnerabilities. Anticipation techniques are not considered in the classical definition, although the concept of foreseeing future vulnerabilities perfectly fits the vision of flexible and adaptive systems. Therefore, the idea is to begin solving basic problems within the sub-area of vulnerability assessment and progress towards FCA-based mechanisms for anticipating and remediating security vulnerabilities. We understand that a clever use of available knowledge requires a formal and robust underlying machinery that allows systems to process, reason, extract, and extrapolate information and knowledge among other features. In the medium term (II), we aim at investigating the link between current security standard efforts such as the STIX language [3] and knowledge representation methods such as security ontologies. The results of this research

activity may provide a robust support to intelligently deal with security issues. In the long term (III), the objective is to integrate the results and experience obtained in (I) and (II) to develop novel approaches to deal with cyber security threats supported by KDD-based techniques. In the following section, we explain in detail each one of these stages, their impact and importance, and how we envision their development.

3 Research roadmap

3.1 Enriching vulnerability management techniques with FCA

One of the main objectives of our research is the study of vulnerability anticipation mechanisms from the perspective of FCA. Usually, a vulnerability is considered as a combination of conditions that if observed on a target system, the security problem described by such vulnerability is present on that system [5]. Each condition in turn is understood as the state that should be observed on a specific object. When the object under analysis exhibits the specified state, the condition is said to be true on that system. In this context, a vulnerability is a logical combination of conditions and therefore, identifying known vulnerabilities implies the evaluation of logical predicates over computer system states. In brief, we characterize vulnerabilities and system states by the properties they present. From a technical perspective, the OVAL language [2] maintained by MITRE [1], is a standard XML-based security language which permits the treatment and exchange of this type of vulnerability descriptions in a machine-readable manner.

V_1: $c_1 \wedge c_2$
V_2: $c_1 \wedge (c_2 \vee c_3)$
V_3: $\neg c_2 \vee c_3 \vee c_4$
V_4: $\neg c_3$

Table 1: Vulnerabilities as logical formulæ

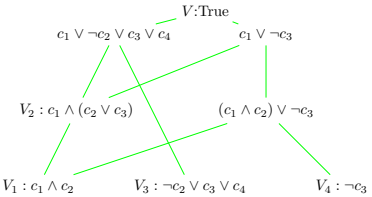


Table 2: Semi-lattice representation of the vulnerability set

As an example, let us consider Table 1 depicting four vulnerabilities $V = \{V_1, V_2, V_3, V_4\}$ as logical formulæ, where $\wedge, \vee, \neg$ represent the logical connectors AND, OR, NOT respectively, and $C = \{c_1, c_2, c_3, c_4\}$ are four system conditions (e.g. "port 80 is open", "httpd server is up", "firewall is off", etc.). A system state s is defined as a set of conditions $c_i \in C$ such that c_i is true on s. Therefore, the process of vulnerability assessment over a system state s can be defined as follows:

$$f(s) = \begin{cases} vulnerable & \exists V_i \in V, s.t. V_i(s) = true \\ safe & otherwise \end{cases}$$

A system state s is considered *vulnerable* if there exists at least one vulnerability that evaluates to true when taking the values from the system for the involved conditions, and *safe* otherwise. For example, considering $s = \{c_1, c_3\}$, it can be observed that $f(s) = vulnerable$ since $V_2(s) = V_3(s) = true$.

From the perspective of FCA [13] and particularly, using the formalization of Logical Concept Analysis (LCA) [12], this can be formalized as follows. Let V be a set of vulnerability labels associated to formulæ in the logic $\mathcal{L}$ with $\wedge, \vee, \neg$ denoting the logical operators and atoms $\mathcal{A}$ containing a set of system conditions $c_i \in C$. A vulnerability label $v \in V$ is associated to a formula in $\mathcal{L}$ through the mapping function $\delta(v) \in \mathcal{L}$.

Let us define the logical context $\mathbb{K} = (V, (\mathcal{L}, \vDash), \delta)$ with the following derivation operators for a subset of vulnerabilities $A \subseteq V$ and a formula $d \in \mathcal{L}$:

$$A^{\square} = \bigvee_{v \in A} \delta(v) \qquad d^{\square} = \{v \in V \mid \delta(v) \vDash d\}$$

For any two vulnerabilities labels $v_1, v_2 \in V$, we have that $v_1 \vDash v_2 \iff v_1 \vee v_2 = v_2$ denotes that v_1 *is a model of* v_2. A pair (A, d) is a formal concept if and only if $A^{\square} = d$ and $d^{\square} = A$. It can be shown that the derivation operators generate a Galois connection between the power set $\wp(V)$ of vulnerability labels and the set of formulæ $\mathcal{L}$ and thus, a concept lattice can be obtained from the logical context $\mathbb{K}$. Within our approach, such a concept lattice generates the search space for vulnerability assessment and correction.

Analogously to the Boolean model of *Information Retrieval* [15], we can use the concept lattice to *classify* the system state s and search for exact or partial answers, i.e. vulnerabilities which affect or *may* affect the system. For instance, the semi-lattice illustrated in Table 2 can be used to understand that if a system is affected by vulnerabilities $V2$ and $V3$, then *it may* be also affected by vulnerability $V1$. In particular, the formula labeled by v satisfies a formula d in some context $\mathcal{K}$ if and only if the concept labeled with v is below the concept labeled with d in the concept lattice of $\mathcal{K}$ [11]. Additionally, using the classification algorithm inspired in case-based reasoning presented in [9], it is easy to show that the assessment process becomes a search in the hierarchy generated by the semi-lattice, i.e. the assessment has a sub-linear complexity.

Vulnerability remediation on the other hand consists in changing the right properties of a system ($c_i \in C$) to bring it into a safe state. This is an explosive combinatorial problem [4]. However, we believe that a concept lattice can be useful to guide the search for corrective actions that do not lead to new vulnerable states. Furthermore, there might be no solution in some cases, so an interesting approach would be to approximate safe solutions by weighting the impact of vulnerabilities using scoring languages such as CVSS (Common Vulnerability Scoring System) [10]. Lastly, our final goal is to understand to what extent FCA can contribute to the process of anticipating vulnerabilities, which basically consists in predicting potential vulnerable states due to changes in the system. Considering known vulnerabilities, a concept lattice can be used as an

approximation map to avoid unsafe configuration changes. Extrapolation and pattern detection mechanisms are also worth to be explored though ontological engineering and data mining techniques might better suit such objectives as discussed in the following section.

3.2 Improving security knowledge representation for automated reasoning

Several vocabularies have been proposed in the context of cyber security. Some of the most important ones are: Structured Threat Information eXpression (STIX), Common Attack Pattern Enumeration and Clasification (CAPEC), Common Vulnerability and Exposures (CVE), Cyber Observables eXpression (CybOX), Malware Attribute Enumeration and Characterization (MAEC) and Common Weakness Enumeration (CWE) [24]. Most of these vocabularies were defined by particular organizations, like MITRE and NIST, to facilitate the exchange of information regarding vulnerabilities, security issues and attack descriptions.

The benefits of introducing vocabularies are plenty and well-known. They establish a common language that can be used by different organizations to describe the same concepts and provide a framework for documentation allowing the structured and systematized creation of a body of knowledge. Vocabularies have proven not only be relevant for humans, but for autonomous agents in several applications as well. At the syntactic level, they enable different systems to communicate in a common pre-defined structured manner. At the semantic level, vocabularies have played a major role in the last decade allowing autonomous agents to *reason* about the information within a dataset. For example, let us consider a security analyst looking through different databases for a *malware* that could affect a given system. A malware is a very generic term used to identify a piece of software specially designed to violate the security integrity of a computer system. Thus, the search task can be very difficult given that there are several types of malware, namely trojan horses, spywares, backdoors, worms, among others. Instead, a vocabulary could easily integrate these descriptions by stating that trojan horses, spywares, backdoors and worms are *types of* malware. An autonomous agent can profit from the vocabulary by automatically inferring that an object catalogued as a "trojan horse" is relevant for the search of "malware".

In the semantic web, vocabularies are usually supported by ontologies, a meta-model to provide a structured description of the concepts in a given domain [21]. Ontologies can provide different levels of description, namely at the entity level, at the relational level and at the instance level. The entity level describes the concepts that compose a given domain (Malware, Trojan Horse, Spyware) and their attributes (Malware *has_name*, Trojan Horse *has_target_os*, etc.). The relational level describes relations among concepts (Trojan *is_a_type_of* Malware, Trojan Horse *has_target_operating_system* Windows, etc.) and their attributes (*is_a_type_of* is a non-symmetric, transitive relation). Finally, the instance level describes the relations between instances, their types (trojan1 *is_a*

Trojan), and their attributes (trojan1 *has_name* "Zeus"). Furthermore, ontologies support a similar level of inference as first-order logic through its logical formalism called description logics.

Several research communities have undertaken the task of formalizing their domain knowledge with vocabularies. and many of them have moved forward towards describing their vocabularies through ontology definitions. For example, in [8] an ontology learning approach is proposed for the astronomical domain. In [14] the authors propose an ontology to document software architecture decisions providing an automated annotation process over software design documents. In [22], the authors propose a knowledge discovery process to build and populate an ontology for the cultural heritage domain using a relational database schema. Extensive reviews on ontology learning and construction using formal concept analysis can be found in [18, 20, 23].

As mentioned before, the domain of cyber security has already acknowledged the benefits of defining common vocabularies. Furthermore, initial steps have been taken towards building a comprehensive ontology definition which integrates the different vocabularies within the domain. In [24], the authors describe the process through which they manually crafted a domain ontology with the goal of supporting security analysts in the task of detecting cyber threats. This work is indeed a big step forward, however we are confident that the use of state of the art ontology learning techniques, particularly formal concept analysis, can greatly improve the quality of an ontology for cyber security. For instance, techniques like ontology alignment [23] can overcome overlapping issues in current vocabularies for cyber security, a fact that is oversought in [24]. The great potential for automatically building description logic knowledge bases using FCA [8, 20] would allow to further extend the support provided to security analysts in a more dynamic environment, a major drawback in manual approaches for ontology building. Finally, the definition of a domain ontology for cyber security is a necessary condition to support more advanced data mining techniques. In our project, this represents a milestone that would enable us to provide security analysts with advanced features for threat detection such as integrated search from multiple repositories [16], partial matching based on case-based reasoning [9], or document annotation [14].

3.3 Enhancing cyber threat intelligence mechanisms

The traditional approaches for cyber security, which have mainly focused on understanding and addressing vulnerabilities in computer systems, are still necessary but not longer sufficient enough. Effective defense against current and future threats requires a deep understanding of the behavior, capability and intent of the adversary. Threat environments have evolved from widespread disruptive activity to more targeted, lower-profile multi-stage attacks aiming at achieving specific tactical objectives and establishing a persistent foothold into the threatened organization. This is what is called an Advanced Persistent Threat (APT). The nature of APTs requires for more proactive defense strategies in contrast to the traditional reactive cyber security approach. To be proactive, defenders need

to move beyond traditional incident response methodologies and techniques. It is necessary to stop the adversary before he can exploit the security weaknesses of the system. In the cyber domain, cyber intelligence is the understanding of the adversary capabilities, actions and intent. According to [19]: *Cyber intelligence seeks to understand and characterize things like: what sort of attack actions have occurred and are likely to occur; how can these actions be detected and recognized; how can they be mitigated; who are the relevant threat actors; what are they trying to achieve; what are their capabilities, in the form of tactics, techniques and procedures (TTP) they have leveraged over time and are likely to leverage in the future; what sort of vulnerabilities, misconfigurations or weaknesses are likely to target; what actions have they taken in the past; etc.*

One important objective of our research is to develop techniques and tools for providing assistance to accomplish different cyber threat intelligence procedures. In particular, we are focused on processes aiming at leveraging capacities for threat environment identification (type of attack, from where, how) and early detection of vulnerability exploitation attempts. We also aim at the generation and enrichment of (semantically structured) knowledge repositories, preferably in a way that is decoupled from the specifics of a particular technology for conducting threat analysis and correlation.

For a threat analysis tool to be useful in practice, two features are crucial: i) the model used in the analysis must be able to automatically integrate formal vulnerability specifications from the bug-reporting community and formal attack scenarios from the cyber security concerned community; ii) it is desirable for the analysis to be able to scale to complex networks involving numerous machines and devices. As a more ambitious goal, we aim at developing a prototype of an engine, in the spirit of MulVAL [17], able to consume low-level alerts (e.g. taken from OVAL scanning activities) and produce high-level attack predictions based on the scenario under analysis.

4 A machine-assisted approach for cyber threat analysis

In this section we put forward a cyber threat analysis process aimed at detecting and/or recognizing (potential) security attacks. We explain the most relevant procedures involved in the analysis and point out how and where automated support can be provided using the techniques discussed in sections 3.1, 3.2 and 3.3. The cyber threat analysis process, depicted in Fig. 2, embodies procedures that give support to the key phases of the search of compromise: derivation of threat indicators, collection of evidence, evaluation of the results and decision. In what follows we explain the process in further detail.

1. The process begins at step 1 with a security analyst providing information about some identified threat or anomaly, and characteristics of the target system. This information constitutes the *initial seed* for the cyber threat analysis, and might specify for instance, a compromise involving a suspicious file found on a Linux system. The involved information shall be represented using the STIX language, in particular using the notion of *indicator of com-*

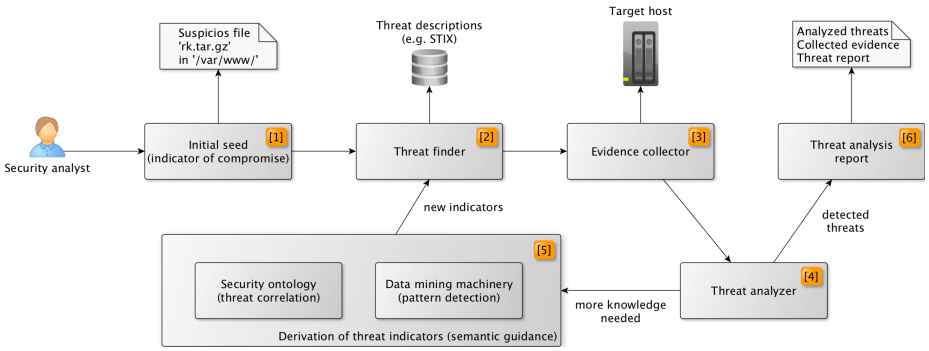


Fig. 2: Cyber threat analysis overview

promise. One such indicator allows to specify the different types of objects that can be found on a computing system/network such as ports, processes, threads, files, etc. Additionally, an indicator may capture metadata for the involved objects as well as logical relations between them thus providing further information to security analysts.

2. Once the seed has been provided, a *search of compromise* is performed at step 2. To that end, the threat finder component queries a database containing machine-readable descriptions of known threats specified in a formal language such as STIX. Only those cyber threats which are found to be related with the provided information are considered for subsequent analysis.
3. The retrieved threat descriptions are then used at step 3 by the evidence collector component to gather all the relevant information from the target system in order to decide whether the latter is compromised by at least one of the identified related cyber threats. The process of information gathering involves, for instance, collecting the list of open ports or running processes in the system. Standard languages such as OVAL provide great support for evidence specification and automated collection procedures [6].
4. The collected evidence is then evaluated by the threat analyzer component at step 4 in order to determine the level of compromise of the system. A target system may be considered compromised by a specific cyber threat if it presents a combination of objects (*threat indicator*) which are commonly found on infected systems. The threat analyzer decides whether the collected evidence is sufficient enough to indicate that the target system has been compromised or conversely whether more knowledge is needed to diagnose its status. In the first case, the process moves to step 6 where the information about the detected cyber threats is provided to the security analyst. Otherwise, the process continues at step 5 where a semantic machinery is used to derive new indicators that may lead to cyber threats not previously evaluated.

5. In the case that none of the spotted cyber threats are found on the system, a derivation process is triggered at step 5 in order to select new cyber threats that were not analyzed before. This new selection is performed by deriving threats related to the relevant evidence found on the system while gathering information in the previous stage. Derivation mechanisms may vary according to the available information and context, and they constitute a key objective within this research work. The FCA-based technique described in Section 3.1 may provide a map for finding vulnerable configurations close to the current system state. Additionally, two sub-components may semantically guide the search for new related threats. As discussed in Section 3.2, a security ontology may relax strict descriptions making context awareness procedures more flexible, i.e. security information that is not explicitly encoded a priori can be derived by considering semantic associations. Data mining techniques on the other hand may provide the ability to extrapolate information and extract security patterns thus increasing detection capabilities even more. The process of derivation (step 5), threat identification (step 2), collection (step 3) and analysis (step 4) shall be repeated until a conclusion or a stop condition is reached.
6. The outcome of a finished search process may be either that the system appears to be compromised or not enough evidence has been found to determine its compromise status. In any case, the process informs about the tested cyber threats as well as the evidence found on the system at step 6 in order to assist the security analyst to proceed with the analysis.

Open discussion. The selection of information and techniques for inferring and discovering new knowledge might be assisted by a human being, the security analyst in this case, thus following a methodology closer to CKDD. However, interesting research questions arise from this scenario. One of them is to what extent can we automate the whole process and let a security solution to make decisions for us? Going one step further we pose the question of autonomic solutions where self-adaptive and self-governed approaches come into scene. Our vision is that to achieve any of these objectives, a clever knowledge management is essential. In that context, we believe that FCA and CKDD may highly contribute to accomplish such goal.

5 Conclusions and perspectives

In this paper we have motivated and explained how different artificial intelligence techniques, in particular FCA and CKDD, can be used to enhance the state of the art of machine-assisted cyber security analysis. In addition to the objectives depicted in our research roadmap, we also target the construction of an experimental testbed for emulating hostile and unsafe environments. This can provide the ability to deploy implementation prototypes and anticipation solutions in order to evaluate the feasibility, scalability and accuracy of our approach. We have already experimented with a preliminary version of a tool that provides

mechanical support for conducting the cyber threat analysis process described in section 4. We are convinced that the extension of the tool with mechanisms that make use of conceptual knowledge discovery techniques will greatly improve the accuracy and efficiency of the process.

References

1. MITRE Corporation. `http://www.mitre.org/`. Last visited on May 17, 2015.
2. OVAL Language. `http://oval.mitre.org/`. Last visited on May 17, 2015.
3. Structured Threat Information expression. `http://stix.mitre.org/`. Last visited on May 17, 2015.
4. M. Barrère, R. Badonnel, and O. Festor. A SAT-based Autonomous Strategy for Security Vulnerability Management. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS'14)*, May 2014.
5. M. Barrère, R. Badonnel, and O. Festor. Vulnerability Assessment in Autonomic Networks and Services: A Survey. *IEEE Communications Surveys & Tutorials*, 16(2):988–1004, 2014.
6. M. Barrère, G. Betarte, and M. Rodríguez. Towards Machine-assisted Formal Procedures for the Collection of Digital Evidence. In *Proceedings of the 9th Annual International Conference on Privacy, Security and Trust (PST'11)*, pages 32 –35, July 2011.
7. M. Barrère et al. Autonomic Knowledge Discovery for Security Vulnerability Prevention in Self-governing Systems. `http://www.sticamsud.org/`. Last visited on May 17, 2015.
8. R. Bendaoud, Y. Toussaint, and A. Napoli. Pactole: A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In *Proceedings of the 16th international conference on Conceptual Structures: Knowledge Visualization and Reasoning*, pages 203–216, 2008.
9. V. Codocedo, I. Lykourentzou, and A. Napoli. A semantic approach to concept lattice-based information retrieval. *Annals of Mathematics and Artificial Intelligence*, pages 1–27, 2014.
10. CVSS, Common Vulnerability Scoring System. `http://www.first.org/cvss/`. Last visited on April 12, 2015.
11. S. Ferré and R. D. King. A dichotomic search algorithm for mining and learning in domain-specific logics. *Fundam. Inform.*, 66(1-2):1–32, 2005.
12. S. Ferré and O. Ridoux. A Logical Generalization of Formal Concept Analysis. In B. Ganter and G. W. Mineau, editors, *ICCS*, volume 1867 of *LNCS*, pages 357–370, 2000.
13. B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer, Dec. 1999.
14. C. López, V. Codocedo, H. Astudillo, and L. M. Cysneiros. Bridging the gap between software architecture rationale formalisms and actual architecture documents: An ontology-driven approach. *Sci. Comput. Program.*, 77(1):66–80, 2012.
15. C. D. Manning, P. Raghavan, and H. Schtze. *Introduction to Information Retrieval*. July 2008.
16. N. Messai, M.-D. Devignes, A. Napoli, and M. Smaïl-Tabbone. BR-Explorer: A sound and complete FCA-based retrieval algorithm (Poster). In *ICFCA*, Dresden/Germany, 2006.

17. X. Ou, S. Govindavajhala, and A. W. Appel. Mulval: A logic-based network security analyzer. In *Proceedings of the 14th Conference on USENIX Security Symposium - Volume 14*, SSYM'05, pages 8–8, Berkeley, CA, USA, 2005. USENIX Association.
18. J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene. Formal concept analysis in knowledge processing: A survey on applications. *Expert Syst. Appl.*, 40(16):6538–6560, 2013.
19. S. Barnum. Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX). Technical report, The MITRE Corporation, 2013.
20. B. Sertkaya. A survey on how description logic ontologies benefit from formal concept analysis. *CoRR*, abs/1107.2822, 2011.
21. S. Staab and R. Studer. *Handbook on Ontologies*. Springer Publishing Company, Incorporated, 2nd edition, 2009.
22. R. Stanley, H. Astudillo, V. Codocedo, and A. Napoli. A conceptual-kdd approach and its application to cultural heritage. In *Concept Lattices and their Applications*, pages 163–174, 2013.
23. G. Stumme. Formal concept analysis. In *Handbook on Ontologies*, pages 177–199. 2009.
24. B. E. Ulicny, J. J. Moskal, M. M. Kokar, K. Abe, and J. K. Smith. Inference and Ontologies. In *Cyber Defense and Situational Awareness*, Advances in Information Security. 2014.

RAPS: A Recommender Algorithm Based on Pattern Structures

Dmitry I. Ignatov[1] and Denis Kornilov[1]

National Research University Higher School of Economics
dignatov@hse.ru
http://www.hse.ru

Abstract. We propose a new algorithm for recommender systems with numeric ratings which is based on Pattern Structures (RAPS). As the input the algorithm takes rating matrix, e.g., such that it contains movies rated by users. For a target user, the algorithm returns a rated list of items (movies) based on its previous ratings and ratings of other users. We compare the results of the proposed algorithm in terms of precision and recall measures with Slope One, one of the state-of-the-art item-based algorithms, on Movie Lens dataset and RAPS demonstrates the best or comparable quality.

Keywords: Formal Concept Analysis, Pattern Structures, Recommender Systems, Collaborative Filtering, RAPS, Slope One

1 Introduction and related work

Formal Concept Analysis (FCA)[1] is a powerful algebraic framework for knowledge representation and processing [2,3]. However, in its original formulation it deals with mainly Boolean data. Even though original numeric data can be represented by so called multi-valued context, it requires concept scaling to be transformed to a plain context (i.e. a binary object-attribute table). There are several extensions of FCA to numeric setting like Fuzzy Formal Concept Analysis [4,5]. In this paper, to recommend particular user items of interest we use Pattern Structures, an extension of FCA to deal with data that have ordered descriptions. In fact, we use interval pattern structures that were proposed in [6] and successfully applied, e.g., in gene expression data analysis [7].

The task of recommending items to users according to their preferences expressed by ratings of previously used items became extremely popular during the last decade partially because of famous NetFlix 1M$ competition [8]. Numerous algorithms were proposed to this end. In this paper we will mainly study item-based approaches. Our main goal is to see whether FCA-based approaches are directly applicable to the setting of recommender systems with numeric data. Previous approaches used concept lattices for navigation through the recommender space and allowed to recommend relevant items faster than online computation in user-based approach, however it requires expensive offline computations and a substantial storage space [9]. Another approach tries to effectively use Boolean factorisation based on formal concepts and follows user-based k-nearest neighbours strategy [10]. A parameter-free approach that exploits a neighbourhood of the object concept for a particular user also proved its effectiveness

[11] but it has a predecessor based on object-attribute biclusters [12] that also capture the neighbourhood of every user and item pair in an input formal context. However, it seems that within FCA framework item-based techniques for data with ratings have not been proposed so far. So, the paper bridges the gap.

The paper is organised as follows. In Section 2, basic FCA definitions and interval pattern structures are introduced. Section 3 describes SlopeOne [13] and RAPS with examples. In Section 4, we provide the results of experiments with time performance and precision-recall evaluation for MovieLens dataset. Section 5 concludes the paper.

2 Basic definitions

Formal Concept Analysis. First, we recall several basic notions of Formal Concept Analysis (FCA) [1]. Let G and M be sets, called the set of objects and attributes, respectively, and let I be a relation $I \subseteq G \times M$: for $g \in G$, $m \in M$, gIm holds iff the object g has the attribute m. The triple $\mathbb{K} = (G, M, I)$ is called a *(formal) context*. If $A \subseteq G$, $B \subseteq M$ are arbitrary subsets, then the *Galois connection* is given by the following *derivation operators*:

$$\begin{aligned} A' &= \{m \in M \mid gIm \text{ for all } g \in A\}, \\ B' &= \{g \in G \mid gIm \text{ for all } m \in B\}. \end{aligned} \tag{1}$$

The pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$ is called a *(formal) concept (of the context K)* with *extent* A and *intent* B (in this case we have also $A'' = A$ and $B'' = B$).

The concepts, ordered by $(A_1, B_1) \geq (A_2, B_2) \iff A_1 \supseteq A_2$ form a complete lattice, called *the concept lattice* $\underline{\mathfrak{B}}(G, M, I)$.

Pattern Structures. Let G be a set of objects and D be a set of all possible object descriptions. Let $\sqcap$ be a similarity operator. It helps to work with objects that have non-binary attributes like in traditional FCA setting, but those that have complex descriptions like intervals or graphs. Then $(D, \sqcap)$ is a meet-semi-lattice of object descriptions. Mapping $\delta : G \to D$ assigns an object g the description $d \in (D, \sqcap)$.

A triple $(G, (D, \sqcap), \delta)$ is a pattern structure. Two operators $(\cdot)^{\square}$ define Galois connection between $(2^G, \subseteq)$ and $(D, \sqcap)$:

$$A^{\square} = \prod_{g \in A} \delta(g) \text{ for } A \subseteq G \tag{2}$$

$$\begin{aligned} d^{\square} = \{g \in G \mid d \sqsubseteq \delta(g)\} \text{ for } d \in (D, \sqcap), \text{ where} \\ d \sqsubseteq \delta(g) \iff d \sqcap \delta(g) = d. \end{aligned} \tag{3}$$

For a set of objects A operator 2 returns the common description (pattern) of all objects from A. For a description d operator 3 returns the set of all objects that contain d.

A pair (A,d) such that $A \subseteq G$ and $d \in (D,\sqcap)$ is called a pattern concept of the pattern structure $(G,(D,\sqcap),\delta)$ iff $A^{\square} = d$ and $d^{\square} = A$. In this case A is called a pattern extent and d is called a pattern intent of a pattern concept (A,d). Pattern concepts are partially ordered by $(A_1,d_1) \leq (A_2,d_2) \iff A_1 \subseteq A_2 (\iff d_2 \sqsubseteq d_1)$. The set of all pattern concepts forms a complete lattice called a pattern concept lattice.

Intervals as patterns. It is obvious that similarity operator on intervals should fulfill the following condition: two intervals should belong to an interval that contains them. Let this new interval be minimal one that contains two original intervals. Let $[a_1,b_1]$ and $[a_2,b_2]$ be two intervals such that $a_1,b_1,a_2,b_2 \in \mathbb{R}$, $a_1 \leq b_1$ and $a_2 \leq b_2$, then their similarity is defined as follows:

$$[a_1,b_1] \sqcap [a_2,b_2] = [\min(a_1,a_2), \max(b_1,b_2)].$$

Therefore

$$\begin{aligned} [a_1,b_1] \sqsubseteq [a_2,b_2] &\iff [a_1,b_1] \sqcap [a_2,b_2] = [a_1,b_1] \\ &\iff \big[\min(a_1,a_2), \max(b_1,b_2)\big] = [a_1,b_1] \\ &\iff a_1 \leq a_2 \text{ and } b_1 \geq b_2 \iff [a_1,b_1] \supseteq [a_2,b_2] \end{aligned}$$

Note that $a \in \mathbb{R}$ can be represented by $[a,a]$.

Interval vectors as patterns. Let us call p-adic vectors of intervals as interval vectors. In this case for two interval vectors of the same dimension $e = \langle [a_i,b_i] \rangle_{i \in [1,p]}$ and $f = \langle [c_i,d_i] \rangle_{i \in [1,p]}$ we define similarity operation via the intersection of the corresponding components of interval vectors, i.e.:

$$e \sqcap f = \langle [a_i,b_i] \rangle_{i \in [1,p]} \sqcap \langle [c_i,d_i] \rangle_{i \in [1,p]} \iff e \sqcap f = \langle [a_i,b_i] \sqcap [c_i,d_i] \rangle_{i \in [1,p]}$$

Note that interval vectors are also partially ordered:

$$e \sqsubseteq f \iff \langle [a_i,b_i] \rangle_{i \in [1,p]} \sqsubseteq \langle [c_i,d_i] \rangle_{i \in [1,p]} \iff [a_i,b_i] \sqsubseteq [c_i,d_i]$$

for all $i \in [1,p]$.

3 Recommender Algorithms

3.1 Slope One

Slope One is one of the common approaches to recommedations based on collaborative filtering. However, it demonstrates comparable quality with more complex and resource demanding algorithms [13]. As it was shown in [14], SlopeOne has the highest recall on MovieLens and Netflix datasets and acceptable level of precision: "Overall, the algorithms that present the best results with these metrics are SVD techniques, tendencies-based and slope one (although its precision is not outstanding)."

We use this algorithm for comparison purposes.

Slope One deals with rating matrices as input data. In what follows the data contains movies ratings by different users. That is $M = \{m_1, m_2, \ldots, m_n\}$ is a set of movies, $U = \{u_1, u_2, \ldots, u_k\}$ is a set of users. The rating matrix can be represented by many-valued formal context (U, M, R, I), where $R = \{1, 2, 3, 4, 5, *\}$ is a set of possible ratings and a triple $(u, m, r) \in I$ means that the user u marked by the rating r the movie m. Whenever it is suitable we also use r_{ij} notation for rating of movie m_j by user u_i.

In case a user u has not rated a movie m, we use $m(u) = r = *$, i.e. missing rating.

Let us describe the algorithm step by step.

1. The algorithm takes a many valued context of all users' ratings, the target user u_t for which it generates recommendations. It also requires $left\_border$ and $right\_border$ for acceptable level of ratings, i.e. if one wants to receive all movies with ratings between 4 and 5, then left and right borders should be 4 and 5 respectively. The last pair of parameters: one needs to set up minimal and maximal scores (min_border and max_border) that are acceptable for our data. It means that if the algorithm predicts rating 6.54 as a score and maximal score is bounded by 5, then 6.54 should be treated as 5.
2. The algorithm finds the set of all movies evaluated by the target user $S(u_t)$.
3. For every non-evaluated movie $m_j \in M \setminus S(u_t)$ by u_t execute step 4), and by so doing calculate the predicted rating for the movie m_j. After that go to step 5).
4. For every evaluated movie $m_i \in S(u_t)$ by u_t calculate $S_{j,i}(U \setminus \{u_t\})$, the set of users that watched and evaluated movies m_i and m_j. In case $S_{j,i}(U \setminus \{u_t\})$ is non-empty, that is $|S_{j,i}(U \setminus \{u_t\})| > 0$, calculate the deviation: $dev_{j,i} = \sum\limits_{u_k \in S_{j,i}(U \setminus \{u_t\})} \frac{r_{k,j} - r_{k,i}}{|S_{j,i}(U \setminus \{u_t\})|}$ and add i to R_j.
   After all current deviations found, calculate the predicted rating: $P(u_t)_j = \frac{1}{|R_j|} \sum\limits_{i \in R_j} (dev_{j,i} + r_{t,i})$, where $R_j = \{i | m_i \in S(u_t), i \neq j, |S_{j,i}(U \setminus \{u_t\})| > 0\}$. In case R_j is empty, the algorithm cannot make a prediction.
5. By this step Slope One found all predicted ratings $P(u_t)$ for movies from $M \setminus S(u_t)$. The algorithm recommends all movies with predicted ratings in the preferred range $left\_border \leq P(u_t)_j \leq right\_border$, taking into account minimal and maximal allowed values.

If one needs top-N ranked items, she can sort the predicted scores from the resulting set in decreasing order and select first N corresponding movies.

Example 1.

Consider execution of Slope One on the dataset from Table 1.

Table 1. Example of data for Slope One

user\movie	m_1	m_2	m_3
u_1	5	3	2
u_2	3	4	*
u_3	*	2	5

Let us try to predict the rating for u_3 and movie m_1.

1. Let $left\_border = 4$, $right\_border = 5$, $min\_border = 1$, and $max\_border = 5$.
2. We find $S(u_3) = \{m_2, m_3\}$, the set of evaluated movies by the target user.
3. $M \setminus S(u_3) = \{m_1\}$
4. $S_{1,2}(U \setminus \{u_3\}) = \{u_1, u_2\}$
   $dev_{1,2} = \frac{(r_{1,1}-r_{1,2})+(r_{2,1}-r_{2,2})}{(|\{u_1,u_2\}|)} = ((5-3)+(3-4))/2 = 0.5$
   $S_{1,3}(U \setminus \{u_3\}) = \{u_1\}$
   $dev_{1,3} = (r_{1,1} - r_{1,3})/(|\{u_1\}|) = (5-2)/1 = 3$
   $R_1 = \{2,3\}$
   $P(u_3)_1 = 1/|R_j|(dev_{1,2} + r_{3,2} + dev_{1,3} + r_{3,3}) = 1/2(0.5+2+3+5) = 5.25$
5. Taking into account the maximal rating boundary, the algorithm predicts 5 for movie m_1, and therefore recommends user u_3 to watch it.

3.2 RAPS

Our approach, RAPS (Recommender Algorithm based on Pattern Structures), works with the same many valued context as Slope One.

Let us describe the algorithm.

1. It takes the context (U, M, R, I) with all ratings, and a target user u_t . It also requires $left\_border$ and $right\_border$ for preferred ratings, i.e. if one wants to get all movies rated in range from 4 to 5, then $left\_border = 4$ and $right\_border = 5$.
2. Define the set of movies $M_t = \{m_{t_1}, \ldots, m_{t_q}\}$ that the target user u_t liked, i.e. the ones that she evaluated in the range $[left\_border, right\_border]$.
3. For each movie $m_{t_i} \in M_t$ apply eq. 3. and find the set of users that liked the movie $A_{t_i} = [left\_border, right\_border]^{\square}_{m_{t_i}}$ for $1 \leq i \leq q$. As a result one has the set of user subsets: $\{A_{t_1}, \ldots, A_{t_q}\}$.
4. For each $A_{t_i}, 1 \leq i \leq q$ apply eq. 2 to find its description; in our case it is a vector of intervals $d_{t_i} = A^{\square}_{t_i} = \langle [a_1^{t_i}, b_1^{t_i}], \ldots, [a_n^{t_i}, b_n^{t_i}] \rangle$ for $1 \leq i \leq q$. Note that, in case a particular user u_x from A_{t_i} has not rated m_y, i.e. $r_{x,y} = *$, then the algorithm does not take it into account.
5. At the last step compute the vector $\mathbf{r} = (R_1, \ldots, R_n) \in \mathbb{N}^n$ (or $\mathbb{R}^n$ in general case), where

$$R_j = |\{i | 1 \leq i \leq q, [a_j^{t_i}, b_j^{t_i}] \subseteq [left\_border, right\_border]\}| \text{ , i.e.}$$

   for each movie m_j the algorithm counts how many of its descriptions $[a_j^{t_i}, b_j^{t_i}]$ are in $[left\_border, right\_border]$. If $R_j > 0$, then the algorithm recommends watching the movie.

Top-N movies with the highest ratings can be selected in similar way.

Let us shortly discuss the time computational complexity. Step 2 requires $O(|M|)$ operations, steps 3, 4 and 5 perform within $O(|M||U|)$ each. Therefore, the algorithm time complexity is bounded by $O(|M||U|)$.

Example 2

Consider execution of RAPS on the tiny dataset from Table 2.

Let us find a recommendation for user u_7.

Table 2. Example of data for RAPS

user\movie	m_1	m_2	m_3	m_4	m_5	m_6
u_1	5	3	1	3	5	3
u_2	4	4	1	5	4	3
u_3	5	*	*	3	*	4
u_4	*	3	4	*	2	4
u_5	4	*	4	5	4	*
u_6	3	4	5	5	*	3
u_7	5	4	2	*	*	*

1. The input of the algorithm: $t = 7$, $left\_border = 4$ and $right\_border = 5$.
2. $M_7 = \{m_1, m_2\}$
3. $A_1 = [4,5]^{\square}_{m_1} = \{u_1, u_2, u_3, u_5\}$
   $A_2 = [4,5]^{\square}_{m_2} = \{u_2, u_6\}$
4. $$d_1 = A_1^{\square} = \langle [a_1^1, b_1^1], [a_2^1, b_2^1], [a_3^1, b_3^1], [a_4^1, b_4^1], [a_5^1, b_5^1], [a_6^1, b_6^1] \rangle = \\ = \langle [4,5], [3,4], [1,4], [3,5], [4,5], [3,4] \rangle$$
   For example, interval $[a_6^1, b_6^1]$ is found as follows:
   $$[a_6^1, b_6^1] = [\min(r_{1,6}, r_{2,6}, r_{3,6}, r_{5,6}), \max(r_{1,6}, r_{2,6}, r_{3,6}, r_{5,6})] = \\ = [\min(3,3,4,*), \max(3,3,4,*)] = [\min(3,3,4), \max(3,3,4)] = [3,4].$$
   The rest intervals are found in similar way.
   $d_2 = A_2^{\square} = \langle [3,4], [4,4], [1,5], [5,5], [4,4], [3,3] \rangle$
5. Taking into account the left and right bounds, the algorithm recommends movies m_1 and m_5 from d_1 and m_2, m_4 and m_5 from d_2. Therefore $R = (1,1,0,1,2,0)$, i.e. without already assessed movies by u_7, we recommend her to watch m_4 and m_5.

4 Experimental evaluation

4.1 Data

For our experimentation we have used freely available data from MovieLens website[1]. The data collection was gathered within The GroupLens Research Project of Minnesota University in 1997–1998. The data contains 100000 ratings for 1682 movies by 943 different users. Each user rated no less than 20 movies. That is we have 100000 tuples in the form:

user id | item id | rating | timestamp.

Each tuple shows user id, movie id, the rating she gave to the movie and time when it happened.

[1] `http://grouplens.org/datasets/movielens/`

4.2 Quality assessment

Firstly, for quality assessment of Slope One and RAPS we used precision and recall measures. Note that we cannot use Mean Absolute Error (MAE) directly, since RAPS actually assume a whole interval like [4,5] for a particular movie, not a number. We select 20% of users to form our test set and for each test user we split her rated movies into two parts: the visible set and the hidden set. The first set consists of 80% rated movies, and the second one contains the remaining 20%. Moreover, to make the comparison more realistic, movies from the first set were evaluated earlier than those from the second one. It means that first we sort all ratings of a given user by timestamp and then perform splitting.

There is a more general testing scheme based on bimodal cross-validation from [15], which seems to us the most natural and realistic: users from the test set keep only $x\%$ of their rated movies, and the remaining $y\%$ of their ratings are hidden. Thus, by considering each test user in this way, we model a real user whose ratings to other movies are not yet clear, but at the same time we have all ratings' information about the training set of users. In other words, we hide only rectangle of size $x\%$ of test users by $y\%$ of hidden items. One can vary x and y during the investigation of the behaviour of methods under comparison, where the size of top-N recommended list is set to be equal to $y\%$. The part of hidden items can be selected randomly or by timestamp (preferably for realistic scenario). Note that there is no a gold standard approach to test recommender systems, however, there are validated sophisticated schemes [16]. The main reason is the following: with only off-line data in hands we cannot verify whether the user will like a not yet seen movie irrespective of assumption that she has seen our recommendation. However, for real systems there is a remedy such as A/B testing, which is applicable only in online setting [17].

The adjusted precision and recall are defined below:

$$precision = \frac{|\{relevant\ movies\} \cap \{retrieved\ movies\} \cap \{test\ movies\}|}{|\{retrieved\ movies\} \cap \{test\ movies\}|} \tag{4}$$

$$recall = \frac{|\{relevant\ movies\} \cap \{retrieved\ movies\} \cap \{test\ movies\}|}{|\{relevant\ movies\} \cap \{test\ movies\}|} \tag{5}$$

These measures allow us to avoid the uncertainty since we do not know how actually a particular user would assess a recommended movie. However, in real recommender system, we would rather ask a user whether the recommendation was relevant, but in our off-line quality assessment scheme we cannot do that. In other words, we assume that for a test user at the moment of assessment there are no movies except the training and test ones.

Another issue, which is often omitted in papers on recommender systems, is how to avoid uncertainty when denominators in Precision and Recall are equal to zero (not necessarily simultaneously).

To define the mesaures precisely based on the peculiarities of recommendation task and common sense, we use two types of the definitions for cases when the sets of retrieved and relevant movies for particular user and recommender are empty.

Precision and Recall of the first type are defined as follows:

– If the sets of relevant movies and retrieved ones are empty, then $Precision = 0$ and $Recall = 1$.
– If the set of relevant movies is empty, but the set of retrieved ones is not, then $Precision = 0$ and $Recall = 1$.
– If we have the non-empty set of relevant movies, but the set of retrieved movies is empty, then $Precision = 0$ and $Recall = 0$.

Precision of the second type is less tough, but Recall remains the same:

– If the sets of relevant movies and retrieved ones are empty, then $Precision = 1$ (since we should not recommend any movie and the recommender has not recommended anything).
– If the set of relevant movies is empty, but the set of retrieved ones is not, then $Precision = 0$.
– If we have the non-empty set of relevant movies, but the set of retrieved movies is empty, then $Precision = 1$ (since the recommender has not recommended anything, its output does not contain any non-relevant movie).

4.3 Results

We have performed three series of tests:

1. A movie is worth to watch if its predicted mark is 5 (i.e. it is [5,5]).
2. A mark is good if it is from [4,5].
3. Any mark from [3,5] is good.

All the tests are performed in OS X 10.9.3 with Intel Core i7 2.3 GHz and 8 Gb of memory. The algorithm were implemented in MATLAB – R2013a. The results are presented in Table 3. Note that the reported Precision and Recall are of the first type.

Table 3. RAPS vs Slope One Results

Algorithm name	Preference Interval	Average time, s	Average precision	Average recall	F1-measure
RAPS	[5,5]	**3**.62	**1**9.42	**5**0.52	28.06
Slope One	[5,5]	18.90	1.57	23.41	2.94
RAPS	[4,5]	**1**8.23	**5**5.61	**6**3.33	59.22
Slope One	[4,5]	18.90	53.99	30.39	38.89
RAPS	[3,5]	32.98	80.11	**8**3.65	81.84
Slope One	[3,5]	**1**8.90	**8**3.81	81.88	82.83

The criteria are average execution time in seconds, average precision and recall. From the table one can see RAPS is drastically better than Slope One by the whole set of criteria in [5,5]. For [4,5] interval both approaches have comparable time and precision, but Slope One has two times lower recall. For [3,5] interval the algorithms demonstrate similar values of precision and recall but RAPS 1.5 times slower on average.

However, since the compared approaches are different from the output point view (RAPS provides the user with an interval of possible ratings but SlopeOne does it by a single real number), we perform thorough comparison varying the lower bound of acceptable recommendations and using both types of the adjusted precision and recall measures.

From Fig. 1 one can conclude that RAPS dominates SlopeOne in most cases by Recall. As for Precision, even though for [5,5] interval RAPS is significantly better, after lower bound of 4.4 SlopeOne shows comparable but slightly better Precision in most cases.

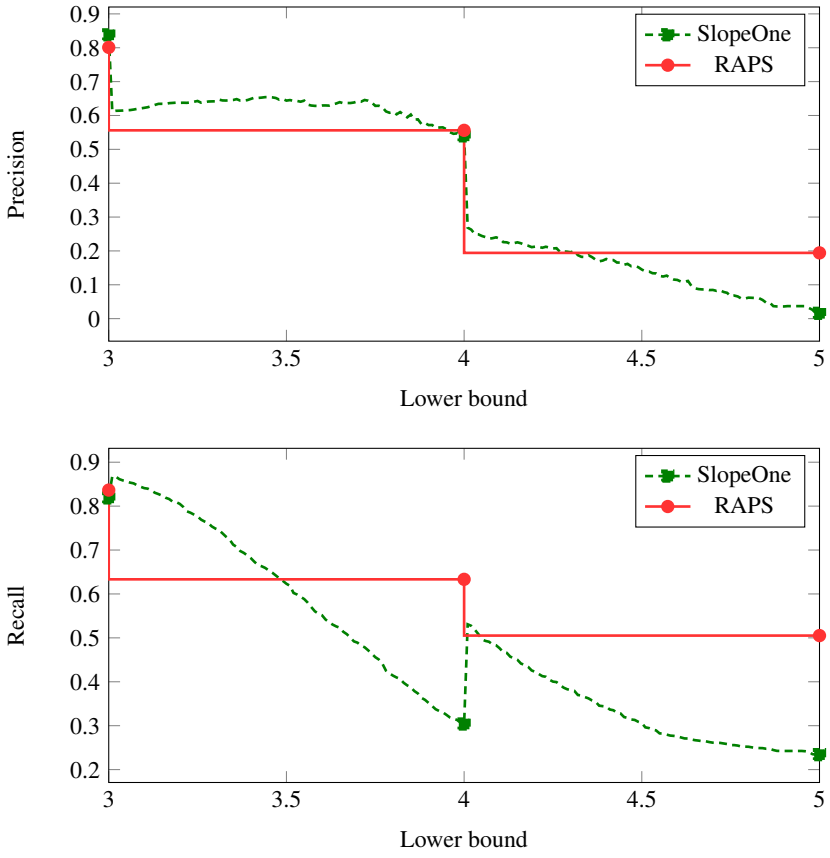


Fig. 1. Precision and Recall of the first type for RAPS and SlopeOne for the varying lower bound

From Fig. 2 one can see that SlopeOne is significantly better in terms of precision. Only on the interval [3,5] the difference between SlopeOne and RAPS is negligible (the lower bound value equals 3). The reasonable explanations is as follows: for

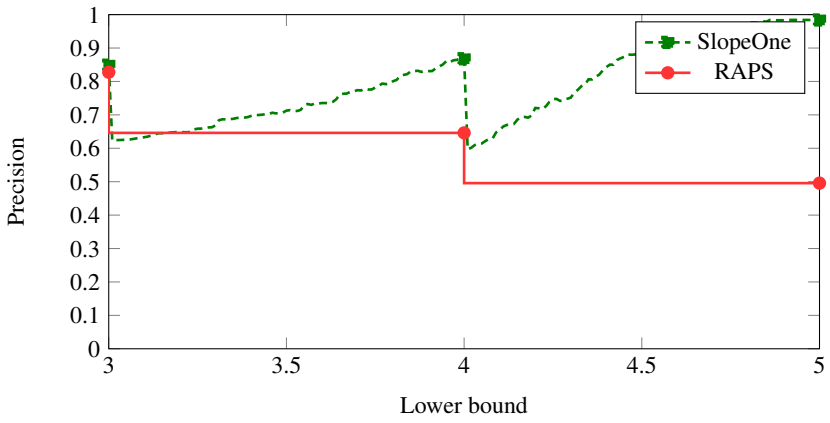


Fig. 2. Precision of the second type for RAPS and SlopeOne for the varying lower bound

SlopeOne there are more cases when $\{retrieved\ movies\} = \emptyset$ irrespective of the size of $\{relevant\ movies\}$. Remember that in such cases Precision of the second type is equal to 1. In other words SlopeOne is really more precise (or even concise): in such cases it just does not recommend anything. However, it can be hardly judged in movie recommendation domain that a recommender is good when it does not recommend.

We can conclude that the proposed recommender technique based on pattern structures has its right to be used. Since the Slope One algorithm was exploited in real recommender systems [13], we can suggest our technique for usage as well.

5 Conclusion and further work

In this paper we proposed the technique for movie recommendation based on Pattern Structures (RAPS). Even though this algorithm is oriented to movie recommendations, it can be easily used in other recommender domains where users evaluate items.

The performed experiments (RAPS vs Slope One) showed that recommender system based on Pattern Structures demonstrates acceptable precision, better recall in most cases and reasonable execution time.

Of course, in future RAPS should be compared with other recommender techniques to make a final conclusion about its applicability. An interplay between interval-based recommendations and regression-like ones deserves a more detailed treatment as well.

The further work can be continued in the following directions:

1. Further modification and adjustment of RAPS.
2. Development of the second variant of Pattern Structures based recommender. There is a conjecture that for the second derivation operation (operator Galois from eq.3) being applied to more than one movie with high marks we may obtain relevant predictions as well.

3. Comparison with existing popular techniques, e.g. SVD and SVD++.

Acknowledgments We would like to thank Mehdi Kaytoue, Sergei Kuznetsov and Sergei Nikolenko for their comments, remarks and explicit and implicit help during the paper preparations. The first author has been supported by the Russian Foundation for Basic Research grants no. 13-07-00504 and 14-01-93960 and made a contribution within the project "Data mining based on applied ontologies and lattices of closed descriptions" supported by the Basic Research Program of the National Research University Higher School of Economics. We also deeply thank the reviewers for their comments and remarks that helped.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer-Verlag New York, Inc., Secaucus, NJ, USA (1999)
2. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. Expert Syst. Appl. **40**(16) (2013) 6538–6560
3. Poelmans, J., Kuznetsov, S.O., Ignatov, D.I., Dedene, G.: Formal concept analysis in knowledge processing: A survey on models and techniques. Expert Syst. Appl. **40**(16) (2013) 6601–6623
4. Belohlávek, R.: What is a fuzzy concept lattice? II. In Kuznetsov, S.O., Slezak, D., Hepting, D.H., Mirkin, B., eds.: Rough Sets, Fuzzy Sets, Data Mining and Granular Computing - 13th International Conference, RSFDGrC 2011, Moscow, Russia, June 25-27, 2011. Proceedings. Volume 6743 of Lecture Notes in Computer Science., Springer (2011) 19–26
5. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Fuzzy and rough formal concept analysis: a survey. Int. J. General Systems **43**(2) (2014) 105–134
6. Ganter, B., Kuznetsov, S.: Pattern structures and their projections. In Delugach, H., Stumme, G., eds.: Conceptual Structures: Broadening the Base. Volume 2120 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2001) 129–142
7. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Information Sciences **181**(10) (2011) 1989 – 2001 Special Issue on Information Engineering Applications Based on Lattices.
8. Bell, R.M., Koren, Y.: Lessons from the netflix prize challenge. SIGKDD Explorations **9**(2) (2007) 75–79
9. du Boucher-Ryan, P., Bridge, D.: Collaborative recommending using formal concept analysis. Knowledge-Based Systems **19**(5) (2006) 309 – 315
10. Ignatov, D.I., Nenova, E., Konstantinova, N., Konstantinov, A.V.: Boolean matrix factorisation for collaborative filtering: An fca-based approach. In Agre, G., Hitzler, P., Krisnadhi, A.A., Kuznetsov, S.O., eds.: Artificial Intelligence: Methodology, Systems, and Applications - 16th International Conference, AIMSA 2014, Varna, Bulgaria, September 11-13, 2014. Proceedings. Volume 8722 of Lecture Notes in Computer Science., Springer (2014) 47–58
11. Alqadah, F., Reddy, C., Hu, J., Alqadah, H.: Biclustering neighborhood-based collaborative filtering method for top-n recommender systems. Knowledge and Information Systems (2014) 1–17
12. Ignatov, D.I., Kuznetsov, S.O., Poelmans, J.: Concept-based biclustering for internet advertisement. In Vreeken, J., Ling, C., Zaki, M.J., Siebes, A., Yu, J.X., Goethals, B., Webb, G.I., Wu, X., eds.: 12th IEEE International Conference on Data Mining Workshops, ICDM Workshops, Brussels, Belgium, December 10, 2012, IEEE Computer Society (2012) 123–130

13. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. In: Proceedings of the 2005 SIAM International Conference on Data Mining. 471–475
14. Cacheda, F., Carneiro, V., Fernández, D., Formoso, V.: Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. ACM Trans. Web **5**(1) (February 2011) 2:1–2:33
15. Ignatov, D.I., Poelmans, J., Dedene, G., Viaene, S.: A new cross-validation technique to evaluate quality of recommender systems. In Kundu, M.K., Mitra, S., Mazumdar, D., Pal, S.K., eds.: Perception and Machine Intelligence - First Indo-Japan Conference, PerMIn 2012, Kolkata, India, January 12-13, 2012. Proceedings. Volume 7143 of Lecture Notes in Computer Science., Springer (2012) 195–202
16. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems. RecSys '10, New York, NY, USA, ACM (2010) 39–46
17. Radlinski, F., Hofmann, K.: Practical online retrieval evaluation. In Serdyukov, P., Braslavski, P., Kuznetsov, S.O., Kamps, J., Rüger, S.M., Agichtein, E., Segalovich, I., Yilmaz, E., eds.: Advances in Information Retrieval - 35th European Conference on IR Research, ECIR 2013, Moscow, Russia, March 24-27, 2013. Proceedings. Volume 7814 of Lecture Notes in Computer Science., Springer (2013) 878–881

Finding a Lattice of Needles in a Haystack: Forming a Query from a Set of Items of Interest

Boris A.Galitsky

Knowledge-Trail Inc San Jose CA USA
bgalitsky@hotmail.com

Abstract. We introduce a new type of query, a lattice query, which is intended to assist a user of a search engine in query formulation. An associated methodology of search is proposed, where instead of submitting an exact query, the user provides a set of text samples which are the basis of generalization. The lattice query system automatically forms a search query from this generalization and verifies the relevancy of search results to the provided set of samples. Lattice queries are also designed to assist in building templates for information extraction tasks: instead of specifying certain keywords or linguistic patterns, a developer can give a list of samples and leave generalization task to the system. Lattice queries are formed from individual sentences and from paragraphs of text as well. An open source contribution of lattice query search component as a part of OpenNLP is described.

Keywords: search engine query, generalizing from samples, interactive search

1 Introduction

Today, the significant portion of information is obtained via search engines. Horizontal web search engines and well as specialized vertical search engines such as product search and health recommendations are the essential sources of information in the respective domains. Modern open source big data search and exploration systems like Solr and Elasticsearch are broadly used for access and analysis of big data. However, intelligence features such as search relevance and adequate analysis, retrieval and exploration of large quantities of natural language texts are still lacking. It is still had to find information in a horizontal or vertical domain unless precise search keywords are known to the user [1,2,3].

Frequently, novice users of search engines experience difficulties formulating their queries, especially when these queries are long. It is often hard for user who is new to a domain to pick proper keywords. Even for advanced users exploring data via querying, including web queries, it is frequently difficult to estimate proper generality/specificity of a query being formulated. Lattice querying makes it easier for a broad range of user and data exploration tasks to formulate the query: given a few examples, it formulates the query automatically.

In this work we intend to merge the efficiency of distributed computing framework with the intelligence features of data exploration provided by NLP technologies. We

introduce the technique of lattice querying which automatically forms the query from the set of text samples provided by a user by generalizing them in the level of parse trees. Also the system produces search results by matching parse trees of this query with that of candidate answers. Lattice queries allow increase in big data exploration efficiency since they form multiple "hypotheses" concerning user intent and explore data from multiple angles (generalizations).

Exploring data, mostly keyword query and phrase query are popular, as well as natural language-like ones. Users of search engines appreciate more and more 'fuzzy match' queries, which help to explore new areas where the knowledge of exact keywords is lacking. Using synonyms, taxonomies, ontologies and query expansions helps to substitute user keywords with the domain-specific ones to find what the system believes users are looking for [7].

Nowadays, search engines ranging from open source to enterprise offer a broad range of queries with string character-based similarity. They include Boolean queries, span queries which restrict the distances between keywords in a document, regular expressions queries which allow a range of characters at certain positions, fuzzy match queries and more-like-this which allow substitution of certain characters based on string distances. Other kinds of queries allow expressing constraints in a particular dimension, such as geo-shape query.

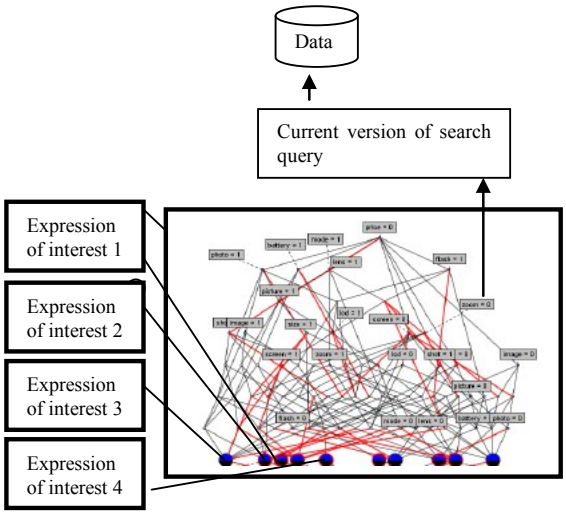


Fig. 1: The idea of lattice query

The idea of lattice query is illustrated in Fig. 1. Instead of a user formulating a query exploring a dataset, he or she proves a few samples (expressions of interest) so that the system builds the lattice of all generalizations for these samples. The system then formulates a query for each lattice node.

Proceeding from a keyword query to regexp or fuzzy one allows making search more general, flexible, assists in exploration of a new domain, as set of

document with unknown vocabulary. What can be a further step in this direction? We introduce lattice queries, based on natural language expressions which are generalized into an actual query. A lattice query contains rich linguistic information, which is derived by generalization of sample expressions (phrases or sentences) specified by the user. Instead of getting search results similar to a given expression (done by 'more like this' query), we first build the commonality expression between all or subsets of the given sample expressions, and then use it as a query. A lattice query includes words as well as attributes such as entity types and verb attributes.

2 Simple lattice queries

Let us start with an employee search example. Let us imagine a company looking for the following individuals:

A junior sale engineer expert travels to customers on site.

A junior design expert goes to customer companies.

A junior software engineer rushes to customer sites.

Given the above set of samples, we need to form a job-search query which would give us candidates somewhat similar to what we are looking for. A trivial approach would be to just turn each sample into a query and attempt to find an exact match. However most of times it would not work, so such queries need to release some constraints. How to determine which constraints need to be dropped and which keywords are most important?

To do that, we apply generalization to the set of these samples. For the entities and attributes, we form the least general generalization. The seniority of the job (adjective) 'junior' will stay. The job activity (noun phrase) varies, so we generalize them into <job-activity>. The higher-level reference to the job is 'expert' and is common for all three cases, so stays. The verb for job responsibility varies, so we use <action>, which can be further specified as <moving_action>, using verb-focused ontologies like VerbNet. To generalize the last noun phrase, we obtain the generalization <customer, NP>.

junior <any job activity> expert <action> customer-NP.

This is a lattice query, which is expected to be run against job descriptions and find the cases which are supposed to be most desired, according to the set of samples.

In terms of parse trees of the potential sentences to be matched with the lattice query, we rewrite it as

JJ-junior NP- NN-expert VP-* NN-customer NP-**

The lattice query read as *find me a junior something expert doing-something-with customer of-something.*

Now we show how this template can applied to accept/reject a candidate answer Cisco junior sale representative expert flew to customers data centers.

We represent the lattice query as a conjunction of noun phrases (NP) and verb phrases (VP) set:

[[NP [DT-a JJ-junior NN- NN-*], NP [NN*-customers]], [VP [VB-* TO-to NN*-customers]]]*

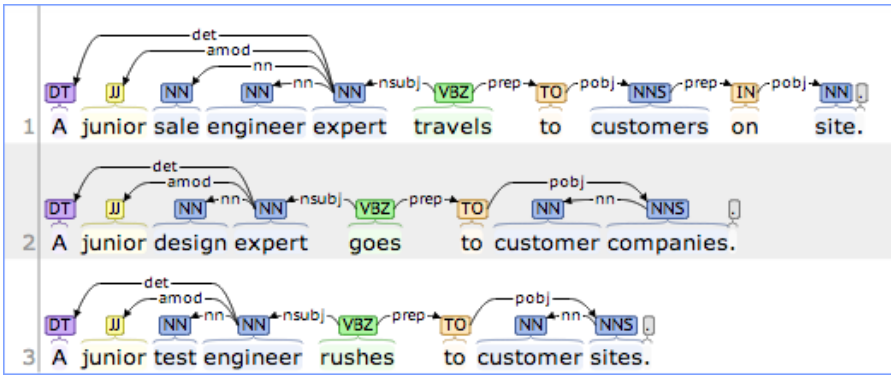
The first NP covers the beginning of the lattice query above, and the second NP covers the end. VP covers the second half of the lattice query starting from *doing-something...*

The generalization between the lattice query and a candidate answer is

[[NP [JJ-junior NN- NN-*], NP [NN*-customers]], [VP [VB-* TO-to NN*-customers]]]*

One can see that the NP part is partially satisfied (the article a does not occur in the candidate answer) and VP part is fully satisfied.

Here are parse trees for three samples



Generalizing these three, we obtain the lattice query to run against a dataset:

[[NP [DT-a JJ-junior NN-* NN-*], NP [NN*-customers]], [VP [VB-* TO-to NN*-customers]]]

One can see that using lattice queries, one can be very sensitive in selecting search results. Searching for a token followed by a word with certain POS instead of just a single token gives a control over false-positive rate. Automated derivation of such constraint allows user to focus on cases instead of making efforts to generate a query which would keep expected search results in and unwanted out.

Definition: a lattice query Q is satisfied by a sentence S, if Q^S = S.

In practice a weak satisfaction is acceptable, where Q^S <= S, but there are constraints on the parts of the lattice query:

- A number of parts in in Q^S should be the same as in Q;
- All words (not POS-* placeholders) from Q should also be in Q^S.

3 More complex lattice queries

Text samples to form a lattice query can be typed, but also can be taken from text already written by someone. To expand the dimensionality of content exploration, samples can be paragraph-size texts.

Let us consider an example of a safety-related exploration task, where a researcher attempts to find a potential reason for an accident. Let us have the following texts as incidents descriptions. These descriptions should be generalized into a lattice query to be run against a corpus of texts for the purpose of finding a root cause of a situation being described.

Crossing the snow slope was dangerous. They informed in the blog that an ice axe should be used. However, I am reporting that crossing the snow field in the late afternoon I had to use crampons.

I could not cross the snow creek since it was dangerous. This was because the previous hiker reported that ice axe should be used in late afternoon. To inform the fellow hikers, I had to use crampons going across the show field in the late afternoon.

As a result of generalization [5, 6, 8] from two above cases, we will obtain a set of expressions for various ways of formulating commonalities between these cases. We will use the following snapshot of a corpus of text to illustrate how a lattice query is matched with a paragraph:

I had to use crampons to cross snow slopes without an ice axe in late afternoon this spring. However in summer I do not feel it was dangerous crossing the snow.

We link two phrases in different sentences since they are connected by a rhetoric relation based on *However* …

```
  rel: <sent=1-word=1..inform> ===> <sent=2-word=4..report>
  From [<1>NP'They':PRP]
  TO    [<4>NP'am':VBP,    NP'reporting':VBG,    <8>NP'the':DT,
<9>NP'snow':NN, <10>NP'field':NN, <11>NP'in':IN, <12>NP'the':DT,
<13>NP'late':JJ,      <14>NP'afternoon':NN,      <15>NP'I':PRP,
<16>NP'had':VBD,         <17>NP'to':TO,         <18>NP'use':VB,
<19>NP'crampons':NNS]
```

We are also linking phrases of different sentences based on communicative actions:

```
  rel: <sent=1-word=6..report> ===> <sent=2-word=1..inform>
  From [<4>NP'the':DT, <5>NP'previous':JJ, <6>NP'hiker':NN]
  TO    [<1>NP'To':TO,    <2>NP'inform':VB,    <3>NP'the':DT,
<4>NP'fellow':JJ, <5>NP'hikers':NNS]
```

As a result of generalizing two paragraphs, we obtain the lattice query:

```
  [[NP [NN-ice NN-axe ], NP [DT-the NN-snow NN-* ], NP [PRP-i
], NP [NNS-crampons ], NP [DT-the TO-to VB-* ], NP [VB-* DT-the
NN-* NN-field IN-in DT-the JJ-late NN-afternoon (TIME) ]], [VP
[VB-was JJ-dangerous ], VP [VB-* IN-* DT-the NN-* VB-* ], VP
[VB-* IN-* DT-the IN-that NN-ice NN-axe MD-should VB-be VB-used
], VP [VB-* NN-* VB-use ], VP [DT-the IN-in ], VP [VB-reporting
IN-in JJ-late NN-afternoon (TIME) ], VP [VB-* NN*-* NN-* NN*-*
], VP [VB-crossing DT-the NN-snow NN-* IN-* ], VP [DT-the NN-*
```

```
NN-field IN-in DT-the JJ-late NN-afternoon (TIME) ], VP [VB-had
TO-to VB-use NNS-crampons ]]]
```

Notice that potential safety-related "issues" are *ice-axe, snow, crampons, being at a … field during later afternoon, being dangerous, necessity to use ice-axe, crossing the snow*, and others. These issues occur in both samples, so that are of a potential interest. Now we can run the formed lattice query against the corpus and observe which issues extracted above are confirmed. A simple way to look at it is as a Boolean OR query: find me the conditions from the list which is satisfied by the corpus. The generalization for the lattice query and the paragraph above turns out to be satisfactory:

```
  [[NP [NN-ice NN-axe ], NP [NN-snow NN*-* ], NP [DT-the NN-
snow ], NP [PRP-i ], NP [NNS-crampons ], NP [NN-* NN-* IN-in JJ-
late NN-afternoon (TIME) ]], [VP [VB-was JJ-dangerous ], VP [VB-
* VB-use ], VP [VB-* NN*-* IN-* ], VP [VB-crossing NN-snow NN*-*
IN-* ], VP [VB-crossing DT-the NN-snow ], VP [VB-had TO-to VB-
use NNS-crampons ], VP [TO-to VB-* NN*-* ]]] => matched
```

Hence we got the confirmation from the corpus that the above hypotheses, encoded into this lattice query, are true. Notice that forming a data exploration queries from the original paragraphs would contain too many keywords and would produce too much marginally relevant results.

4 Evaluation of Performance of Lattice Queries

We conduct evaluation for complex information extraction tasks such as identifying communicative actions and detecting emotional states. Also, we perform evaluation for the rhetoric relation domain: this task is necessary to build a set of parse trees for a paragraph, linking its parse trees. We draw the comparison between information extraction based on the means available within Elasticsearch and Solr framework:

- keyword Boolean queries,
- span queries where the distance between keywords in text is constrained, and
- lattice query-based information extraction.

The corpus is based on he set of customer complains, where both communicative actions and emotions are frequent and essential for complaint analysis tasks. Evaluation was conducted by quality assurance personnel.

We observe in Table 1 that the information extraction F-measure for Keywords and Regular expressions is both 64% for querying indexed data and string search (although the former is about 50 times faster). Relying on span queries gives just 2% increase in F-measure, whereas using lattice queries delivers further 10% improvement.

In this work we introduced a new type of query for search engine framework, the lattice query, which is intended to facilitate the process of an abstract data exploration. Instead of having a user formulate a query, one or more instances are automatically formed from sample expressions. To derive a lattice query, as well as measure relevance of a question to an answer, an operation of syntactic generalization [8, 9] is

used. It finds a maximal common sub-trees between the parse trees for the sample text fragments, and also it finds the maximum common sub-trees between the parse trees for the lattice query and that of the candidate answers. In the latter case, the size of the common sub-trees is a measure of relevance for a given candidate search results.

Method / Task	Keywords and Regexps – finding in string			Keywords and Regexp Queries – Lucene index			Span and 'Like' Queries – Lucene index			Lattice queries First Lucene index then verification by ^		
	P/R		Speed	P/R		Speed	P/R		Speed	P/R		Speed
Extracting communicative actions	64	71	1	63	72	0.02	68	70	0.05	82	75	15.1
Extracting emotional state	62	70	1.2	59	70	0.02	64	68	0.05	80	74	18.2
Extracting rhetoric relation	56	65	1.5	56	66	0.02	59	70	0.05	77	70	25.4

Table 1: Evaluation of lattice query-based information extraction tasks.

We now proceed to an information extraction example in the security domain. One needs to identify text which contains some information of personal interest, such as social security numbers or driver's license numbers, as well as names and addresses of individuals. The requirement is to identify a reference to a person, her activity and a certain document with an id number. Rather than attempting to come up with a rule, a developer of this system specifies the training samples:

"John Doe send her california license 4567456"
"Mary Smith hid her US social number 666-66-6666"
"Jennifer Poppins got her identification 8765"
"Andrew Chen lost his Oregon driver license 731234"

The rules obtained from this samples cover following cases:

"Judith Jain received her washington license 4567456"
"Mary Jones send her Canada prisoner id number 666666666"
"Mary Jones send her Canada prisoner id number 666666666"
"Mary Stewart hid her Mexico cook id number 666666666"
"Robin mentioned her Peru fisher id 2345"

“Peter Doe hid his Bolivia set id number 666666666"
"Robin mentioned her best Peru fisher man id 2345"

But leaves the following cases out:

"Spain hid her Canada driver id number 666666666"
"John Poppins hid her prisoner id 666666666"
"Microsoft announced its Windows Azure release number 666666666"
"John Poppins hid her Google id 666666666"

It should be obvious for the reader the negative set included cases not related to a possible leakage of personal information.

In our evaluation we compared the conventional information extraction approach where extraction rules are expressed using keywords and regular expressions, with the one where rules are frame queries. We observed that frame queries improve both precision and recall of information extraction by producing more sensitive rules, compared to sample expressions which would serve as extraction rules otherwise. An importance of the lattice queries in data exploration is that only the most important keywords are submitted for web search, and neither single document nor keyword overlap deliver such the set of keywords.

References

1. Borgida ER, DL McGuinness, Asking Queries about Frames. Proceedings of the 5th Int. Conf. on the Principles of Knowledge Representation and Reasoning 1996, 340—349.
2. Bill MacCartney, Michel Galley, and Christopher D. Manning, A phrase-based alignment model for natural language inference. The Conference on Empirical Methods in Natural Language Processing (EMNLP-08), Honolulu, HI, October 2008.
3. Galitsky, B. *Natural Language Question Answering System: Technique of Semantic Headers*. Advanced Knowledge International, Australia (2003).
4. Galitsky, B., Josep Lluis de la Rosa, Gábor Dobrocsi. *Inferring the semantic properties of sentences by mining syntactic parse trees*. Data & Knowledge Engineering. Volume 81-82, November (2012) 21-45.
5. Galitsky, B., Daniel Usikov, Sergei O. Kuznetsov: *Parse Thicket Representations for Answering Multi-sentence questions*. 20th International Conference on Conceptual Structures, ICCS 2013 (2013).
6. Galitsky, B. Machine Learning of Syntactic Parse Trees for Search and Classification of Text. Engineering Application of AI, http://dx.doi.org/10.1016/j.engappai.2012.09.017, (2012).
7. Galitsky, B. Transfer learning of syntactic structures for building taxonomies for search engines. Engineering Applications of Artificial Intelligence. Volume 26 Issue 10, November, 2013, Pages 2504-2515.
8. Galitsky B, Ilvovsky D, Kuznetsov SO, Strok F. Matching sets of parse trees for answering multi-sentence questions. Recent Advances in Natural Language Processing. 2013. doi:http://www.aclweb.org/anthology/R13-1037.
9. Galitsky, B., Learning parse structure of paragraphs and its applications in search. Engineering Applications of Artificial Intelligence 01/2014; 32:160–184.